Evaluating the Use of Pedagogical Virtual Machine with Augmented Reality to Support Learning Embedded Computing Activity

Malek Alrashidi, Michael Gardner, and Vic Callaghan
School of Computer Science and Electronic Engineering,
The University of Essex, Colchester, UK
+447446102060
{mqaalr, mgardner, vic}@essex.ac.uk

ABSTRACT
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. To the student, this kind of technology often has abstract behaviours where the only information/things people 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. Augmented reality (AR) can overcome this issue and produce a magic-lens view for revealing hidden embedded computing activities. This can result in learners achieving a better level of knowledge and awareness of the technology, as well as higher learning outcomes. AR on its own will not improve the learning processes without first considering how to manage and represent the hidden information. Therefore, a pedagogical virtual machine (PVM) model was employed, and to evaluate the learning effectiveness of the proposed model. We conducted an experiment based on a problem-solving educational mobile robot task. Twenty students participated in the experimental (AR approach) and control (conventional approach) group. The result showed that the augmented reality approach was more effective in increasing students' computational thinking and learning outcomes. In addition, the augmented reality approach reduced both time completion and debugging times.

CCS Concepts
Human-centered computing → Mixed / augmented reality

Keywords
Mixed Reality; Augmented Reality; Real-Time Feedback; Robot; Embedded Computing; Learning and Teaching; Learning Object; Pedagogical Virtual Machine.

1. INTRODUCTION

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
ICCAE '17, February 18-21, 2017, Sydney, Australia
© 2017 ACM. ISBN 978-1-4503-4809-6/17/02…$15.00
DOI: http://dx.doi.org/10.1145/3057039.3057088

With the advanced technology available today, people no longer feel isolated from every aspect of the world. People can discover, show, immerse and communicate with everyday objects and involve themselves in new experiences whether in education, entertainment, or industry by using varied means. By adding a magic layer between people and objects, invisible things become visible and this allows people to create a clear picture of the object. The emergence of immersive learning environments has shown great potential for teaching and learning [1]–[3]. These environments exploit new advances in mixed reality, augmented reality, and virtual reality technologies, to provide new and more context-sensitive learning experiences for students. 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 allow learners to discover, visualise, engage and interact more meaningfully within the learning context [4]. The learning context is now more embedded in the learning activities and are, 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 [5]. 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 [6]. 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 [7]. 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 [8]. Students' view should involve the invisible things that have no real representation when learning abstract concepts [9].

However, augmented reality (AR) can overcome this issue and produce a magic-lens view for revealing hidden embedded computing activities. This can result in learners achieving a good level of knowledge and awareness of the technology, as well as higher learning outcomes. AR on its own will not improve the learning processes without first considering how to manage and represent the hidden information. Therefore, in a previous paper [10], we proposed a pedagogical virtual machine (PVM) model that aims to embed learning processes with computational objects that provide an explanation to learners and developers to acquire a deep understanding of the technology being learnt (e.g., making invisible pedagogical and computation components visible).

The PVM is composed of four main layers: the data layer, aggregation layer, pedagogical layer, and the user interface layer. The data layer is responsible for sniffing any data that is being transmitted on the embedded computing. Then, the aggregation layer collects this data and produces a more meaningful representation. The pedagogical layer takes the aggregated data and maps it to the learning activities based on the preset learning design. Finally, the user interface layer is where learners visualise and view the workflow of the learning activities using AR with information from the other PVM layers (data to pedagogical layers). In addition, the learner can track their progress and obtain instant feedback based on their performance on the learning activity.

In this paper, we present the experimental evaluations for the PVM model based on a problem-solving mobile robot learning activity task based on modularised embedded computing components and explore the effects of the model on students' learning achievements. The experiment examines three questions:

1. Does augmented reality improve students' computational thinking on decomposing problem-solving tasks compared to the traditional approach?
2. Do students who use PVM with AR gain better learning achievement than those who use the conventional (hands-on) approach?
3. Does PVM with AR reduce the time taken for solving a learning activity task and with fewer trials?

The paper starts by stating related work (section 2), then describes the experimental design used to gather evidence of the value of the concepts proposed in the model (section 3). After that, the paper presents the experimental results (section 4) and concludes by discussing the findings of the study and their broader values for the research area (section 5).

2. RELATED WORK

The use of mixed reality educational environment has been promoted especially for learning activity that involves embedded computing or artefacts. Peña-Ríos et al. [11] proposed a mixed reality collaboration environment for science and engineering laboratories. The environment allows remote students to work with students at classroom lab on embedded computing physical objects. The physical object has identical virtual copy in virtual world called xReality object. They defined xReality objects as a smart networked object that tied to virtual representation, updated and maintained in real time within a mixed-reality environment. Therefore, any change or action happening in either worlds, it is reflecting the other in real-time. This allows remote students to not feel isolated from classroom labs, and can collaborate without any recourse limitations/restriction. However, this environment designed to help remote student to collaborate and work with their pairs in physical laboratories, but for students who work in physical laboratories do not gain extra knowledge or enrich their understanding about the physical object.

Another study related to mixed reality educational environment has been developed in [12]. They designed a toolkit called virtual touch which aims to connect tangible object to virtual environment for learning programming concept in computer science. The system architecture aims to facilitate teacher or creator to install, configure and programming various type of technologies without concerning about technical details that involve using tangible elements with virtual world. Thus, with the ease of use of the system, teachers can create a learning activity for their students, regardless teachers' programming and electronic skills. The system architecture consists of four layers that describe the workflow of the system. The fist layer is communication which related to physical object and specify which connection will be utilised whether Bluetooth, USB, Ethernet, etc. The second layer is device management which facilitate the interaction of middleware with hardware technologies such as Kinect, Arduino and Phidgets. Then, service layer provides several facilities such as gesture and object recognition, data management, virtual environment communication and configuration. The final layer is presentation which display graphical information to the user in a virtual world, and it illustrates the interaction with the tangible interface. The system helps students to understand abstract concept such as sorting algorithm, and reducing the complexity. However, the system looked at tangible elements with embedded sensor and presented the information of these elements on virtual world, but did not consider computation objects such as robotics. Also, it did not exploit AR as virtualising tool to assess users in solving learning activity.

In terms of real time feedback, Liu et al. (2012) proposed an AR 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. By pointing mobile phones camera at the marker, it presents an AR view with outlines of the physical controls as well as instruction to complete the task such as entering values or pressing buttons. The system provides colour overlay as real-time visual feedback which indicate correct or wrong uses. This approach improves significantly task performance and learning experience compared to text, picture and AR without real-time feedback.

AR has been used to aid users view complex data in robotic systems. Freund et al. [13] developed a state oriented modelling technique that integrated in virtual reality worlds. This method allows users to view 3D state information about autonomous systems. Daily et al. [14] use AR to visualise information from distributed robots swarms. The robots are used to communicate and work cooperatively to provide information about intruder. Users use AR to collect and overlay information from each robot and highlights the shortest path to the intruder. Similarly, Amstutz and Fagg [15] use AR to present information of large number of sensors and robots. Collecting the state of these objects and visualise them via AR help users in rescue and search task.

Collett and MacDonald [16] argues that mobile robot programming lacks of suitable tools that assist developers at debugging phase. Robot is interacting with the environment which makes programming challenging. This interaction causes additional complexity to programmers by first understanding what data robot is receiving and, second how robot is interpreting that data. To overcome this problem, they proposed an AR debugging tool that supports programmer views of robot's world. Instead of using simulated world by developers, AR allows them to view the robot world with additional virtual object in the real environment. Thus, developers do not need to shift between the real environment and the robot world as AR 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 robot may miscalculate the orientation and the distance to the nearest obstacle due to incorrect values entered by developers, in this case, AR can immediately show this error in graphical representation. However, the limitation of this study is that they used AR as visualisation tools that represent robot data only in 3D or 2D graphical view, and placed the abstract data such as current state and task progress to the console in plain text output. The system lacks of text support which if considered could add more value to discover errors and bugs. Visualisation without meaningful explanation could impede developers to identify the cause of the problem immediately. Thus, there is a need to structure robot data in AR view and explain how these data help developers and learners achieving their task goal.

Magnenat et al. [17] claims that AR and visual feedback enhance high-school students ability to understand event-handling concept in computer science. They proposed an AR system with integrated visual feedback that overlays the executed events on robot in real-time. The system provides timelines that show location and time of the execution at the physical location. This help student to understand what the robot is doing and 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 AR system were stress and this was due to AR system complexity such as complex setup and system sometime lose tracking. Lalonde et al [18] proposed a predictor for programming mobile robot known as robot observability which is utilised for diagnostic transparency by ensuring that the incremental procedure of building and debugging robot programs is made available. This is an essential tool students can use in the diagnosis of a misbehaving robot. Students, by audio feedback, are capable of creating a tool that enhances the predictor performance through identification of the robots internal state evolution. The robot, for example, is capable of speaking its action and stating its purpose. Furthermore, the authors reported their survey, by indicating that 86% of students were convinced that data logging and visual interfaces are valuable tools for debugging. The study had not taken the broader implication in the use of AR as a visual interface capable of guiding students and revealing the procedures of internal communication, which in real time are taking place within the physical objects, into consideration.

Chan et al.[19] in a different study, proposed and evaluated of LightUp design which is an AR 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 among themselves magnetically. The implementation of LightUp is in the form of a mobile application that makes "informational lens" utilising computer recognition in the identification of electrical parts available, and supplementing the image with visualisations by making the invisible circuit visible. Children are helped by this system in real time learning, understanding, and constructing of circuits through simulation. The disadvantage of this study is its reliance on the use of a simulation in extracting information for the learning activity while failing to utilise physical objects.

Similarly, Ibáñez et al. [20] designed and evaluated an AR learning application to learn electromagnetism concepts. This application was made so that students can utilise them in making the manipulation of 3D shapes and emulation of the circuit elements possible. Each element is attached a fiducial marker to make it possible for recognition. Each element is associated with a particular learning material of problem solving for the manipulation of the students. Such aids 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 effect of AR on learning results, specifically those that require electromagnetic invisible forces understanding.

Temerinac et al. [21] have proposed a unified embedded engineering learning platform, which covers a complete learning process. The movement from hardware to software is the focus of the platform as it motivates the learning of the embedded systems while making knowledge related to the hardware design unavailable. The platform makes use of AR to visualise, simulate, and monitor of the invisible principle embedded in the fields of electronics. The system consisted of a magnifying glass, which had a transparent screen to display data extracted from a datasheet for the component in question.

Based on the literature, designing a structured AR learning environments that embed the learning process with the computational process has not been looked at, especially in providing a pedagogical explanation framework for real-time computational activities. Thus, this study tries to close the gap by evaluating the proposed PVM model and reveal its effectiveness in learning and teaching.

3. EXPERIMENTAL DESIGN

The experiment compared two educational approaches (PVM with AR application and the traditional approach of programming debugging). The experiment followed an experimental and group design using the type of approach (AR, traditional) as an independent variable. A between-subjects design was used to examine AR systems and the traditional approach. This was used to avoid knowledge transfer, which may result in identifying the effectiveness of the approach used more difficult. Twenty students from the School of Computer Science and Electronic Engineering (CSEE), University of Essex were invited to take part in the experiment. All students were postgraduate (masters and PhDs) with experience in programming. The students were divided into two groups, AR and TR, and each group had 10 students.

The experimental evaluation employed a learning activity that required students to design a behaviour based robot that follows a wall and avoid obstacles (Figure 1). Instead of making students create the robot's behaviour from scratch, students were given programming source code. The source code was designed to make the robot misbehave at runtime, and students were asked to solve the problem based on the assigned approach (AR or TR). This approach was chosen to reduce students' workload, as the aim of the experiment was to compare which approaches (AR or TR) assisted students in solving the learning task by analysing the robot's actions. The programming source code was written using an object-oriented style, which makes code easy to read, understand, and modify, especially for small programs. A complete list of robot classes was given and made available to use for modifying or replacing functions in the code. Both approaches (AR and TR) required students to use the Python programming language (https://www.python.org/) to edit and debug the given programming code for the mobile robot. This was based on the Geany text editor (https://www.geany.org/), which supports integrated development environments (IDE). The learning activity was based on Buzzboard modules specifically utilising Fortito's BuzzBot (an educational mobile robot) [22], [23]. In addition, Raspberry Pi[3] (https://www.raspberrypi.org/) was used to interface with BuzzBot, which allows communication and discovery of objects possible using an inter-integrated circuit bus (I2C).

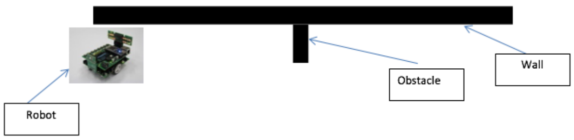

Figure 1 Learning activity test-bed

3.1 Learning Activity Design

The learning design of the activity followed the learning objects concept, where the learning activity decomposed into reasonable learning objects. Thus, the learning task for designing a behaviour-based robot that follows a wall and avoids obstacles was decomposed into four learning objects:

- 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

The decomposition of the learning activity into small learning objects was grounded on the hierarchical 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 2). These modules were then treated as individual learning objects. The reason was because of its modularity feature, as each one can incorporate with any other module to create robot behaviour: for example, maze escape. Modularity is a feature of object-oriented programming, where multiple modules are made first, and then integrated together to create a complete system [24]–[27]. Moreover, each learning object has its own learning objective and requirement, which all feeds into the main learning activity aim and objective.

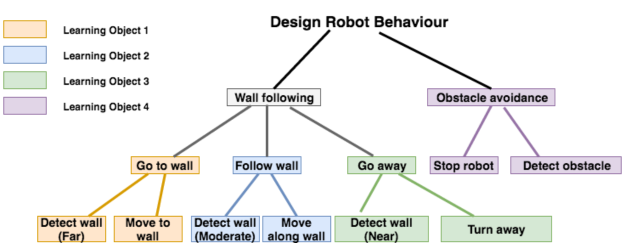

Figure 2 Hierarchical decomposition of wall following and obstacle avoidance

3.2 PVM with AR Approach

An AR application was developed to visualise and analyse the robot's actions and states and produce pedagogical feedback based on the learning activity. This approach followed the PVM model for structuring learning activity and analysing computational objects in terms of pedagogy. Once students debug or execute the source code, they can point their tablet's camera at the robot to overlay the robot's actions and behaviours in real-time. The robot's actions, behaviours and sensor values were updated in real-time on students' tablets. This gave students the ability to explore the robot's actions from low-level to high-level views that correspond to learning objects. The application provides students with four features that allow them to explore the construction of the robot's actions and behaviours as well as obtain instant feedback regarding the learning activity.

The system has four features. The first feature of the system is the data layer button. It is responsible for overlaying the robot's actions in real-time and learners perceive these data in the form of 2D representations. For example, infra-red sensor values represented visually in AR view in the form of 2D lines and 2D text that represent the actual value of the sensor with abstract meaning such as ***Object Far: 22***. All components' values and information in the robot, such as sensors and actuators, were represented visually in AR view and aligned roughly to the actual location of the component on the robot. The second feature is the aggregation layer button, whose main responsibility is to show the meaning of these data layer actions, such as ***Detect Wall***, in form of 2D representation: images. This layer allows interactivity where students can click on the aggregated object (behaviour) to reveal which actions belong to it. The third feature is the learning outcome button, which represents students' progress toward achieving the learning task. Progress feedback includes the learning activity (objects) state, achieved behaviours, and emotional expression clipart and indicates whether the learning objective has been met. The last feature is hiding activity, which allows learners to hide all behaviours and actions.

The PVM with AR system was developed using Unity3D (https://unity3d.com/), a cross-platform game engine for creating interactive 3D content that supports C# and JavaScript routines. AR implementation was deployed using the Vuforia Augmented Reality Software Development Kit (https://www.vuforia.com/) within Unity3D.

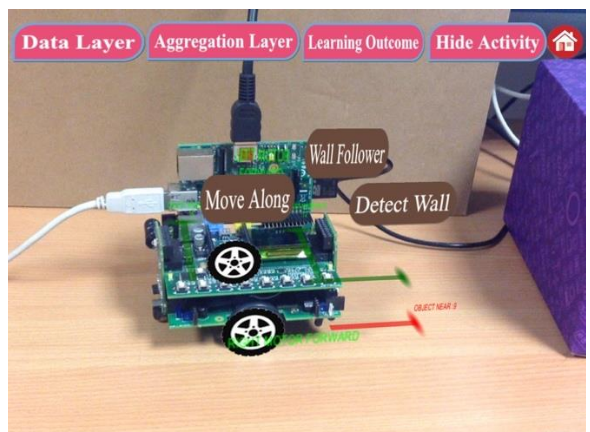

Figure 3 AR shows learning objects' behaviour with sub-modules in real-time

3.3 Traditional Approach

The traditional 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 in the tool's console. It was added to assist students in understanding robot program behaviour. It was noted by Ahmadzadeh et al. [28] that that 'print statement' is used in programming code as a debugging strategy to find errors. In addition, it helps in examining particular behaviours of programs and revealing value of interest [29]. Thus, during runtime, 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 representation was the same but the appearance was different: visualised in AR and log file text in the traditional approach. In both approaches, students were allowed to debug programming code many times until task learning objectives were achieved.

3.4 Measurement Instruments

Research instruments designed and selected for this experiment were aimed at understanding the impact of PVM with AR systems in solving and understanding embedded computing processes and communication. Quantitative data was studied, focusing on the learning effectiveness of the PVM model for analysing and exploring embedded computing learning tasks, especially on real-time systems. The first instrument was computational thinking, which was measured using pre and post-tests, which assess participants' ability to decompose the problem into smaller parts. One aspect of computational thinking is to use abstraction and decomposition for solving complex tasks [30]. Therefore, the pre and post-test were designed according to the hierarchical decomposition of behaviour based on a robot that follows a wall and avoids obstacles, as appears in Figure 2. Participants were giving hints, as some parts of the hierarchy were given. The second instrument was a knowledge test. Participants were provided with a short post-test to assess their learning outcomes based on the approach used (AR or TR). The test consisted of six questions, four questions required participants to indicate two things: what was the wrong action that the robot made and what is the correct action, whereas the other two questions examined participants' ability to decompose behaviour into actions. The last instrument was observation. 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 construct was the number of debugging time to solve the task. These were observed to measure the effectiveness of each approach (AR and TR) in terms of time and number of trials.

4. RESULTS

4.1 Exploratory Data Analysis

The first step in analysing the data from an experiment was to perform Exploratory Data Analysis in order to check assumptions, detect outliers, and properly select statistical techniques. As research variables were scale (continuous), Shapiro-Wilk was used to examine the distribution of the factors when the two learning applications were used.

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

Pre Thinking: the mean score of Pre Thinking was higher than the median score, suggesting a positively skewed distribution in the AR group. However, the Skewness value was within the normal limits. In the TR group, the mean was relatively higher than the median, suggesting a highly positively skewed distribution. This was confirmed by the high positive Skewness value exceeding the normal limits. This may affect the normality of the distribution. The normality tests showed that Pre Thinking was normally distributed in AR group, $p > .05$. However, in TR group, the normality test showed that Pre Thinking was deviant from normality, $P < .01$. Therefore, nonparametric tests should be used for Pre Thinking as they are robust to outliers.

Post Thinking: the mean score was relatively lower than the median, suggesting an asymmetric distribution of the AR sample. This suggested a non-normal distribution. In the TR sample, the mean score was largely higher than the median, suggesting a positively skewed distribution; skewness exceeds the normal limits. The normality tests showed that Post Thinking is deviant from normal distribution in AR sample, $p < .01$. In TR sample, the test showed that the Post Thinking distribution found to be deviant from normal, $p < .05$. This suggested that nonparametric tests should be used for Post Thinking.

Post Test: the mean score was exactly equal to the median, suggesting a symmetric distribution of data in the AR sample. Yet, the distribution was not deviant from normality. In the TR sample, the mean was slightly higher than the median and there is positive skewness statistic but within the normal limits. Therefore, Post Test was eligible for parametric tests.

4.2 Computational Thinking

The first research question was tested by performing the Sign test, which is a nonparametric or distribution free test. The Sign test, was used to determine whether there is a median difference between paired or matched observations. The test can be considered as an alternative to the paired-samples t-test or Wilcoxon signed-rank test when the distribution of differences between paired observations is neither normal nor symmetrical, respectively. Twenty students were tested to understand the students' 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 AR students group, the "Post Thinking" elicited a statistically significant median increase in computational thinking scores (50) compared to "Pre Thinking", $p = .002$. However, in the Traditional Approach students group, there was no statistically significant change in the median of computational thinking scores between the two trials; "Pre Thinking" and "Post Thinking".

Table 1 Sign Test Frequencies

		AR	TR
PostThinking - PreThinking	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 1 evaluates the number of positive, negative and tied paired differences to understand each student's (relative) response to the two trials. For AR group, the table shows that no students 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 Traditional Approach group, the table shows that no students had decreased scores, three students (total sample) had improved scores, and seven students witnessed no change (the "Ties" row) in their performance.

4.3 Learning Achievement

In order to answer the second question, two independent sample t-tests were performed to examine the significant difference in mean post-test scores between students who used PVM with AR and students who used the traditional approach. The test revealed that students who used PVM with AR had statistically significantly higher post-test scores (M = 10.50, SD = 1.650) compared to students who used the traditional approach (M= 6.40, SD = 2.119), t (18) = 4.828, p < .001. Therefore, students who use PVM with AR have gained better learning achievement than who use the conventional (traditional) approach.

4.4 Learning Performance

To answer question three, two independent sample t-tests were performed to test whether there is a statistically significant difference in mean time and mean debugging times between AR and TR groups. The independent sample t-tests revealed a significant difference in mean time and mean debugging times between students using AR and students using the traditional approach, p < .01. The test results are reported in Table 2. From the table, the mean time and mean debugging times for the AR students were significantly lower than those for the TR students, indicating that AR reduces the time for solving a learning activity task and with fewer trials.

Table 2 Independent Samples T Test Findings for Time and Debugging Times, grouped by Approach

	Group Statistics					
	AR		TR		T-Test	
	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 PVM with AR had statistically significantly lower time (M = 08:50, SD = 01:25 minutes: seconds) compared to students who used the traditional approach (M = 14:13, SD =03.44 minutes: seconds), t(18) = -4.252, p = .001.

Similarly, this study found that students who used PVM with AR had statistically significantly lower debugging times (M = 4.30, SD =.949 trials) compared to students who used the traditional approach (M = 8.10, SD =1.663 trials), t (18) = -6.275, p < .001.

5. DISCUSSION AND CONCLUSIONS

This study presented and evaluated an AR learning application that utilises the PVM explanation framework for constructing pedagogical meaningful information of computational objects. The study compared the AR-based application with its equivalent traditional approach to study its learning effectiveness on students' performance in solving problem tasks, especially with real-time systems. Both applications were designed to provide the same information and workflow capabilities. The main findings and their implications are discussed below.

With regards to the students' learning achievement in both groups, after conducting the post-test knowledge, the statistical results of this study reveal that students who used the PVM with AR application have gained a higher level of learning outcomes than those using the traditional approach. This result indicates the effectiveness of integrating the PVM framework with AR as its integrates abstract data in a pedagogical meaningful way. Thus, it allows learners to explore data in detail, which enriches their learning experience compared to the traditional approach. This finding supports research that found the use of AR technology enhances learning achievements [7], [20], [31].

Similarly, improving students' ability to understand computational objects behaviour and processes was found to be better with AR with the PVM group than with the traditional. The reason is because the AR with the PVM system has a feature that allows the learner to visualise the workflow of the data from low-level to high-level data. The system provides students with interactivity ability with the behaviour, which allows them to decompose it into parts to see which data belongs to it. This result supports the study that found AR systems improve learners' understanding of computer science concepts, such as event-handling[17].

The PVM with AR system enabled learners to find misbehaving actions faster and with less debugging time. It reduces the time taken to complete a complex task, such as solving behaviour-based robotics. As one of the AR affordances is to superimpose information on physical objects, this allows learners to not shift between real environments and physical objects compared to the traditional approach that isolated both worlds. Therefore, using PVM with AR enables learners to focus on the learning context. This finding is on track with the outcomes of the research studies in [17]–[19].

The limitations of the study are that first, it compared AR application with the conventional approach. This was because of the time limit; future work needs to compare AR with virtual environments or with AR with fewer PVM functionalities. Another limitation worth looking at is AR system usability evaluation. This is one of the aims for our future work to involve learners assessing the system. Qualitative data is not covered in this study, but we aim to gather users' opinion. Lastly, providing an authoring tool for teachers enables them to construct the whole learning activity from technical and pedagogical views.

Based on the finding of these results, it can be concluded that the PVM with AR framework was better than the traditional approach in improving learners' outcomes of real-time embedded computing systems.

6. REFERENCES

[1] C. Dede, 'Immersive Interfaces for Engagement and Learning', *Science*, vol. 323, no. 5910, pp. 66–69, Jan. 2009.

[2] L. Ştefan, 'Immersive Collaborative Environments for Teaching and Learning Traditional Design', *Procedia - Soc. Behav. Sci.*, vol. 51, pp. 1056–1060, Jan. 2012.

[3] L. Kerawalla, R. Luckin, S. Seljeflot, and A. Woolard, '"Making it real": exploring the potential of augmented reality for teaching primary school science', *Virtual Real.*, vol. 10, no. 3–4, pp. 163–174, Dec. 2006.

[4] K.-H. Cheng and C.-C. Tsai, 'Affordances of Augmented Reality in Science Learning: Suggestions for Future

Research', *J. Sci. Educ. Technol.*, vol. 22, no. 4, pp. 449–462, Aug. 2012.

[5] Y.-H. Suh, K.-W. Lee, M. Lee, H. Kim, and E.-S. Cho, '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)*, 2012, pp. 1541–1548.

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

[7] J. M. Andujar, A. Mejias, and M. A. Marquez, 'Augmented Reality for the Improvement of Remote Laboratories: An Augmented Remote Laboratory', *IEEE Trans. Educ.*, vol. 54, no. 3, pp. 492–500, Aug. 2011.

[8] C. Dede, M. C. Salzman, R. B. Loftin, and D. Sprague, 'Multisensory Immersion as a Modeling Environment for Learning Complex Scientific Concepts', in *Modeling and Simulation in Science and Mathematics Education*, W. Feurzeig and N. Roberts, Eds. Springer New York, 1999, pp. 282–319.

[9] D. P. Maloney, T. L. O'Kuma, C. J. Hieggelke, and A. V. Heuvelen, 'Surveying students' conceptual knowledge of electricity and magnetism', *Am. J. Phys.*, vol. 69, no. S1, pp. S12–S23, Jul. 2001.

[10] M. Alrashidi, V. Callaghan, and M. Gardner, 'An Object-Oriented Pedagogical Model for Mixed Reality Teaching and Learning', in *2014 International Conference on Intelligent Environments (IE)*, 2014, pp. 202–206.

[11] A. Pena-Rios, V. Callaghan, M. Gardner, and M. J. Alhaddad, 'Towards the Next Generation of Learning Environments: An InterReality Learning Portal and Model', in *Proceedings of the 2012 Eighth International Conference on Intelligent Environments*, Washington, DC, USA, 2012, pp. 267–274.

[12] J. Mateu, M. J. Lasala, and X. Alamán, 'Developing Mixed Reality Educational Applications: The Virtual Touch Toolkit', *Sensors*, vol. 15, no. 9, pp. 21760–21784, Aug. 2015.

[13] E. Freund, M. Schluse, and J. Rossmann, 'State oriented modeling as enabling technology for projective virtual reality', in *2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001. Proceedings*, 2001, vol. 4, pp. 1842–1847 vol.4.

[14] M. Daily, Y. Cho, K. Martin, and D. Payton, 'World embedded interfaces for human-robot interaction', in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences, 2003*, 2003, p. 6 pp.-.

[15] P. Amstutz and A. H. Fagg, 'Real time visualization of robot state with mobile virtual reality', in *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02*, 2002, vol. 1, pp. 241–247 vol.1.

[16] T. H. J. Collett and B. A. MacDonald, 'An Augmented Reality Debugging System for Mobile Robot Software Engineers', *J. Softw. Eng. Robot.*, vol. 1, no. 1, pp. 18–32, Jan. 2010.

[17] S. Magnenat, M. Ben-Ari, S. Klinger, and R. W. Sumner, 'Enhancing Robot Programming with Visual Feedback and Augmented Reality', in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, New York, NY, USA, 2015, pp. 153–158.

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

[19] J. Chan, T. Pondicherry, and P. Blikstein, 'LightUp: An Augmented, Learning Platform for Electronics', in *Proceedings of the 12th International Conference on Interaction Design and Children*, New York, NY, USA, 2013, pp. 491–494.

[20] M. B. Ibáñez, Á. Di Serio, D. Villarán, and C. Delgado Kloos, 'Experimenting with Electromagnetism Using Augmented Reality: Impact on Flow Student Experience and Educational Effectiveness', *Comput Educ*, vol. 71, pp. 1–13, Feb. 2014.

[21] M. Temerinac *et al.*, 'E2LP: A Unified Embedded Engineering Learning Platform', in *Proceedings of the 2013 Euromicro Conference on Digital System Design*, Washington, DC, USA, 2013, pp. 266–271.

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

[23] V. Callaghan, '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*, 2012, pp. 12–13.

[24] G. Agha, 'Concurrent Object-oriented Programming', *Commun ACM*, vol. 33, no. 9, pp. 125–141, Sep. 1990.

[25] D. A. Moon, 'Object-oriented Programming with Flavors', in *Conference Proceedings on Object-oriented Programming Systems, Languages and Applications*, New York, NY, USA, 1986, pp. 1–8.

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

[27] C. Szyperski, J. Bosch, and W. Weck, 'Component-Oriented Programming', in *Object-Oriented Technology ECOOP'99 Workshop Reader*, A. Moreira, Ed. Springer Berlin Heidelberg, 1999, pp. 184–192.

[28] M. Ahmadzadeh, D. Elliman, and C. Higgins, '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*, New York, NY, USA, 2005, pp. 84–88.

[29] X. Li and M. Flatt, 'Medic: Metaprogramming and Trace-oriented Debugging', in *Proceedings of the Workshop on Future Programming*, New York, NY, USA, 2015, pp. 7–14.

[30] J. M. Wing, 'Computational Thinking', *Commun ACM*, vol. 49, no. 3, pp. 33–35, Mar. 2006.

[31] H. Kaufmann and D. Schmalstieg, 'Mathematics and geometry education with collaborative augmented reality', *Comput. Graph.*, vol. 27, no. 3, pp. 339–345, Jun. 2003.