Presented at iLRN'15, Prague, Czech Republic, 15-17 July, 2015

Structured Learning Activities in Embedded Computing Using a Pedagogical Virtual Machine (PVM)

Malek ALRASHIDI[a,1], Michael GARDNER[a], Vic CALLAGHAN[a], and Jennifer B. ELLIOTT[b]
[a] *University of Essex, UK*
[b] *University of North Carolina, USA*

Abstract. In this work-in-progress paper, we investigate the creation of a technical framework (the Pedagogical Virtual Machine or PVM) which provides a layered analysis of the technical and pedagogical processes that are interacting together for any given learning activity (in the context of learning about embedded computing). We particularly focus on describing the structure of the pedagogical layer and how it handles the computational objects within it. This model is used as a means to create more effective tools for students who are studying embedded computing, which are typified by topics such as the internet of things, pervasive computing and robotics. This approach aims to enrich the experience for learners by constructing a meaningful view of the invisible things around us. Finally, we propose an embedded computing scenario that makes use of the PVM model.

Keywords. Pedagogical Virtual Machine, Object Oriented Programming, Augmented Reality, Mixed Reality, Pedagogical Framework, Learning Objects, Buzzboards.

1. Introduction

In Computer Science (CS) many of the computational processes are hidden inside the computer. As a human it is often difficult for us to see these processes as they are invisible. Often all that we can see is the final results from a computing process, with very little information about the underlying computational processes that caused the result. This is particularly true for embedded computing projects where often students will be constructing applications by assembling computing components which 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 is very little explanation for how the internal processes have operated to achieve the result. Often, the only way to discover this is by using some form of Augmented Reality (AR) or by using more traditional debugging tools to test the programs being used. The key challenge for this study is to construct a technical framework (PVM) which provides a layered analysis of the technical and pedagogical processes that are interacting together for any given learning activity (for learning about embedded computing). The user interface for this PVM will make use of AR to allow the student to visualise the static and dynamic information

---

[1] Corresponding Author: Malek Alrashidi, School of Computer Science & Electronic Engineering, The University of Essex, 5B.525, Wivenhoe Park, Colchester, CO4 3SQ, UK, E-mail: mqaalr@essex.ac.uk

within, and provide mechanisms for interacting with the underlying computing environment.

In the following section of this paper we start by describing some related work. Then, we extend the PVM model to describe the structure of the pedagogical layer and how it handles the computational objects within the pedagogical context inside the learning environment in section 3. The conclusion and future work will be outlined in section 4.

2. Related Work

Several previous studies have explored different approaches for supporting students learning embedded computing and robotics. For example, Lalonde et al [1] proposed a predictor for mobile robot programming called robot observability. This predictor is used for diagnostic transparency, which provides guidance about the incremental process of constructing and debugging robot programs. It is considered to be an important tool for students as it can be used for diagnosing a misbehaving robot. Students can build a tool that improves the performance of the predictor by identifying the evolution of the robots internal state through the use of audio feedback. For example, the robot can speak its actions and state its purpose. In addition, the authors reported in their survey that 86% of students believed that data logging and visual interfaces are very valuable debugging tools. This study has not considered the wider implications of using augmented reality as a visual interface that could be used to guide the students and reveal the internal communication processes that are happening inside the physical objects in real time.

In another study, Chan et al. [2] presented and evaluated the design of LightUp, an augmented reality learning platform for electronics. LightUp has several electronic components such as wire, bulb, motor and microcontroller. To form circuits, the learner needs to connect these components to each other magnetically. LightUp is implemented as a mobile application that provides an “informational lens” that uses computer recognition to identify electrical components, augmenting the image with visualisations, which makes invisible circuit behavior visible. The system was used to help children to learn, understand and construct circuits in real time via simulation. The drawback of this study is that they relied on extracting information using a simulation for the learning activity, and did not use physical objects.

3. Structuring Learning Activities using a PVM

In a previous paper [3] we presented a PVM model that uses an object oriented approach to combine concepts from computing together with a pedagogical model. The model consists of a data layer, aggregation layer, pedagogical layer and user interface layer. In this model, the pedagogical layer is responsible for managing and providing a structured description of the pedagogical context (i.e. for the learning activities). This layer maps data from the the computational (compound technical) objects that are provided from the lower layer (aggregation layer) to support the teaching and learning activities which are then used to guide the student using the user interface layer above. By correlating learning and computational objects we are able to make sense of a learning activity, providing guidance or feedback to the various learning stakeholders (e.g. teachers, learners) via the user interface layer. This layer consists of three main sublayers; the pedagogical context, the learning design description and an algorithmic state machine which are explained as follows:

- ***Algorithmic State Machine (ASM):*** this sublayer utilizes the ASM methodology for organizing the state flow of the compound objects and the state of the learning activity. 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 objects. Then, it maps the states to the related learning object steps, so that each state is actually a compound of two things; the step of the learning activity and the state of the compound object itself. Finally, we check each state to determine whether the learning outcomes of the learning task have been met or not.
- ***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) [4] . The main benefit of designing the learning activity in this way is to maximize their portability and re-usability. Also, it simplifies the structure of the learning activity, and it can be more easily modified. Thus, in this sublayer, we follow a well-known learning design specification called IMS (Instructional Management System) to define our learning object structure [5]. This allows the teacher/instructor to define the learning activities, the task steps, the learning objectives, the description of each task and the expected outcomes. Each 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 appropriate stage in the learning activity.
- ***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 [6]. 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 levels in the Bloom taxonomy. Using the PVM it should also be possible to make use of other structured pedagogical frameworks if this is required.

To demonstrate the structure of the learning activity, we present a learning activity in which the student will be asked to build a wall following robot. To accomplish the learning activity, the students must follow three phases:

- ***Introduction:*** this phase is where students are introduced to the learning activity, the requirements, learning objectives and goals. By the use of augmented reality, the learner can look at the physical object (robot) and reveal the objects services/operations available that they can make use of. This phase is related to knowledge and comprehension in Blooms taxonomy.
- ***Operation:*** Students will write their program using the Python programming language environment and then compile and load it onto the robot. This phase is related to the application level in Blooms taxonomy.
- ***Assessment:*** Once the program is loaded, the PVM model will listen to and reveal the processes being communicated inside the robot. For example, Figure 1 shows the hierarchy of the wall following task decomposition which contains several sub-processes. In practice, when learners debug/execute their program, they will see the final result which is the robot following the wall, but from the pedagogical perspective this does not tell them what causes this result. Thus, the PVM model starts to inspect the learners program, and will feed each layer with the required

information related to the learning activity. The data layer receives the objects data which are *detect-wall*, *move-to-wall*, *detect-continuation*, and *move-along-wall*. The aggregation layer takes the objects data and enhances the functionality by aggregating the data to provide higher value information to the pedagogical layers, for example, it takes *detect-wall* and *move-to- wall* to produce *go-to-wall* etc. Furthermore, the pedagogical layer maps the aggregated objects (*go-to-wall and follow-wall)* to the learning activity, *wall following*, and informs the learners via the user interface layer about their learning achievements and whether the wall following robot has accomplished the task or not. The learners use their tablet/smartphones as an augmented reality display to see the internal communication processes of the robot and obtain feedback and guidance based on each learning activity. This phase is related to the evaluation and analysis stages in Blooms taxonomy.

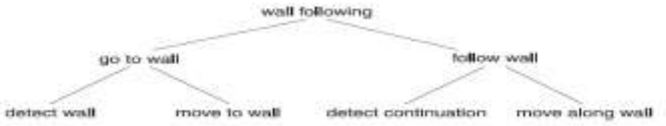

Figure 1. Hierarchical decomposition of wall following [7]

4. Conclusion

In this work-in-progress paper we described the structure of the pedagogical layer and how it handles the technical activities that are derived from use of the physical objects. It shows the realisation of the pedagogical layer, which combines learning and computational objects within a pedagogical context. We have demonstrated the workflow of the PVM model based on an embedded computing scenario. Clearly, this is still work-in-progress. We hope to demonstrate that the PVM model will have achieved its aim of seamlessly combining hardware, software and AR events within a seamless learning environment.

References

[1] J.-F. Lalonde, C. P. Hartley, 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.

[2] J. Chan, T. Pondicherry, and P. Blikstein, “LightUp: an augmented, learning platform for electronics,” 2013, pp. 491–494.

[3] 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.

[4] A. Koohang and K. Harman, *Learning Objects and Instructional Design*. Informing Science, 2007.

[5] I. G. . Consortium, “IMS learning design best practice and implementation guide, Final Specification.” Jan-2003.

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

[7] C. Scheier and R. Pfeifer, *Understanding Intelligence*. MIT Press, 2001.