Presented at Intelligent Environments 2014, Shanghai Jiaotong University, China, 2-4 July 2014

An Object-Oriented Pedagogical Model for Mixed Reality Teaching and Learning

Malek Alrashidi, Prof. Vic Callaghan, Dr. Michael Gardner
Department of Computer Science and Electronic Engineering
University of Essex, Wivenhoe Park, Colchester CO7 9QU, Essex, UK
{mqaalr, vic, mgardner}@essex.ac.uk

***Abstract*— the object-oriented (OO) paradigm is a well-known model that is used widely in the fields of both artificial intelligence (AI) and software engineering. OO models have been shown to be very powerful tools for dealing with complex human oriented activities. In the world of technology, object-oriented programming has been shown to be a very effective way of dealing with the complexity of programming advanced software applications. By bringing the object-oriented world of computing together with the object-oriented aspect of a pedagogical model, we extend our pedagogical virtual machine (PVM) model to be able to link human activities with technical activities inside learning environments. We propose a conceptual 4 layered architecture for our PVM and explain what each layer performs. Finally, the paper concludes by reviewing the main findings and discussing our future research plans.**

Keywords- Mixed-Reality; Augmented-Reality; Internet-of-Things; Cloud-of-Things; Learning Activity; Mobile Augmented Reality; Buzz-Boards; Object-Oriented Paradigm; Mobile Learning; eBooks

I. INTRODUCTION

A. Object-Oriented Paradigm

The object-oriented (OO) paradigm is a well-known model that is 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, behaviors and interactions with other objects [1][2][3][4]. Brad J Cox [5] 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, which 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 life [5]. Moreover, in the world of technology, object-oriented programming 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 behavior while shield its implementation details from other objects [5]. 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.

B. Rapid Prototyping System

One of the rapid prototyping development systems is BuzzBoard Fig 1. BuzzBoard is an educational technology toolkit that contains several software and hardware modules. There are over 30 BuzzBoard modules for developers to choose from; they can be found on the FortiTo website (www.FortiTo.com). This technology allows students and developers to create Internet-of-Things, Pervasive Computing and Intelligent Environments products. It leaves students more time to focus on creative design elements and programming systems [6]. It is used to help produce students' assignments and projects that are both interesting and simple, such as mobile robots, mp3 players, heart monitors, etc. However, an important architectural principle underlying BuzzBoards is modularity (both software and hardware), together with plug-and-play functionality (boards are identified to the system, and to each other, as they are plugged in), based on a shared bus (Buzz-Bus). This principle leads to a highly flexible and reconfigurable modular system that can be seen as an ideal infrastructure solution for rapid-prototyping and construction of pervasive and intelligent environments [6]. A key innovation arising from the use of BuzzBoard is that it provides an internal hardware network that provides both user driven events (e.g. plugging different boards together) that signal deep soft and hard behaviors. Both of these features play a key enabling role in the scheme, as they provide a way to get essential system information from the learning objects without disturbing the system, which most forms of instrumentation cause. Interestingly, according to Brad J Cox [5], when he started thinking about object-oriented programming he had a vision that everything in this world could be regarded as an object. Interestingly, he also thought of encapsulating hardware as a means to create worlds populated by mixing both hard and soft objects. Both Brad J Cox's thoughts and BuzzBoard have inspired us to think about hardware and software in computer systems as being objects.

In a previous paper [7] we produced a new concept that we referred to as a 'Pedagogical Virtual Machine' (PVM) that aims to cater for development and learning

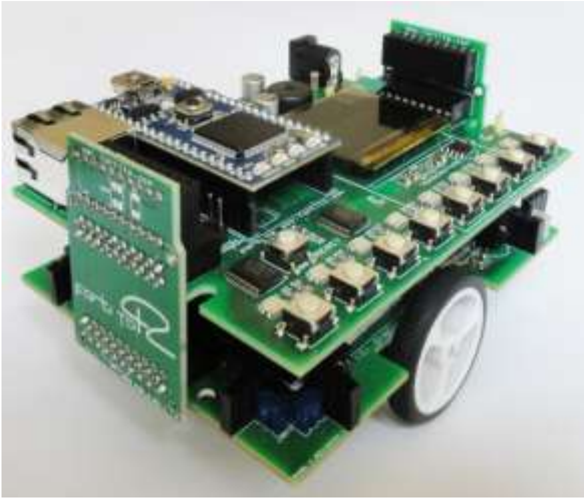

Figure 1 The BuzzBot (a modularised educational robot)

needs. The main purpose of the PVM is to act as a manager for revealing educational learning related functions and behavior in a computer. This model implies that all computer objects (hardware or software) contain data that represent the object's state and can be communicated with other objects.

The challenge of this research is how to bring the object-oriented world of computing together with the object-oriented aspect of a pedagogical model. In order to overcome this research challenge, the aim of this paper is to extend the Pedagogical Virtual Machine model to encompass object orientation and show how this can be integrated with object-oriented computing, thereby producing a novel and an effective mobile augmented-reality learning tool.

II. RELATED WORK

A. Data Modeling For Augmented Reality Application

Reitmayr et al. [8][9] proposed a 3-tire 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 tire hides data from presentation so that applications did not have to understand data details. Application types were derived from basic abstract types, such as SpatialObjectType and ObjectType that were predefined. Data storage and presentation layers were linked ensuring virtual representations are consistent with the monitored technology. This was achieved using an XML Object tree that was interpreted geometrically.

Nicklas et al. [10] 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 from. 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 delay due to the delegation mechanism and also separated underlying data operations from the client layer. In addition, multiple copies of the object on different servers caused data inconsistency.

Tonnis [11] 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 object; 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.

B. Augmented Reality Learning Platform

Chan et al. [12] presented and evaluated the design of LightUp, an augmented reality learning platform for electronics. LightUp consists of electronic components (e.g. wire, bulb, motor and microcontroller). These components are mounted on blocks that can be connected to each other magnetically to form circuits. LightUp is implemented as a mobile application that provides, what the author calls, 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. Similar to this, Ibáñez et al. [13] presented and evaluated an augmented reality learning application to learn electromagnetism concepts. This application was designed to be used by students and allowed them to manipulate 3D shapes and emulate the circuit elements. A fiducial marker was attached to each element to enable its recognition. A specific learning material, problem to solve and simulated properties were associated with each element for the students to manipulate. This helped the students to discover the behavior of the circuit or visualize the electromagnetic forces.

Temerinac et al. [14] have provided a unified embedded engineering learning platform, which covers a complete learning process. This serves as a general educational framework for future embedded system engineering. The focus of the platform is to move from hardware to software and it encourages the learning of embedded systems, but without giving knowledge that is related to the hardware design. The platform utilised augmented-reality as an interface for visualising, simulating and monitoring invisible principals in embedded electronic fields. The augmented-reality platform consisted of a magnifying glass, which had a transparent screen to display data extracted from a datasheet for the component in question.

Based on our literature review, there are no earlier published studies that have investigated the creation of a portable virtual machine engineered to extract pedagogical information and more specifically, the use of object-orientation to unify computational and pedagogical activities inside a learning environment, which are the subject of this research.

III. The Pedagogical Virtual Machine Model

Fig.2 shows the multi-layered architecture of the pedagogical virtual machine. This model consists of four main layers that range from low-level data collection to high-level data presentation. The pedagogical virtual machine (PVM) can be understood from two perspectives: an augmented reality view and an object-oriented view, which are explained in the following section.

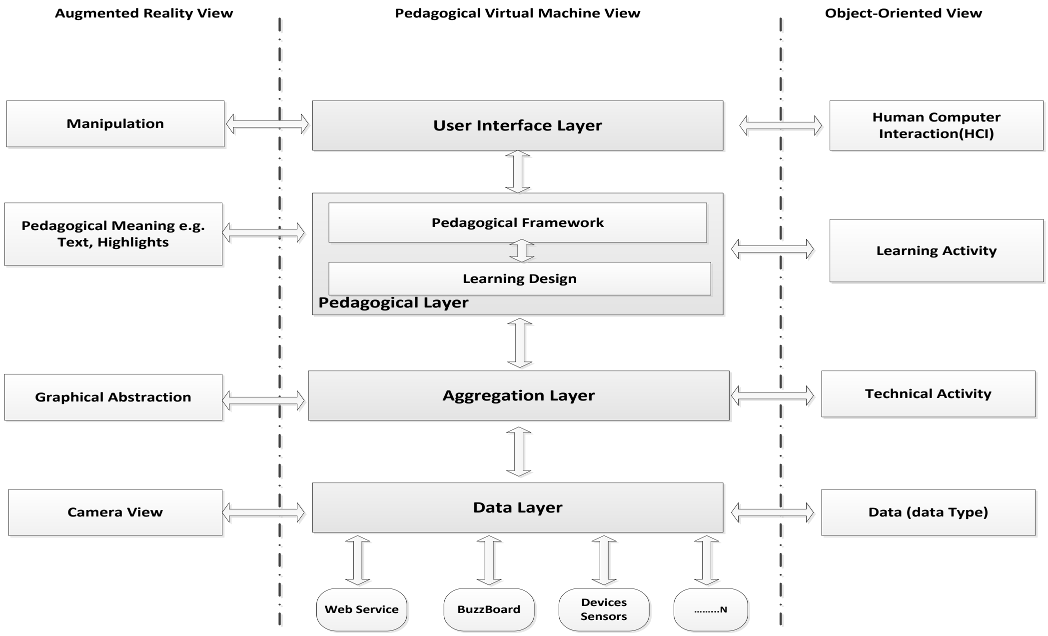

Figure 2 PVM Model

- **Data layer:** Ferscha et al. [15] stated "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". 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 [16]. This world is somewhat messy as it contains different types of devices that are interconnected in ad-hoc ways. Their forms are not structured as well, as each one has a different design to the other. In this layer, we are trying to achieve an object-oriented approach. The problem for the data layer is that nothing above this layer needs to know any details about these sensors at all. At this point, we are trying to present data in a simplified way. The most obvious way is to encapsulate sensors as an object, as we do not care what is inside it and we do not care what the sensors are, rather we only care about the messages. However, at this level, there are no methods and there is just raw data that are delivered/presented to the user, with which the user can make use; in this way we are using the notion of encapsulation. Thus, by encapsulating these data-objects', we are making them accessible to the layers above. This layer corresponds to the data type in the object-oriented view, and it may correspond to several forms in the augmented reality view, such as CameraView, sensor, actuators, events and data streaming.

- ***Aggregation layer:*** This takes the basic objects and enhances their functionality by aggregating their data to provide higher value information to the layers above. It groups data the lower level data-objects to provide higher value information to the layer above (pedagogical layer). Aggregating objects can be viewed as a form of inheritance as the low objects formed inherit characteristics (data) from their parent but more primitive objects (data layer objects). Thus, this layer enhances the functionality of lower level objects by creating a compound object. It is not a new object; rather it is a different object that is made up of things that are inherited (data object that is inherited in the layer below). So, the aggregation layer is intended to collect objects that are inherited from the data layer. The reason why we are aggregating is to close the understanding-gap by making the information that comes from the lower level of the model more similar to that need for learning in the pedagogical layer. In this sense the layer corresponds to technical activity from an OO viewpoint, where it packages the low level data and makes meaningful sense out of the data sequences (sequence of actions). It represents a graphical abstraction in AR view, which shows the object behavior in graphical representation.

- ***Pedagogical layer:*** This layer combines learning object and learning design to support teaching and learning activities. In this model the educational component is based on ‘learning objects’ (a well-established scheme for creating and delivering bite-sized lessons, frequently referred to as units of learning) and the technical activities are represented by computational objects (a well-established computing development paradigm). By correlating learning and computational objects we are able to make sense of the learning activity, providing guidance or feedback to the various learning stakeholders (e.g. teachers, learners or examiners) via the user interface layer. From object-oriented perspective, this layer utilises the principle of the OO schema for representing a network or society of objects although not as an explicit notation, but rather implicitly. The Augmented-Reality view provide a pedagogical meaning to the physical objects use in the student learning activities by overlaying information on the physical views, in the form of text, highlights, graphics etc.

- ***User interface layer:*** This layer provides an interface for teachers, students and examiners to the learning system. For the student/learner the interface guides him/her through the required sequence of actions need to achieve the learning goal as well as presenting him with supporting pedagogical information, such as information overlays. For the teacher, it enables him/her to set up the learning tasks as well as providing a record of how well the student has achieved the learning goal. This information can also be accessed by examiners or other moderators. The most visually striking feature of this layer are the image processing aspects connected to views derived from the device’s camera. For example, artefacts can be rendered, recognised and tracked in order to overlay virtual content in the user display, such as highlighted text, icons, video, graphical images, and 3D models. In addition, it allows learners to manipulate and interact with the tracked object.

Fig 3 illustrates the workings of the PVM Model. The learning activities tasks utilise a modular computing educational technology toolkit that contains several software and hardware modules called buzzboards. There are varieties of components such as BuzzBot, BuzzBus, BuzzBox, BuzzFree (a wireless IO link) and so on. In the example provided in this paper, we have used the BuzzBot (a modularised desktop robot) for the physical object. The BuzzBot can be programmed to perform different tasks such as line-following, light-seeking, and maze escapes (which are the classic robotic challenges) among others. The BuzzBot includes 8 IR Range Finders, 5 line following sensors, 2 light following sensors, Lithium-Ion Battery, 2 dual mode motors, motor load monitoring, quadrature motor feedback, and USB and external DC charging. The pedagogical theory is based on the Mayes and Fowler’ framework, which characterises the learning cycle into three stages, conceptualisation, construction and dialogue, which are well documented on [17].

In our implementation, the root of information in the Pedagogical Virtual Machine (PVM) is data derived from the BuzzBoards. These boards have a unique internal bus (BuzzBuss) that provides real-time data on the modules state and activity, without cause a computational overhead to the system. For example, if a student joins two modules together, or if a motor moved, that information would appear of the BuzzBuss. This data of raw digital data (e.g. bytes) is given a meaningful semantics (e.g. robot module plugged into the system, Obstacle detected, motor rotating clockwise etc), before being passed upwards in the Pedagogical Machine. The aggregation layer then, it receives this information, and analyses sequences or combinations of states, to deduce meaningful behaviors (compound sequences or states, without explicit pedagogical value). The pedagogical layer receives these compound activities and then correlates them with corresponding learning objects (Learning activities) in order to produce meaningful pedagogical achievements. Each learning-object is part of one stage of the Mayes and Fowler’ learning cycle. Finally, the overall representation and manipulation are done via the user interface.

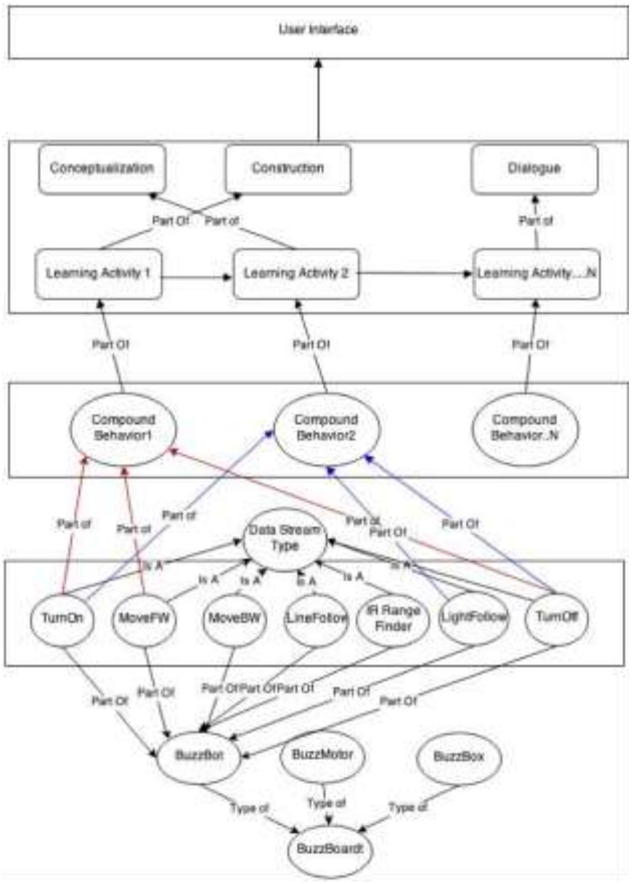

Figure 3 Example of PVM Workflow

IV. CONCLUSION

In this paper we proposed and explained a 4-layer Pedagogical Virtual Machine model. The model consists of a data layer, aggregation layer, pedagogical layer and user interface layer. We showed how these layers ultilise an object-oriented perspective of both computation and learning to create a novel approach to augmented reality and learning. In order to build and evaluate the performance and benefits of the system we have commenced building the system and have assembled the data layer (BuzzBoards) and HCI Layer (augmented reality). We still have some significant challenges ahead, most notably in the realisation of the pedagogical layer, which combines learning and computational objects.

Clearly, there is still much research remaining to be done, which we look forward to reporting at future events.

ACKNOWLEDGMENT

We are pleased to acknowledge FortiTo Ltd and Anasol Pena- Rios for her contributions of software and advice relating to the BuzzBot technology.

REFERENCES

[1] B. P. Pokkunuri, “Object Oriented Programming,” SIGPLAN Not, vol. 24, no. 11, pp. 96–101, Nov. 1989.

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

[3] B. J. Cox, “Message/Object Programming: An Evolutionary Change in Programming Technology,” IEEE Softw., vol. 1, no. 1, pp. 50–61, Jan. 1984.

[4] G. Booch, “Object-oriented development,” IEEE Trans. Softw. Eng., vol. SE-12, no. 2, pp. 211–221, Feb. 1986.

[5] B. J. Cox, Object Oriented Programming: An Evolutionary Approach. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1986.

[6] V. Callaghan, “Buzz-Boarding: Practical Support for Teaching Computing Based on the Internet-of-Things", 1st Annual Conference on the Aiming for Excellence in STEM Learning and Teaching, Imperial College, London, 12-13 April 2012.

[7] M. Alrashidi, V. Callaghan, M. Gardner, J. B. Elliott, “The Pedagogical Virtual Machine: Supporting Learning Computer Hardware and Software via Augmented Reality“, Immersive Education 2013 (iED’13), King’s College London, 28-29 November 2013

[8] G. Reitmayr and D. Schmalstieg, Data Management Strategies for Mobile Augmented Reality. 2003.

[9] D. Schmalstieg, G. Schall, D. Wagner, I. Barakonyi, G. Reitmayr, J. Newman, and F. Ledermann, “Managing Complex Augmented Reality Models,” IEEE Comput. Graph. Appl., vol. 27, no. 4, pp. 48–57, Jul. 2007.

[10] D. Nicklas, M. Großmann, T. Schwarz, S. Volz, and B. Mitschang, “A Model-Based, Open Architecture for Mobile, Spatially Aware Applications,” in Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases, London, UK, UK, 2001, pp. 117–135.

[11] D. Schmalstieg and G. Hesina, “Distributed applications for collaborative augmented reality,” in IEEE Virtual Reality, 2002. Proceedings, 2002, pp. 59–66.

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

[13] 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.

[14] M. Temerinac, I. Kastelan, K. Skala, B. M. Rogina, L. Reindl, F. Souvestre, M. Anastassova, R. Szewczyk, J. Piwinski, J. R. L. Benito, E. A. Gonzalez, N. Teslic, V. Sruk, and M. Barak, “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.

[15] A. Ferscha and M. Keller, “Real time inspection of hidden worlds,” in Seventh IEEE International Symposium on Distributed Simulation and Real-Time Applications, 2003. Proceedings, 2003, pp. 51–58.

[16] 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.

[17] J.T.MayesandC.J.Fowler,“Learning technology and usability: a framework for understanding courseware,” Interact. Comput., vol. 11, no. 5, pp. 485–497, May 1999.