Design and Development of a Component-Based System for Virtual Patients in the Virtual World of Second Life®

Maria Toro-Troconis
Imperial College London, Faculty of Medicine, London, UK
Email: m.toro@imperial.ac.uk

Ashwin Kamat
Tata Interactive Systems, Mumbai, India
Email: ashwink@tatainteractive.com

Martyn R Partridge
Imperial College London, Faculty of Medicine, London, UK
Email: m.partridge@imperial.ac.uk

***Abstract*—This paper presents the development of a Component-Based System structured as a distributed three-tier architecture, enabling monitoring and information visualisation of application activity as well as presentation of feedback to learners via a Heads-Up-Display (HUD) in the virtual world of Second Life®. The activities follow a game-based learning approach and take place in a Respiratory Ward in Second Life®, where learners interact with virtual patients receiving *intrinsic* feedback about their diagnosis, investigations and treatments. The proposed architecture developed by the authors, consists of different virtual patient components that provide the relevant personal and clinical data to the clinical scenario; a data availability model that enables the sequencing and progressive disclosure of a virtual patient identifying triggers and scaffolding information, and an activity model which encodes the activities available and how the learner will be able to engage with the virtual patients.**

***Index Terms*— Component-based system, three-tier architecture, Second Life®, virtual world, virtual patients**

I. Introduction

Virtual worlds are 3D spaces in which users interact and meet using a virtual representation of their persona in the form of 'Avatars'. The first virtual worlds appeared in the early 1970s with the characteristics of modern virtual worlds. However, all the broadband and multimedia capabilities currently offered, were not available at the time. In the 1980s Massively Multiplayer Online Games (MMOGs), such as Everquest, Guild Wars and the World of Warcraft were developed and widely used as online 3D game-platform environments.

According to Reference [10] and [4], these environments offer interactive, engaging and rich collaborative learning experiences. However, it is difficult to differentiate between virtual worlds, games and social networking worlds. For the purpose of this study, we decided to concentrate on open-ended social virtual worlds such as: Second Life®, Active Worlds Educational Universe, and There.com. These open-ended virtual worlds allow the use of video and sound for live chat and events as well as the use of online authoring tools for the construction of 3D assets within the worlds.

Second Life® was selected as the virtual world for this study in 2007 due to its growing use in UK Higher Education institutions [7], as well as for its potential for developing the Component-Based System intended.

The Component-Based System intended at the time, and later developed, manages the way learners access a series of virtual patients in a virtual hospital created in Second Life®. The design follows a game-based learning approach in the virtual world using triggers and scaffolding information to engage the learner in the process of diagnosing and treating virtual patients [18], [19], [20]. Figure 1 shows the Respiratory Ward in the virtual hospital.

Figure 1. Respiratory Ward in Second Life®

doi:10.4304/jetwi.3.4.308-316

A demonstration of the activity in Second Life® can be accessed on YouTube at: http://tinyurl.com/mwpm2r and from the Faculty of Medicine e-learning website at: http://tinyurl.com/65myeck

II. Component-Based System Application Architecture in Second Life®

Component-Based Systems (CBSs) are built using independently produced and tested components, offering a flexible alternative between standard and customised software [16]. The most common structure of client/server model in a system, assumes two discrete participants. This model is called "two-tier system", which means the application logic resides within the client or server or shared between the two. If the application logic resides separately from the data and the user interface, the system turns into a "three-tier architecture". According to Reference [8], in an ideal three-tier system, the user interface and the data reside separately from all the application logic. However, this does not happen frequently; usually select portions of the application logic are the responsibility of the client and/or server, and the bulk of the application logic is in the middle tier.

In the words of Reference [8], processes become more robust in the three-tier architecture, being a more advanced and flexible approach than the traditional two-tier model. New levels of autonomy are given to the application logic processes due to the separation of the application logic from the client and server. For example, in a standard web application following a three-tier system, the first tier is the user interface which reflects the interpretation of HTML by a browser. In the middle tier, resides the embedded components, such as: Java applets and ActiveXs, displayed by the browser. The final tear is represented by the data served from a web server. This is quite often a database driven system or groupware system.

Three-tier systems provide increased advantages, such as scalability: logic components can be shared across multiple clients. These systems are easier to maintain since the business logic is separate from the User Interface (UI). The UI can also be changed without the risk of altering by mistake the business logic. At the same time, the business logic can also be shared with applications with different UIs because it resides in its own tier.

In the Second Life® environment, the Second Life® client is installed on the user's machine. The Second Life® client communicates with the Second Life® engine for catering the user responses and rendering the media assets for the user. The http Request call is established through Linden Scripting Language (LSL) from the Second Life® simulation environment in order to access the external web world. The LSL compiler takes the active role in validating the script statement before execution.

Figure 2 shows the three tier-architecture implementation of web application for Imperial College virtual campus, interaction and call management between the Second Life® virtual world environment and the web world environment.

The *web world environment,* which host the web application for the Imperial College virtual campus consists of the three-tier architecture based on Model View Controller (MVC) design pattern. The MVC architecture allows for a clean separation of business logic, data, and presentation/interface logic. The model represents application data and the business rules that govern access and modification of this data. The model object notifies the views when it changes or its state is modified. A view presents the data represented by the model in a way that's targeted at a specific type of client/interface. Views are decoupled from the business logic, so that these can be modified without any affects to the business logic or the model.

The controller performs all the authentication process, before providing a user access to the application. It is made up of many components responsible for taking data posted in an HTTP request or made by a function call and converting it into an event to update the model data. Controller is also responsible for sequencing the logical flow of actions required to extract relevant information from the model layer using one or more component invocation.

The MVC architecture is implemented through Struts Framework using Java 2 Platform Enterprise Edition (J2EE) technology. The core of the Struts framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, Resource Bundles, and XML, as well as various open source packages. Following is a brief description of the various tiers in the 3-tier architecture of web application designed & developed for virtual medical campus:

Tier 1 (web server): static content such as HTMLs, media elements (JPEG, GIFs, JavaScript) are directly served from the web server. The web server forwards requests for server side components such as Java Servlets, JavaServer Pages (JSPs), and other Java classes (Action classes, Delegates, Service Locater) to the Servlet runner. Web application designed & developed for virtual medical campus uses Apache HTTP server, a well known web server which can serve HTML/ JPG and other static content but at the same time has bridge component for forwarding the JSP/Servlet request to relevant J2EE application server on a different port.

Tier 2 (application server): the application server is responsible for running the J2EE environment over which the Struts framework resides. It generally has a servlet container like tomcat at the minimum though commercial heavy weight J2EE application server like Websphere/Weblogic can be used. Application server is also responsible for deployment, object pooling (Activation and Passivation) of session beans, and transaction support to the session beans. The persistence layer is implemented on the application server.

Tier 3 (database): the third tier is the database, which is the central repository of all data that the system generates and queries to create the reports. Web application designed & developed for virtual medical campus uses MySQL 5.0 as database repository to store data tracked during various activities conducted by users in the virtual campus.

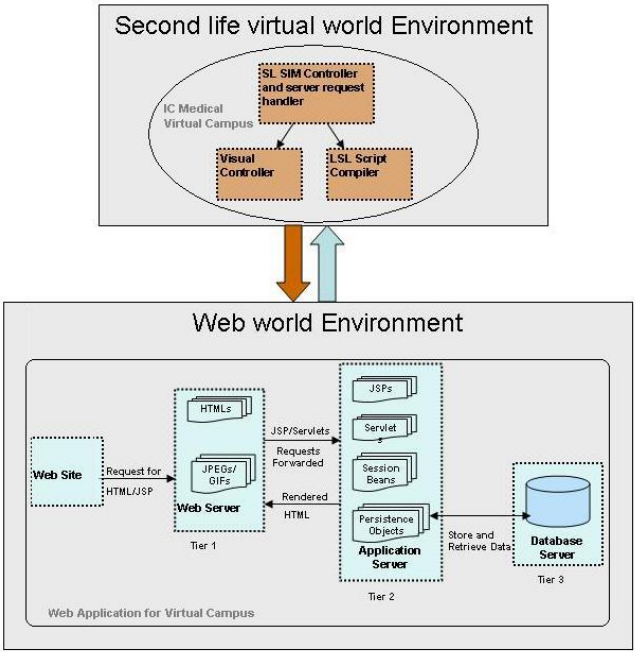

Figure 2. Three-tier architecture implementation of web application

A. Web application development

A website service was developed as part of the web application development. The website aims to record the Second Life® activity produced by the user in the database external to Second Life® and then create a presentation layer for when the data is pulled into the different report formats. This website service helps academics and administrators keep track of the activity generated by users when accessing the virtual patients in the virtual hospital.

Figure 3 and 4 show an example of the Report module. This module offers several reporting options. For example, all the registrations to the virtual hospital till date or during specific dates can be displayed; the activities attempted by all users or specific users can be generated. This helps academics keep track of the progress made by the students when accessing the virtual patients. Investigations needed for the diagnosis of virtual patients have a cost in Linden Dollar (L$). This was designed purposely in order to ensure students think about their budgets when ordering investigations for their virtual patients. Detailed reports on all the money spent by all users or by specific users help manage the process of refunding these payments.

Figure 3. Report module for activities mapped for patient

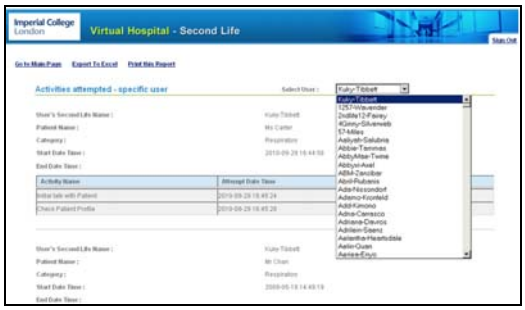

Figure 4. Report module: activities attempted per user

B. Virtual Patients' Functionality

In order to provide a virtual environment that aims to simulate a learning process, it is important to offer all the functionality and the necessary tools and applications to its users in order to offer an efficient space for experimentation, communication and collaboration [3].

In the Respiratory Ward in Second Life®, users have the ability to interact with virtual patients suffering from different Respiratory conditions, such as: Asthma, Chronic Obstructive Pulmonary Disease, Lung Cancer, etc. These activities offer the requirements and steps that the learning process would offer in a real hospital ward scenario. Figure 5 shows a workflow with the steps required to complete a virtual patient.

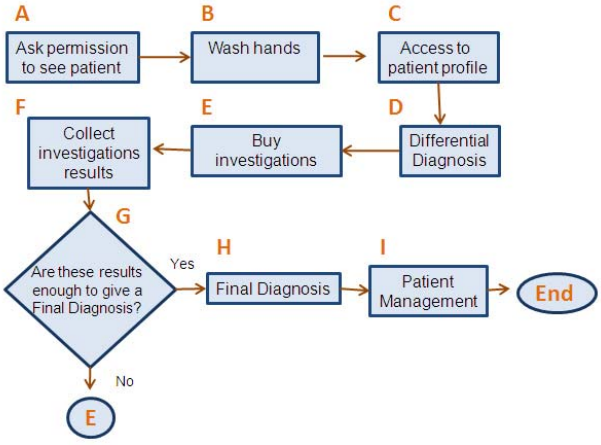

Figure 5. Virtual Patient Workflow

(A) The user finds the 'Virtual Patient Panel' by the entrance of the virtual patient room. The user has to ask permission to see the patient (See Figure 6). Feedback is delivered via the Second Life® feedback panel (See Figure 7).

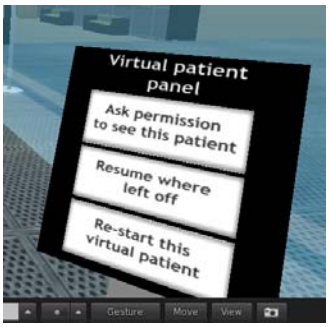

Figure 6. Virtual Patient Panel

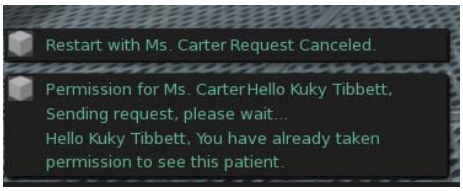

Figure 7. Feedback after asking permission to see the patient

(B) The user must wash his/her hands before talking to the patient. For this, the user needs to 'Touch' the wash basin located in the room (See Figure 8). Feedback is delivered letting the user know his/her hands are clean (See Figure 9).

Figure 8. Wash basin panel

Figure 9. Feedback after user washes her hands

(C) The user can now access all the relevant information available by the patient, e.g. Patient Profile, breath sounds, etc. (See Figure 10). Patient Profile feedback is shown in Figure 11.

Figure 10. Patient Profile panel

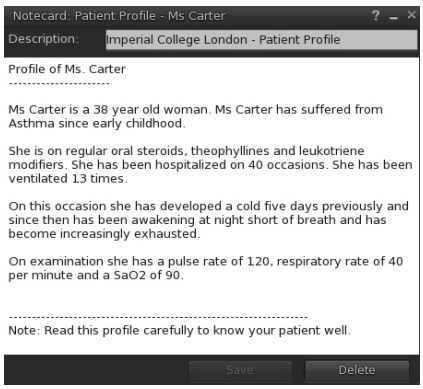

Figure 11. Patient Profile feedback

(D) After accessing the Patient Profile, the user can proceed to make a Differential Diagnosis (See Figure 12 and Figure 13).

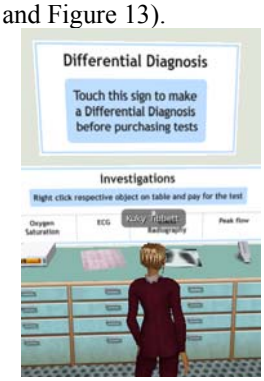

Figure 12. Differential Diagnosis panel

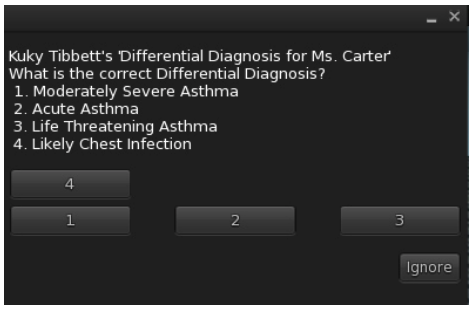

Figure 13. Differential Diagnosis question

(E) Then the Investigations required are purchased (See Figure 14 and Figure 15).

Figure 14. Selecting the Investigation to be purchased (Oxygen Saturation test in this case)

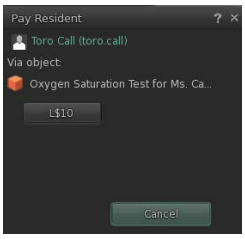

Figure 15. Feedback is received after purchasing the Investigation

(F) The results of the Investigations purchased are then collected from the relevant departments which are adjacent to the virtual patient's area in the Respiratory Ward (See Figure 16 and Figure 17).

Figure 16. Collecting Investigations from the Lung Function Department

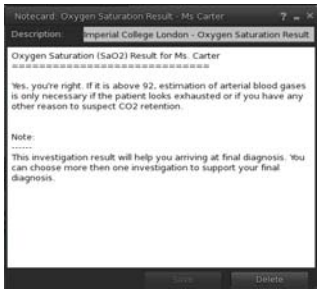

Figure 17. Feedback is received after collecting the investigation

(G) If the Investigations purchased are not sufficient to give a Final Diagnosis, the user will not be able to proceed and further Investigations will need to be obtained.

(H) Once, at least, one correct Investigation is purchased and collected for the Diagnosis of that condition, the user can then proceed with the Final Diagnosis (See Figure 18 and Figure 19).

Figure 18. Final Diagnosis panel

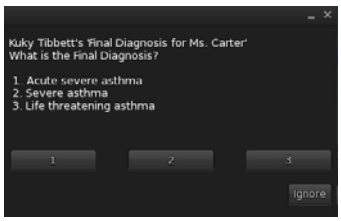

Figure 19. Final Diagnosis question

(I) After receiving feedback about the Final Diagnosis, the user is invited to visit the Patient Management area where more information is provided about the management of that condition (See Figure 20).

Figure 20. Patient Management

C. Feedback provided via Heads-Up Display (HUD)

User performance can be greatly influenced by an active feedback system [2]. The authors recognised the need to maintain users informed of their actions with different levels of feedback, in order to provide a more rewarding user's experience. It is extremely important to provide clear explanations that keep learners informed of their actions [9].

According to Reference [12] and [14], users have limitations in relation to memory, sometimes forgetting what they have done and not being able to perceive subtle differences. On the other hand, Reference [12] also highlights users' strengths in relation to being able to retrieve relevant information rapidly by ways of storing

similar patterns, as well as being able to process visual information quickly.

Any CBS has different types of users at different stages during the life cycle of the application. In this particular CBS, three distinct categories have been identified according to the role they play within the application. The first user is the designer/programmer, who is in charge of the end-user application. The designer/programmer designs and develops the components and is also responsible for providing technical support. The second user is the learner for whom the application has been created. The last user is the academic, who is in charge of the learning experience, tracking learners' responses.

The needs of each of these users are very different. Therefore, this CBS has been designed in order to accommodate the needs of these 3 levels of users:

The *designer/programmer* needs technical feedback from the application to record data pertaining to the user activities and present it in the form of reports. This is achieved in two parts viz. In-World feedback, and Over-Net feedback. The In-world feedback is gathered through listener object coded using LSL scripts and deposited in the virtual campus to listen In-world communication continuously on the Second Life®'s private communication channels. The listener once received the In-world feedback converts immediately into suitable Over-Net feedback. This Over-Net feedback then gets posted using http server request handler at the web service running at the web application developed for virtual campus. The web service secures the data received through Over-Net feedback into database repository for future usage and presentation.

The *learner* or end-user needs to receive feedback about their actions associated with the tasks being carried out. In order to provide this level of feedback to learners, a Heads-Up-Display (HUD) has been developed. Its functionality will be explained in detailed in this section.

The *academic* needs to keep track of the activities carried out by the learners. The Report Module explained before, provides this information to academics allowing them to produce reports about learners' attempts per virtual patient, their actions on the patient, the money spent on investigations, their scores, etc.

> *"The goal of system design in many applications is to give operators sufficient information about the current status and activities, so that, when intervention is necessary, they have the knowledge and the capacity to perform correctly, even under partial failures."*
> (Reference [9], p.p 85)

The process of designing user interfaces for this CBS started with a task analysis and a detailed specification of the end users. A HUD or console was designed to keep learners informed of their progress when accessing the different virtual patients at the virtual hospital (See Figure 21).

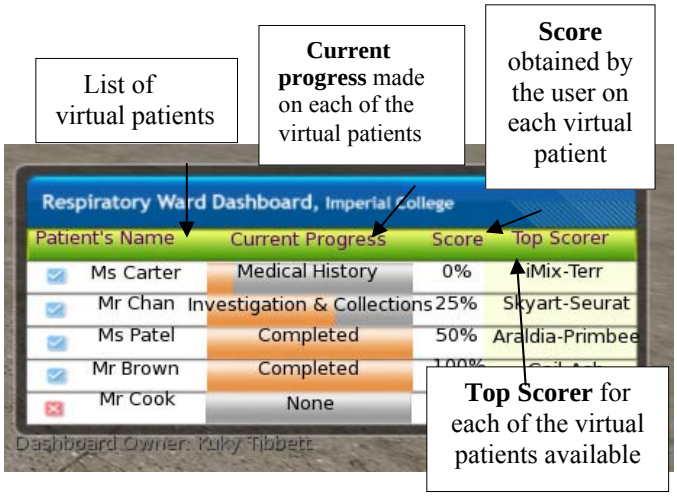

Figure 21. Virtual Patients HUD

At run time environment, the virtual patients HUD provides information, which is dynamically updated as the user accesses the virtual patients.

The list of patients available at the Respiratory Ward is presented on the left hand side. An icon with a tick '√' indicates the user has started treating that patient. An icon with a cross 'X' indicates the user has not attempted to treat the patient. The current progress indicates how far the user is in the evaluation of the patient.

The score section displays the score obtained by the user per virtual patient. The score is calculated depending on the correct answers obtained when giving a Differential Diagnosis and a Final Diagnosis. The top scorer section provides the name of the user who has obtained the best score for that virtual patient.

The HUD can be downloaded from the Respiratory Ward in the virtual hospital (See Figure 22). Once it has been downloaded, it is available in the inventory of the user. The user will need to access the HUD in her/his inventory and 'wear' it so it is available as an icon; not intruding, but always offering the possibility of obtaining feedback at any time (See Figure 23).

Figure 22. Downloading the HUD from the Respiratory Ward

Figure 23. HUD available as an icon

The HUD satisfies feedback requirements providing information about *rewards*, such as: *rank*, *mechanical* and *victory rewards. Rank rewards* let users know about their progress offering a sense of progression and completion. *Mechanical rewards* make users feel their actions have an effect; this is represented by the 'Top Scorer' section. The 'Top Scorer' section also offers the learner *Victory rewards*, giving the learner the opportunity to defeat other participants [20].

III. Comparison with PIVOTE

It is important to mention and discuss another web based authoring environment for virtual patients developed in Second Life® which provides similar functionality to the one developed at Imperial College London and discussed in this paper.

PIVOTE is an authoring system for virtual worlds that consists of 7 main elements: Exercise Definition, PIVOTE Editor, PIVOTE Player, Web Interface, Virtual World Interface, Virtual World Objects, and Student Performance Data [1]. Apart from the virtual world element the rest of the elements are developed for web using Perl CGI::Application framework.

'CGI::Application aims to provide a framework for re-usable code that is based on the run mode concept, but in a way that avoids a lot of spaghetti code. Because it is Object Oriented (OO), it can easily be extended and enhanced. CGI::Application encourages the adoption of a Model-View-Controller (MVC) framework. It includes the logic and rules that the application understands and enforces. The view is simply the user interface, and by default CGI::Application supports HTML::Template. Finally, the controller mediates between the model and the view. It takes input, passes it to the model components and invokes the view components as appropriate.'
(Reference [6], p.p 2)

The architecture of the virtual patients in the virtual Respiratory Ward in Second Life® also uses MVC architecture implemented through Struts Framework using Java 2 Platform Enterprise Edition (J2EE) technology. The core of the Struts framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, Resource Bundles, and XML, as well as various open source packages. The main advantage of the Struts framework is its Centralized File-Based Configuration.

'Struts values/mapping are represented in XML or property files. This loose coupling means that many changes can be made without modifying or recompiling Java code, and that wholesale changes can be made by editing a single file. This approach also lets Java and Web developers focus on their specific tasks (implementing business logic, presenting certain values to clients, etc.) without needing to know about the overall system layout.'
(Reference [17], p. 1)

The PIVOTE infrastructure uses the Medbiquitous Virtual Patient (MVP) XML format to store definition of the exercise [5].

'The MVP specification also allows for the storage of meta-data including a description of the exercise and parameters that can be used to filter a set of exercise (e.g. in the medical case by age/sex/complaint of patient).'
(Reference [1], p.p 176-177)

However, this requires users to have knowledge of XML standards for authoring the virtual world exercise. Although the authoring of MVP cases is supported by PIVOTE text editor, a web-based application, the interface still demands the basic understanding of XML terminology from the author. The Editor also lets authors export the XML data to their hard-drive, and import XML from other authors and it supports user accounts, and sharing and editing privileges so that multiple authors can work through the same instance of the editor program.

The Imperial College virtual world web application has simple and user friendly GUI and provides an easy to use system even for the non-computer savvy users. However, the authoring environment only allows *linear virtual patients* to be authored.

A key advantage presented by PIVOTE in comparison to the Imperial College virtual application is evident in the functionality presented by the PIVOTE Editor. The PIVOTE editor can author branching cases allowing the creation of required nodes, linking them to asset nodes, assigning variables, etc. The Imperial College virtual application Editor only allows for *linear virtual patients* authoring always allowing the same path to be followed with the same links to asset nodes and variables.

The case and user tracking and performance data is stored by the Imperial College virtual application using MySQL RDBMS package. A Relational Database Management System (RDBMS) is simple and easy for users to understand and use. RDBMSs provide data access using a natural structure and organisation of the data. DBMSs feature maintenance utilities that provide database administrators with tools to easily maintain, test, repair and back up the databases housed in the system.

Database queries can search any column for matching entries using "Structured Query Language" (SQL). The SQL syntax is simple, and the use of standard English language keywords and phrases makes it easy to use [15].

The Imperial College virtual world web application provides a set of reports based on powerful SQL syntax to track the performance & usage data of the users captured from in-world activities. The data presented by the reports can be printed or exported into Microsoft Excel format for further analysis.

IV. Conclusion

Access to patients suffering from different respiratory conditions by undergraduate medical students is presently being challenged by time constraints as well as patients' availability during students' rotations. Feedback from consultants and clinicians is also limited. The Respiratory Ward developed in the virtual world of Second Life®, provides an interactive representation of the activities carried out in real Respiratory Wards offering medical students the opportunity to explore in a safe environment, the possibility of seeing and managing different respiratory conditions.

This paper has discussed the development of a CBS as a three-tier architecture, which accommodated the needs of the users identified within the CBS: *designer/programmer*, *learner* and *academics*.

The segmentation of the application into the three-tier architecture has proved to provide the functionality required to display the user interface and perform the main logic of the application and storage and retrieval of data in an effective and efficient manner.

The activities implemented have followed a game-based learning approach which has not been discussed in this paper but in other publications [18], [19], [20].

According to the different evaluations carried out as part of this study, assessing learners' attitudes towards this type of learning using virtual patients in Second Life®, learners find the *linear virtual patient* activities not challenging enough [21]. Although a game-based learning approach was followed in the design of the virtual patient activities and interfaces [18], the development of more challenging '*emergent narratives*' and '*responsive environment*' [11], were not achieved. On the other hand, the PIVOTE authoring environment discussed in this paper allows the development of more challenging learning experiences using branching narratives which learners seem to enjoy and learn from [1].

The CBS model presented in this paper can still be recommended to simulate other learning experiences that fit a linear approach targeting low end Cognitive skills. The same level of users' feedback can be replicated in other fields such as: business and engineering, where instead of virtual patients, other components relevant to the learning experience can be implemented. A similar data availability model enabling the sequencing and progressive disclosure of information may be implemented as well.

All the programming code used in this CBS has been released as open source, licensed under a Creative Commons Attribution-Non Commercial 3.0 License, in order to stimulate other interested parties in the development of similar applications in the virtual world of Second Life®.

Acknowledgment

The authors wish to thank the Faculty Education Office (Medicine) at Imperial College London for the support received during this study.

References

[1] Burden, D., Jinman, A. (2011), "Web Based Authoring for Virtual Worlds Using PIVOTE", in Vincenti, G. & Braman, J. (eds.), *Multi-User Virtual Environments for the Classroom*, USA: IGI Global, pp. 170-189.

[2] Chan, H.C., Wei, K.K. and Siau, K.L. (1995), "The effect of a database feedback system on user performance". *Behaviour and Information Technology*, 14(3), pp. 152-162.

[3] Chitarro L. And Ranon R. (2007), "Web3D Technologies in Learning Education and Training: Motivations, Issues, Opportunities", *Computer Journal of ELSEVIER Computers and Education*, 3(49), pp. 3-18

[4] De Freitas, S. (2008), "Serious Virtual Worlds: A Scoping Study. Prepared for the JISC E-Learning Programme". http://www.jisc.ac.uk/media/documents/publications/seriousvirtualworldsv1.pdf Accessed 10 January 2011.

[5] Ellaway, R., Poulton, T., Fors, U., McGee, J., & Albright, S. (2008). "Building a virtual patient commons". *Medical Teacher*, 30(2), pp. 170–174. doi:10.1080/01421590701874074

[6] Horne, D. (2005), "CGI::Application: A Simple, Extensible Web Framework". http://blogs.sitepoint.com/cgi-application/ Accessed 10 January 2011.

[7] Kirriemuir, J. (2009), "Virtual World Activity in UK Universities and Colleges. Snapshop #7: Winter 2009". Eduserv, http://virtualworldwatch.net/wordpress/wp-content/uploads/2009/12/Snapshot-7.pdf. Accessed 10 January 2011.

[8] Lewandowski, S.M. (1998), "Frameworks for component-based client/server computing", *ACM Computing Surveys*, 30(1), pp. 3-27

[9] Lewis, C. (1986), "Understanding what's happening in system interactions", in Norman, D.A. and Draper, S. W. (eds.), *User Centred System Design. New Perspectives on Human Computer Interaction*, New Jersey: Lawrence Erlbaum Associates, pp. 171-186.

[10] Livingstone, D. (2007), "Learning support in multi-user virtual environments", In *Proceedings of the European Conference on Game-Based Learning,* University of Paisley, Scotland, 25–26 October.

[11] Murray, J. (1997), *Hamlet on the holodeck: the future of narrative in cyberspace*, Cambridge: Massachucetts, MIT Press.

[12] Olsen, J.R. (1987), "Cognitive analysis of people's use of software", in Carroll, J.M. (ed.), *Interfacing Thought: Cognitive Aspects of Human Computer-Interaction*, Cambridge, MA: MIT Press, pp. 260-293.

[13] Shneiderman, B. (1998), *Designing the user interface: strategies for effective human-computer-interaction*, 3rd

ed., United States of America: Addison Wesley Longman, Inc.
[14] Smith, Sid L. And Mosier, Jane N. (1986), 'Guidelines fro Designing User Interface Software', Report ESD-TR-86-278, Electronic Division, MITRE Corporation, Bedford, MA. Available from National Technical Information Service, Spring-field, VA.
[15] Soltesz, L., (1999), "The Advantages of a Relational Database Management System". http://www.ehow.com/list_6121487_advantages-relational-database-management-system.html Accessed 10 January 2011.
[16] Szyperski, C. (1999), *Component Software. Beyond Object Oriented Programming*. Addison Wesley, Harlow, England.
[17] TheServerSide.com (2004) September 13, 'What is the advantage of using Struts, other than MVC?', *Web tier: servlets, JSP, Web frameworks Forum* [Online], http://www.theserverside.com/discussions/thread.tss?thread_id=28667 Accessed 27 May 2011.
[18] Toro-Troconis, M., Mellström, U., Partridge, MR., Meeran, K., Barrett, M., Higham, J. (2008), "Designing game-based learning activities for virtual patients in Second Life®", *Journal of Cyber Therapy & Rehabilitation,* 1:3, pp. 227–239.
[19] Toro-Troconis, M., Mellström, U. (2010), "Game-based learning in Second Life®. Do Gender and Age make a Difference?", *Journal of Gaming and Virtual Worlds* 1:2, 53-76, doi: 10.1386/jgvw.2.1.53_1
[20] Toro-Troconis, M., Meeran, K., Higham, J., Mellstrom, U. and Partridge, M. (2010), 'Design and Delivery of Game-based Learning for Virtual Patients in Second Life®: Initial Findings', in A. Peachey, J. Gillen, D. Livingstone and S. Robbins (eds), *Researching Learning in Virtual Worlds*. London: Springer, pp. 111-138
[21] Toro-Troconis, M., Roberts, NJ, Smith, SF, Partridge, MR. (2010), 'Students' perceptions about delivery of game-based learning for virtual patients in Second Life', in Zagalo, N., Morgado, L. and Boa-Ventura, A. (eds.), *Virtual Worlds and Metaverse Platforms: New Communication and Identity Paradigms*. USA: IGI Global, (In press).

Maria Toro-Troconis is the E-learning Strategy and Development Manager at the Faculty of Medicine, Imperial College London. Her main role is to support the development and delivery of the Faculty's e-learning strategy.

Maria has extensive experience working in e-learning having developed several e-learning programmes for various UK Universities. Maria's background is in Computer Science and Human Factors. Her research interest is game-based learning and virtual worlds. Her key skills include instructional design, coordination across distributed teams, business analysis and project management.

She also has an in depth knowledge of International Learning Standards and their implementation across platforms.

For recent publications by Maria, please visit: http://www.imperial.ac.uk/medicine/teaching/elearning/publications/

Ashwin Kamat is Senior Technical Architect working with Tata Interactive Systems, Mumbai. He is the key architect behind the company's initiative on discovering learning possibilities through immersive, engaging and rich collaborative virtual world environment. During this initiative he has evaluated multiple virtual worlds such as Croquet®, Forterra®, Active World®, and Second Life®. He has successfully executed multiple projects using Second Life® virtual environment for Imperial college London, UK and Federation of American Scientists, US.

Ashwin is graduated from Mumbai University and has overall 18 years of IT industry experience. He has experience working in e-learning domain from last six year with Tata Interactive Systems, Mumbai and he is currently involved in the designing of the lightweight Learning Management System (LMS).

Martyn R Partridge is Professor of Respiratory Medicine, Imperial College London, and Senior Vice Dean, the Imperial College Nanyang Medical School, Singapore. His research interests are in evaluating the delivery of healthcare to those with respiratory illnesses. This includes evaluation of service enhancements such as the organisation of specialist consultation services, the use of lay educators, integration of respiratory healthcare, telephone consultations, Electronic asthma and COPD action plans, and methods of enhancing communication between both patients and doctors, and specialists and General Practitioners. Professor Partridge is past President of the British Thoracic Society (BTS), previous Chief Medical Adviser to Asthma UK, a previous member of the GINA (Global Initiative for Asthma) Executive and Chairman of their Dissemination Committee and he currently chairs the UK Department of Health (DH) Asthma Steering Group.