Aye, aye Captain: Voice active bots in a maritime MUVE

Todd Cochrane
Digital Technologies, Nelson
Marlborough Institute of
Technology
Todd.Cochrane@nmit.ac.nz

Niki Davis
University of Canterbury e-
Learning Lab
Niki.Davis@Canterbury.ac.nz

ABSTRACT

The paper presents an aspect of design based research into the design of 3D Multi User Virtual Environment (MUVE) to support authentic language learning on the bridge of a ship. Voice active Non-Player Characters (NPCs) "voice bots" were included in the design process to increase the authenticity of helm experiences, including helm commands. Using an Agile approach, development of a client based "voice bot" avatar was undertaken to complement human controlled avatar voice communication. While initial analysis indicated a relatively short development phase, the need to work around proprietary voice software led to a number of tasks that became distractingly large and exploratory on the programmer's Agile backlog list. A decision to focus on the first principle in the Agile Manifesto constrained the solutions that were developed, which led to a working system for creating voice bots.

Keywords: Computer Assisted Language Learning (CALL), Agile software development, anti-patterns, Multi User Virtual Environment (MUVE), non-player characters, voice bots, Design Based Research, educational gaming

1. INTRODUCTION

A design based research project into the design of 3D Multi User Virtual Environment (MUVE) includes the creation of a MUVE in which students preparing to become deck officers on ships will be able to practice authentic language learning on the simulated bridge of a ship. The purpose in the larger project is to better understand the multidisciplinary processes of design from both the perspectives of education and Informatics. This paper presents the results of a section of that research, which arose during the design of a voice for an avatar undertaken to complement a Non Playing Character (NPC), or robot, which is being designed to narrate within each scenario.

Following a description of the educational background, the overall design of this research puts the development of a "voice bot" into the multidisciplinary approach of the research. The actual development of the voice for an NPC is described. That is followed by reflection on the application of the Agile technique (Agile Manifesto, 2001).

2. BACKGROUND

A 3D MUVE for learning standard marine communication phrases (SMCP) (International Maritime Orgnization [IMO], 2002) for ship's bridge personnel is being developed as part of research into the design and evaluation of 3D MUVEs in vocational education (Cochrane, Davis, & Mackie, 2015).

Claudia and Alexandru-Florin (2013) describe the SMCP as a "sort of survival kit" (p. 45) through which officers are trained in the use of the protocols encounter less communication difficulties in "managing safety-related situations, performing navigational duties, and organising or supervising cargo operations" (Claudia & Alexandru-Florin, 2013, p. 45). As identified by collaborating maritime tutors, learning maritime communication with appropriate responses to orders is usually embedded in other training situations.

Maritime tutors are familiar with training that uses simulators. For example, IMO (2003) describes training for Watch Keeping in ship handling simulators, with the objective that trainees "will have gained experience in handling ships under various conditions and will make a more effective contribution to the bridge team during ship manoeuvring in normal and emergency situations" (p. 3). Hence, maritime tutors are willing to support development of a 3D MUVE that provides trainees with the facility to practice communication phrases. The tutors selected SMCP Standard Wheel Orders (IMO 2002, p. 61) for the helmsman, as a scenario for trainees to practice in the MUVE, because the trainee can practice as a helmsman while learning how to respond to orders.

In this context, voice communication within the 3D MUVE was selected over text chat. Communication with voice provides trainees with more realistic situations in which they can listen to orders and respond in their own words. Since voice in a 3D MUVE can be recorded and played back to the trainee, the trainee can listen to his or her own pronunciation and practice by comparing that with phrases as pronounced by the trainers. Given the need to practice listening to orders and produce correct pronunciation of orders, it was decided that development of an NPC who talks was required. The talking NPC would provide an opportunity for trainees to return to the 3D MUVE to listen to the SMCP as pronounced in a standard voice. The trainees could then record their own voice responses to the orders and review their pronunciation.

3. THE DESIGN BASED RESEARCH

Development of a NPC voice bot is part of the authors' educational design based research (Plomp, 2007) that includes Agile sprints to produce MUVEs in vocational education (Cochrane, Davis & Mackey, 2015). The research is divided into three phases which are semantically similar to those described by McKenney and van den Akker (2005). At present the research is in phase one. The phases and the overall structure of an educational design based research process are depicted in Figure 1.

This quality assured paper appeared at the 6th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2015) and the 28h Annual Conference of the National Advisory Committee on Computing Qualifications, Queenstown, New Zealand, October 6-9, 2015. Michael Verhaart, Amit Sarkar, Rosemarie Tomlinson and Emre Erturk (Eds)

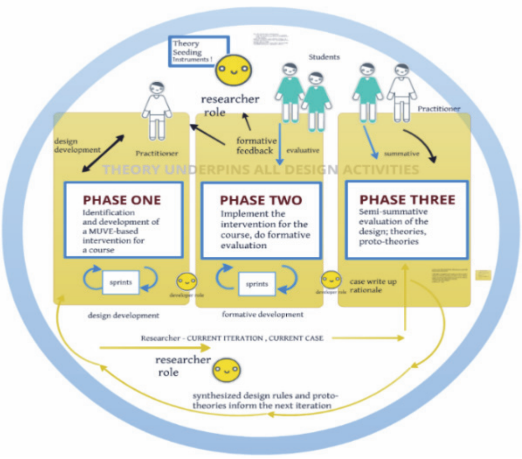
Theory Seeding Instruments !
researcher role
Students
Practitioner
design development
Practitioner
formative feedback
evaluative
summative
THEORY UNDERPINS ALL DESIGN ACTIVITIES
PHASE ONE
Identification and development of a MUVE-based intervention for a course
PHASE TWO
Implement the intervention for the course, do formative evaluation
PHASE THREE
Semi-summative evaluation of the design; theories, proto-theories
sprints
developer role
sprints
developer role
case write up rationale
design development
formative development
Researcher - CURRENT ITERATION , CURRENT CASE
researcher role
synthesized design rules and proto-theories inform the next iteration

Figure 1. An educational design based research process, based on figure 2 (Cochrane, Davis, & Mackey, 2015)

3.1 Three phases

During phase one the researcher works with education practitioners and MUVE developers to identify and develop a 3D MUVE based intervention for a course. There are three types of participants: practitioners, developers and researchers. A researcher may act as a systems analyst or guide a developer in a systems analyst role to identify and collate requirements. Practitioners can be those who may run a class or experts in the domain. Development of the 3D MUVE follows an Agile approach in which stories are identified, backlog lists are produced, sprints determined and undertaken, and burn-down is monitored.

In phase two the intervention is implemented during a course. Formative evaluation is undertaken by practitioners and students. Adjustments to the intervention are fed into the agile process resulting in a set of sprints to produce these.

Phase three undertakes a semi-summative evaluation of the intervention and its design. Reflection of the application of the theories and rationale from the development and implementation of the intervention are used as a case from which further design may proceed with newly formed proto-theories or matching existing theories are identified as useful in the next iteration. In this case, trainees receive adaptive training during the simulation, and a de-brief session after the simulation. Evidence of their performance is collected and integrated in an authentic way by Master tutors, who through the simulation system provide the same complexity as if training was undertaken while on a vessel at sea. This authentic approach supports "positive feedback" adaptive learning design and assessment as described by You (1993). Further, the sensitivity of the design of the system as experienced in the development of this learning environment also reflects the application of chaos theory to learning system design presented by You (1993).

Work in all phases is underpinned by educational theory. For the first iteration of the project, Legitimate Peripheral Participation (LPP) (Lave & Wenger, 1991) and Technological Pedagogical Content Knowledge (Mishra & Koehler, 2006) are used as theoretic filters in the design of tools for determining requirements of the 3D MUVE. LPP provides a model of authentic learning in which participants are actively moving from the periphery of a discipline to becoming strong practicing members. LPP is developed in situ, as a participant develops within the discipline. An example of situated learning through LPP is how a person becomes a Quarter Master on a vessel (Lave & Wenger, 1991). As described by Lave and Wenger (1991) and confirmed by Maritime studies tutors, trainees incrementally take more responsibility as they develop their skills. Trainees start by reading the charts that have been laid out on a chart table, to determine the vessel's current location and direction, they then move on to more critical activities such as actually plotting the vessel's course. This in situ learning is typical of training undertaken by Maritime trainees.

Documentation from the software development during phases one and two is included as part of "systematic reflection and documentation to produce the theories or design principles ... as the scientific yield from the research." (Plomp, 2007, p. 15).

3.2 Current outcomes from phase one

After considering alternative 3D MUVE platforms, OpenSimulator (2014) (OpenSim) was selected. This platform is free, has an open source server code base, and a wide community of developers for both the server and 3D Virtual World viewers. Voice communication is facilitated by donation of a free voice service to all OpenSim installations by a major VoIP company. An OpenSim server was set up on a cloud server and after a series of upgrades it is currently stable and running all the time.

A preliminary scale model of a ship's bridge in the 3D MUVE was constructed. In a visit by the researcher and a 3D designer to the actual ship, images of the interior of the bridge were collected to enhance the authenticity of the 3D MUVE simulation. During the visit the ship's master – the captain of the vessel – confirmed the need for developing communication of orders and correct responses to orders.

A 3D MUVE simulation of a ship with a detailed but static ship's bridge was developed by the researcher and the 3D designer. This simulation includes voice communication for Avatars run by "human players". That was tested by one of the maritime tutors in a laboratory classroom and by the maritime captain and tutor in charge of training with the ship handling simulator at his desk. A meeting with the current ship handling simulation tutor, the maritime communications tutor and the program area manager identified SMCP Standard Wheel Orders (IMO 2002, p. 61) for the helmsman, as a useful scenario for trainees to practice in the MUVE.

The present 3D MUVE can be used for running a classroom intervention for training of SMCP as long as tutors are willing to treat the 3D MUVE as a role-playing environment in which students or tutors act as the officer in charge of the watch. Discussion with tutors determined that trainees should to be able to practice and listen to the SMCP as many times as they needed while in the 3D MUVE. However, if students role play as the officer in charge of the watch they may not provide clear and accurate pronunciation of the SMCPs. Therefore, alternative approaches were considered. For example, although it would be possible to pay an actor, that would be too expensive. Developing a "voice bot" – an NPC that can talk, provides for situations where the trainees can practice, listening and pronunciation in the 3D MUVE.

4. DEVELOPING THE VOICE BOT

Following an Agile approach a new 'story' for a "talking robot" avatar was added to the backlog list, which was maintained using the Trello (2015) tool.

4.1 Work that was part of, and ongoing during the sprint

Techniques for developing NPCs for OpenSim were investigated and tested during the sprint. According to OpenSim (2013a) there are three techniques for operating an NPC or "bot", which are:

1. Express OpenSim NPCs using extensions provided in the scripting system, Linden Scripting Language (LSL), as described in detail in OpenSim (2013b).
2. Create a purpose built Region Module, as a plug-in module expressed as a DLL, and written in C# code. This approach makes use of the same OpenSim code used by the scripting system, but allows extension in a plug-in module to be loaded by the OpenSim server.
3. Client side approaches, which involve creating an equivalent to a virtual world viewer that runs an avatar. For example the libOpenMetaverse code base from the Open Metaverse Foundation (2015) or code derived from libOpenMetaverse such as the Radegast Metaverse Client (Khalifa, 2015).

The first two techniques do not include voice communication as a built in function.

The first technique would require development of voice communication functions for the main scripting system. Using the first technique was rejected since it would lead to extensive development work.

The second technique has been used to develop OpenSim voice communication through Whisper (Canham, 2010), that provided a bridge to the Mumble VoIP server (Mumble, 2014) from the OpenSim server. This presented an opportunity to create a talking bot. A Mumble open source VoIP server was installed on the server along with recompiled Whisper modules. However, since Whisper was developed for earlier versions of OpenSim and Mumble, the Whisper code required substantial refactoring and development. In addition, use of Whisper would have required a reinstallation of the voice process with the client 3D World Viewers, which was also out of date and would have needed substantial refactoring and development. Requiring users of the system to install a different voice process would make the MUVE more difficult to use. The second technique was therefore rejected, especially after recognition that a new Region module would also need to be developed.

The third technique seemed to be relatively easy to achieve using the Radegast Metaverse Client (Radegast) for the visually impaired (Khalifa, 2015), because it includes sound based speech feedback for the operator. A preliminary analysis of the Radegast code indicated that developing voice should require a simple adjustment.

The Radegast code base was selected and development of the voice bot was undertaken. The actual voice communication code was working through a proprietary process that belongs to the VoIP company. That process works directly with the device drivers of the machine it is on. Hence voice sound was not being directed to and from the voice server through the Radegast code. To circumvent the proprietary software, the default device driver of the machine was replaced with a software (virtual) cable that redirects sound to and from the audio devices. That led to the need to locate a free and open source virtual cable from VB-AUDIO Software (2015).

This virtual cable system provided an adequate but not particularly "smart" solution. However, the researcher and developer compiled a test device driver for the machine, with the idea that this could be used to redirect sound for the voice bot to and from the proprietary process. At this point the sprint task list already felt unwieldy, details were being lost. Hence, developing a bespoke device driver was considered, investigated and left for potential future work.

The approach taken was to develop a separate voice scripting application from which voice is directed to an instance of Radegast. The instance of Radegast logs into the 3D MUVE then connects to the voice service through the proprietary process. However, sound is directed to and from the voice scripting application. In order to run voice bots on demand, the Radegast 3D Viewer and the open source device driver redirection system are installed into a virtual machine that can be copied and run to create as many voice bots as required.

4.2 Work tracked during the sprint

Developing a voice bot was to be undertaken in one sprint. The sprint was set up for work over a few days with tasks as follows:

- Run libmetaverse Voice Script
- Download and compile bot code system on server
- Run Voice on New Simland

These were achieved over a four-hour period in one day. However, the actual running of a voice bot did not take place until a number of sprint days later when a better understanding of the requirements had been achieved. The entry "*Download and compile bot code system on server*" was tested and marked off has having been done. However, the voice code for the bot had not actually been implemented. This was mentioned in comments in the Trello activity list: "*Found voicetest.exe, uses voiceManager object", and "looks like the Voice manager can play a sound file, but the method is commented 'appears' to not work*".

Further testing prompted the possibility that the voice server was not running, producing an activity entry "stopped New Simland, will run again as DigitalOcean command console, make sure voice is working before testing voice script". This led to a reconfiguration of the server from a HyperGrid mode to a StandAlone mode before voice worked, producing this activity entry: "In order to run voice on New Simland - had to put it back to a standalone server".

A series of tasks were added to the sprint backlog over a period of hours are shown in table 1.

While these items were being added to the Backlog list they seemed manageable. However, this list was quickly developing into an 'explosion' in the number of relatively large exploratory tasks to be added to the Backlog list; even adding those to that recording became tedious and the recording on the list ceased, despite the occurrence of more items. After reflection overnight the highest priority principle on the Agile Manifesto surfaced, which is "satisfy the customer through early and continuous delivery of valuable software". The decision was then taken to constrain the search space by focusing on the delivery of valuable software with less cost. Hence, entry ten, being high priority and less work, became the focus of the development that led to the resolution of a framework for the voice bot.

Table 1: Sprint Backlog, in order of arrival

Entry	Priority 1 high to 5 low	Description	Perceive-d amount of work (Large Medium or Small)
1	1	Compile a version of Radegast on local machine , upload and test in virtual box (via soundboxbox in google cloud)	Large
2	3	Backup virtual box vm and test restore on local machine	Large
3	1	Implement recording system, use snooper.se	Large
4	1	Test the bot avatar recording voice it hears	Medium
5	1	Put Virtualbox in a Windows 2008 server Vm running in google cloud, test trail voice	Large
6	3	Try running VM from home see what it looks like in the lab	Small
7	2	Check if the system already works in the lab, leave the VM running , log into lab , see if dois dois is talking	Small
8	2	Create a Portable Virtual Box implementation test it in the Lab	Large
9	2	Run VM version in Lab to see if the "cloudy" body problem disappears	Small
10	1	Run a Radegast instance, with Virtual Cable	Medium

5. REFLECTION

The increase in effort required to follow and test multiple development pathways, led to the thought that the work could be encapsulated as an agile anti-pattern. Anti-patterns described by Eloranta, Koskimies, Mikkonen and Vuorinen (2013), were considered, for example: “Too long sprint”, “Unordered product backlog”, “Semi-functional teams”. These anti-patterns identify useful points at which a procedural behaviour could be remedied. Therefore an anti-pattern like “Too many large and costly backlog items” in the present project would be remedied by adjusting selecting the highest priority and least costly items over other items.

However, Nerur and Balijepally (2007) in part of a theoretical reflection on agile development methodology, argue for a “holographic organization”(p. 82) metaphor for agile software development, where all parts of the organization are able to complete the project, in the same way every part of a broken holograph can produce the recorded image. They assert:

"The emerging agile philosophy heralds a new epistemology of software development. Its value depends largely on an organization’s ability to nurture learning, teamwork, self-organization, and personal empowerment. Responsiveness and flexibility are achieved through a “heterarchy” characterized by self-organizing teams whose members collaborate, improvise according to problem context, and use their ingenuity to solve problems" (Nerur & Balijepally, 2007, p 82).

Taking this perspective presents a way to think about the positive effect that came about at a critical turning point in the sprint for the development of the “voice bot”. The work was multi-dimensional, not a linear process, in which many parts of the work can lead to an outcome. The search produced an attitude that was moderated by applying the highest priority principle pervasively in the process. In this way, while an anti-pattern remedy is a useful outcome when reflecting on the process, further reflection identified that applying the principle did indeed produced the desired effect.

6. SUMMARY AND CONCLUSION

The processes of educational design are under researched, particularly the complexity of designing MUVE for vocational education. This paper presents for the first time a detailed account of a critical event in multidisciplinary research into MUVE design, approached through Design Based Research complemented with Agile software development.

A systematic method is applied in the development of a MUVE that situates learning and teaching SMCP by maritime trainees and tutors. Selection of requirements for the system is filtered using educational theory. An Agile development approach is applied for both 3D design and in development of the required software systems.

The development of an environment for creating and running “voice bots” led to increasingly complex software requirements. For the researcher and developer a number of potential solutions were perceived as high priority and requiring a large amount of effort. Focusing on Agile principles helped to reduce development work and led to a new way of thinking about the effect of Agile development as applied in this context.

7. ACKNOWLEDGEMENTS

We are grateful for the invaluable participation of tutoring staff from the NMIT Maritime Studies group who provide for authenticity in the development in the Ship’s Bridge: Captain Katherine Walker (Class 1 Unlimited - Master Mariner Foreign Going) manager of the New Zealand Institute of Maritime Studies, Captain Chris Lordan (Class 1 Unlimited - Master Mariner Foreign Going) who runs a range of courses including Standards of Training, Certification and Watch Keeping, and Officer of the Watch; also tutor Roger Wincer. Finally we acknowledge 3D MUVE designer Warren Masterson.

8. REFERENCES

Agile Manifesto. (2001). Principles behind the Agile Manifesto. Retrieved from http://agilemanifesto.org/principles.html

Canham, N. (2010). Whisper – New Voice Solution for OpenSimHypergrid Business. Retrieved from http://www.hypergridbusiness.com/2010/08/whisper-new-voice-solution-for-opensim/

Claudia, G. P., & Alexandru-Florin, R. (2013). Standard Maritime Vocabulary Phrases. *Analele Universitatii Maritime Constanta*, 14(19), 45-49.

Cochrane, T., Davis, N.E., & Mackey, J. (2015). Design Based Research with AGILE sprints to produce MUVES in Vocational Education. (291-313) In K. Terry & A. Cheney (eds.) *Utilizing Virtual and Personal Learning Environments for Optimal Learning.* Hershey, PA: IGI Global.

Eloranta, V.-P., Koskimies, K., Mikkonen, T., & Vuorinen, J. (2013). Scrum Anti-Patterns -- An Empirical Study (pp. 503–510). *Asia Pacific Software Engineering Conference , IEEE*. http://doi.org/10.1109/APSEC.2013.72

IMO. (2002). *IMO Standard Marine Communication Phrases*. International Maritime Organization.

IMO. (2003). *Model Course 1.22 Ships Simulator and Bridge Teamwork* (2002 Edition). International Maritime Organization.

Khalifa, L. (2015). Radegast Metaverse Client · Lightweight client for connecting to Second Life and OpenSim based virtual worlds. Retrieved from http://radegast.org/wp/

Lave, J. & Wenger, E.(1991). *Situated Learning Legitimate peripheral participation*. Cambridge : Cambridge University Press.

McKenney, S., & van den Akker, J. (2005). Computer-based support for curriculum designers: A Case Developmental Research. *Education Technology Research and Development*, 53(2), 41–66.

Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *The Teachers College Record*, 108(6), 1017–1054.

Mumble. (2014). Mumble, the open source VoIP solution. Retrieved from http://wiki.mumble.info/wiki/Main_Page

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, 50(3), 79–83.

Open Metaverse Foundation. (2015). openmetaversefoundation/libopenmetaverse · GitHub. Retrieved from https://github.com/openmetaversefoundation/libopenmetaverse

OpenSimulator. (2013,a). OSSLNPC - OpenSim. Retrieved from http://opensimulator.org/wiki/OSSLNPC

OpenSimulator. (2013b). NPC - OpenSim. Retrieved from http://opensimulator.org/wiki/NPC

OpenSimulator. (2014). OpenSim. Retrieved from http://opensimulator.org/wiki/Main_Page

Plomp T. (2007). Educational Design Research : An introduction. In *An Introduction to Educational Design Research* , Proceedings of the Seminar at the East China Normal University, Shanghai (PR China), November 23-26, 2007 (pp 9-35) SLO – Netherlands Institute for Curriculum Development

Trello. (2015). Trello. Retrieved August 17, 2015, from https://trello.com/.

VB-AUDIO Software. (2015). VB-Audio Virtual Apps. Retrieved from http://vb-audio.pagesperso-orange.fr/Cable/

You, Y. (1993). What we can learn from chaos theory? An alternative approach to instructional systems design. Educational Technology Research Development, 41(3), 17–32.