

LESSONS LEARNED IN DEVELOPING A SECOND LIFE EDUCATIONAL ENVIRONMENT

Lucia Rapanotti and Jon G. Hall

Computing Department, The Open University, Milton Keynes, U.K.

Keywords: Virtual worlds, Second life, Immersive learning environments, Distance education, Software development.

Abstract: Virtual worlds are rapidly spreading beyond gaming and entertainment into education and the corporate world. Should this trend continue, as forecast by the industry, then immersive applications will become more prominent, with bespoke software developed in the metaverse affording both opportunities and challenges. This paper reflects on the experience of developing a learning virtual space based on Second Life as part of an innovation project at The Open University, UK. The paper focuses on the lessons learnt from the viewpoint of managing the development of the learning environment, and could be of benefit to educators and educational technologists who are thinking to engage in this sort of development.

1 INTRODUCTION

Immersive technologies have known an unprecedented growth in recent years, due to faster, cheaper computers and widespread broadband connections (Castronova, 2006). Besides traditional gaming and entertainment applications, some serious propositions are starting to emerge for their use both in education (ReLive, 2008; de Freitas, 2008; Otto et al., 2006; Heldal et al., 2005; Daily et al., 2000) and in the corporate world (Bartholomew, 2008; IBM, 2010; Nguyen et al., 2008), particularly to foster remote collaboration within increasingly globally distributed institutions and industries. Industry forecasts (Driver and Driver, 2009) predict that within the next decade an immersive layer will be added to current web technology, providing a high-degree of integration between immersive technology and the web, and making virtual worlds a mainstream technology for business and education.

Among existing technology, Second Life (SL, 2010) has proved particularly popular in recent years, due to its distinctive combination of 3D simulation with social networking concepts into an open-access commercial grid offered by Linden Labs. While the underlying server and simulation technology is under the company's control, access to the grid is free and the data content is largely provided by the user community: users can acquire virtual land, on which to develop their own content, from Linden Labs through the payment of purchase and maintenance fees; alter-

natively they can rent or even buy some virtual space from other virtual land owners, usually also for a fee. Currently, Second Life is a grid of tens of thousands of connected virtual worlds, often shaped as sunny islands and moulded by their virtual land owners to offer bespoke immersive experiences to visitors. The embodiment of users as avatars affords a powerful sense of presence in comparison to more traditional 2D web technology, and the high degree of customisation, both of land and avatars appearance, is highly appealing to users.

For the past 18 months, the first author has driven an innovation project (vMPhil, 2010) in the Computing Department at the Open University, part of which has concerned the development of a Second Life environment for eResearch, a virtual campus for Computing academics, their research students and collaborators, called deep|think (deepthink, 2010) (A detailed description of the campus and its pedagogical underpinning is outside the scope of this paper and can be found in (Rapanotti et al., 2009) instead.) The virtual environment is meant to support academics and research students at a distance, this being the default mode of operation of our institution. The project complexity stemmed from the large number of stake-holders involved, the relatively large-scale of the virtual world development, and the strategic significance of such a development within the eResearch programme. The Problem Oriented Engineering (POE) framework (Hall and Rapanotti, 2009a; Hall and Rapanotti, 2009b), developed by the authors,

was adopted for project management, and to guide the overall product design and stake-holder validation throughout the project: an account of how it was applied can be found in (Rapanotti and Hall, 2009).

In this paper we focus on development issues which pertain to the Second Life component of the project, with a view to expose both barriers and enablers which other practitioners wanting to engage in similar enterprises may like to consider. After reporting some of the key issues we encountered, we reflect on our experience and offer some practical advice.

2 ISSUES RAISED BY THE PROJECT

In this section we report some of the issues which emerged during the software development of the Second Life environment and which are distinctive of the technology we used. There were, of course, other generic software issues, like accessibility, privacy or access control, but these were not substantially different from those one may encounter in any other software development, hence are not addressed here.

By following the POE philosophy, project management decisions were guided by an assessment of risk and validation activities involving project stakeholders.

2.1 In-house vs Third-party Development

Although the Second Life user community is large and still growing, sculpting and scripting¹ expertise still remains quite rare. At the start of the project, such in-house expertise was very limited within the authors' institution, and the cost and time of developing it would have been disproportionate to the needs and means of the project. Therefore the involvement of third-party software developers was deemed a more cost-effective solution.

Two separate contractual agreements were set up. One was with Linden Labs for the purchase of the virtual land and its continuous maintenance; the second was with a UK software development company to shape the land and introduce fixtures to our specification.

The third-party development effort was quite compressed: it consisted of twenty-two development days, spread out over two months in real time, with

¹Sculpting refers to the creation of 3D objects within Second Life, while scripting is a form of software development used to add interactive behaviour to those objects.

three software developers sharing the work on the project. Development was punctuated by *inworld*² project reviews on a weekly basis in which both academics and software developers took part.

2.2 Fluidity of Design vs Contractual Agreement

Currently, the provision of immersive solutions remains very much in the realm of craft rather than engineering (Driver and Driver, 2009). Each project is unique, each solution radical, and partnership and collaboration, rather than customer-provider relationships are the norm. In our project, by commissioning third party development, the need to establish a contractual product specification very early on had to be measured against the need to accommodate the emergence of innovative design, this being a desirable outcome of working collaboratively.

A set of requirements were gathered based on what the academic team envisioned the pedagogical use of the space should be, but also informed by the design of other Second Life educational locations in existence at the time. As this happened very early in the project life, such requirements were very generic and open to interpretation. Nevertheless they were used as the basis of the customer-provider contractual agreement. Their inadequacy soon became obvious, and while they were a necessary basis to start development, they also led to time consuming re-negotiations of deliverables. In particular, as the environment was shaping up both limitations and new opportunities became apparent, as well as mismatches in the way the requirements had been interpreted by the two parties. Although this is not unusual in software development, even with more conventional types of software, the peculiar nature of Second Life and an initial lack of understanding by the academic team (the customer in this project) of what such technology could actually deliver were exacerbating factors, and led to some frustration.

2.3 Skill Mix

The academic team on this project was made of Computing academics, with prevalent expertise in Requirements and Software Engineering, as well as the preparation of distance higher education materials. On the other hand, the third party developer team came from a tradition of broadcasting and communication technology for business and education, these representing the company's core business. This is not

²The immersive equivalent of 'online'.

unusual in the current immersive landscape, where innovation has been driven from many directions, not just software specialists. The effect on the project was an interesting and unexpected mix of expertise and culture, with the third party developers providing some highly effective visual solutions, and the academic team driving both the pedagogical content and the engineering process.

2.4 From On-site to Inworld

POE promotes assurance-driven design (Hall and Rapanotti, 2009a), which portrays validation from stakeholders (customer, users, software architects, etc.) as a main driver for design in software as well as in other forms of engineering. A high level of stakeholder involvement in software development is a recognised key success factor (see e.g., (Agile, 2010; Beck, 1999)), as it allows for prompt validation of requirements and problem understanding, testing of the product being developed, and for timely corrective actions to take place. In other words, stakeholder validation should be pervasive throughout development, rather than a bolt-on activity towards the end of the software lifecycle.

Immersive technology adds new dimensions to this concept. In this project, the academic team (the customer) was not just in regular contact with the development; instead through the immersive technology it became part of it. The academic avatars could wander around the virtual world being created for them and materially affect its design, appearance and behaviour. This was very effective during development allowing the academic team to make some material contribution to the end product. It was also invaluable post-delivery: through such a process the academic team gained such an intimate knowledge of the software, to be able to alter and adapt it subsequently to their changing needs. The added bonus was that all this could be done with no travel involved: a broadband connection and the freely available Second Life client software was all that was required for these processes to take place. In fact, academic and development teams remained physically distant throughout the development and never met in real-life.

2.5 Change as a Way of Working

Virtual worlds, like Second Life, are rather different from traditional software applications. First of all, Second Life is a live open grid, which is updated on a very regular basis, and with a high turnover of users and simulations: in this respect, it is not so dissimilar to the ever-changing Web, with the added compli-

cation that application developers are not in control of the underlying server technology. Moreover, each SIM³ is very malleable: users with editing rights (e.g. land owners, like the academic team on this project) can, and do, keep changing the appearance and behaviour of their SIMs. This is due in great part to users coming online and interacting with the land, revealing any shortcomings in the design or prospecting new opportunities. It is also partly due to human nature: people like to affect their virtual world just like they would their physical world, except that it is much easier to do it in software. This phenomenon is particularly acute in Second Life, which has as part of their business model to allow users to release their creativity through a thoroughly malleable medium. (This is in contrast, for instance, to some internet gaming environments, like World of Warcraft, in which the virtual environment is essentially fixed).

The need to cater for a changeable world motivated the academic team to employ a land manager on a long term basis as well as instigating a programme of monitoring, evaluation and continuous improvement of the environment. The latter is particularly critical to this project, which had the added requirement to demonstrate the effectiveness of the virtual space to allow distributed research to flourish. Note that land management goes well beyond traditional software maintenance: it is not just about fixing bugs or adding new features; instead it is about continuous reshaping and repurposing.

2.6 Tracking Software Assets

A key activity in software development is the management of software assets, and traditional software engineering makes use of automatic systems and repositories for tracking and versioning control. This is not a possibility when working within a live grid like Second Life. Besides the basic landscape which comes out of the box when a simulation is purchased, everything is created inworld by generating and/or putting together virtual objects, called prims⁴. The current appearance and behaviour of a SIM is the result of the prims which are currently rezzed⁵ within the SIM together with their associated scripts, and it is this current state which is preserved and restored by the underlying simulator. There is no tracking from one

³SIM is short for simulation or simulator. Essentially each island provided by Linden Lab is rendered graphically as a 3D simulation. Users can purchase and link together more than one SIM at a time.

⁴Short for primitives.

⁵Rezzing means to make prims appear within a virtual world.

state to the next, hence no backtracking to restore any previous state. However, when prims are created or acquired by an avatar they can be stored in the avatar's inventory, a rather primitive repository of virtual assets private to that avatar. In the context of third party software development, such as in our project, it is really down to the developers to keep their avatars inventory in good order and manually version control their assets. Of course, manual control is not conducive of large scale software development. Moreover, with more than one developer on a project, as in our case, there was no obvious way to assemble all assets in a single repository and maintain consistent versioning. Overall, however, this did not turn out to be a problem: possibly because of the participative nature of the development, convergence towards a design satisfactory to all appeared to occur very quickly, and only few small adjustments were necessary on a weekly basis, which could be easily accommodated by making use of the available rudimentary inventory.

2.7 Virtual Ownership

The concept of ownership is very strong within Second Life, which to some extent tries to emulate ownership in the material world, albeit for virtual assets: everything, from the virtual land to its tiniest prim, has its owner, either an avatar or an avatar group. Ownership relates to rights and to permissions; for instance, owners may decide who can use their prims and to which extent; they can choose to sell their prims to other avatars for a revenue (there is a full market economy based on this (Castronova, 2006)) or simply donate them for free; while doing so they can establish which rights the new owner may have, like modifying, making copies or transferring ownership subsequently. This creates a rather complex mechanism of rights and permissions, when compared, for instance, to traditional software assets within a directory structure on a conventional computer. Moreover, when prims with different permissions are put together, the resulting combined permissions are not often easy to disentangle. This fact, compounded by the multi-repository issue discussed earlier, resulted in a major technical headache for the project at hand-over time, when the third party developers tried to transfer assets from the developer avatars to the academic avatars (the contract required all assets to be handed-over both in executable and source forms): although all of the assets were transferred, not all their permissions were correctly assigned, and some editing rights were lost.

2.8 Tracking Ideas and Decisions

POE advocates that as well as tracking software artefacts, tracking ideas and design rationale is key to risk management and decision making, and an integral part of stake-holder validation. When design and development become participative, as in our case, with many distributed players, a real issue is to keep track of conversations, decisions and agreed actions. This could not be easily done within Second Life itself, due to the limited support for recording information whether textual or in other forms. In the first phase of development, alongside the inworld review sessions, the project leaders at both ends—academia and software company—corresponded primarily via email, exchanging summaries, project plans, etc. This created a certain amount of overhead to keep track of messages and documents, as well as communicating essential facts and co-ordinating with the other team members. In the second stage of development, post-delivery, a private Ning (Ning, 2009) social network was set up for communication between the academic team and the contracted land manager. This remains in use to this day and has shown to be a superior solution for project communication and tracking. Not only does it provide a common repository, but all communication can be made transparent and all team members can participate and contribute ideas.

3 LESSON LEARNED

In this section we provide some practical advice we have derived from reflection on the reported issues.

Before embarking in a virtual world development project, it is advisable to become familiar with the technology and gain a good understanding of its affordances and limitations. For Second Life, this is relatively straightforward as the grid is openly accessible and most common operating systems are supported. Although a high spec computer is required for best performance, any recent desktop or laptop computer is likely to work well with the grid. From an educational perspective, it is worth visiting existing educational sites to see examples of how the technology is being used.

Similarly to all software projects, it is well worth thinking about one's requirements in some detail before embarking on design and development activities. Many educational virtual worlds tend to be rather generic, used mainly to allow people to meet up or for advertising purposes. Understanding and envisioning how teaching and learning can take place is a completely different matter.

Getting starting with sculpting and scripting in an environment like Second Life is not too complex, but achieving a professional look-and-feel with usable and reliable user interactions requires some degree of sophistication. An audit of in-house skills may help decide whether third-party developers should be employed. If the budget does not allow for a full development to be contracted out, it is still possible to purchase specific services from within the Second Life market place, which includes a lively community of skilled ‘scripters’. Some fixtures can simply be purchased and installed, so, it is worth doing some virtual shopping.

A long term view of how the technology fits within a pedagogical program or institutional strategy is also advisable as costs do not stop after the initial software development. As well as the ongoing maintenance fees to keep the virtual world simulations running, a serious amount of effort may be required in order to reshape and repurpose the space as new needs emerge, and to look after the virtual assets. The ever changing underlying services and new community trends are other factors to be taken into account in long term planning. One also needs to consider how to engage with users. It is highly likely for user training and support to be needed, particularly for adult users who may not have encountered this type of technology before. Considerations of accessibility are also paramount, as is the relation of this technology with more traditional technologies which may be available to users, and how such a relation may affect the overall teaching and learning experience.

In setting up a virtual world development project, it is worth choosing an appropriate development method or philosophy which fits the needs of your project and team culture, and adhering to its design principles. The choice of POE on this project was particularly fruitful, especially the guidance it provides to handle risk, validation and traceability of artefacts and decisions.

When the development team is distributed, some consideration should be given to which practical tools should be used to track artefacts and decisions, as well as which decision processes should be followed, with a clear understanding of roles and responsibilities. In case of contractual work, IP issues should be explored and a formal agreement reached.

Finally, a successful team will have a good mix of skills, including understanding pedagogy, software design, and user-centred design: virtual worlds are very graphical and interactive, so a major effort should go into envisaging the 3D rendering of any learning or teaching experience one may like to include in the simulation.

4 CONCLUSIONS

The paper has reported on our experience of developing a virtual world for higher education using Second Life. The context was an innovation project in which virtual worlds are used as one of the key technologies to support an eResearch programme. The project is noteworthy as a case study due to its relatively large-scale and the strategic significance of the virtual worlds within the programme. By sharing our experience on key development issues which we have encountered, we hope to be of help to educators and educational technologist wanting to embark in similar enterprises.

From our personal perspective as Computer Scientists, our experience has highlighted some challenges and opportunities arising from developing immersive software, and has shown how some of the basic assumptions and practices which characterise more traditional forms of software development do not always sit comfortably within this new paradigm, while others are enhanced by it. Perhaps the most striking effect of this paradigm on practice which we have observed is the way it changes the nature of the customer/user/developer relation, yielding more participative forms of design and problem solving. The effect is that customer and user validation becomes pervasive throughout the development process, rather than a bolt-on activity towards the end of the life-cycle. From our experience, it appears the paradigm offers much potential for distributed collaborative problem solving and decision making, which could be exploited within software engineering activities, particularly in the establishment and validation of system requirements and early design. The project has however also highlighted a number of shortcomings, some that can be attributed to the immaturity of the technology itself, e.g., the lack of proper repositories, others which are intrinsic to the paradigm, which is highly visual and does not afford any sophisticated handling of, for instance, textual information. The project, nevertheless, has illustrated how the use of more traditional 2D social networking technology could mitigate such shortcomings. Still an open question remains as to the extent the two paradigms could be blended and enhanced for best effect.

Also from a Computing perspective it was interesting to notice how POE, a fully-fledged engineering design framework with a track record in high-assurance systems, could still be employed effectively on this project, albeit in a distilled form. From what we have observed, a light version of POE appears to be highly suitable for socio-technical projects, such as that reported here, particularly when a large number

of stake-holders are involved. Further work is planned to test such an hypothesis.

ACKNOWLEDGEMENTS

Many thanks go to other academic members of the Virtual MPhil team, namely Leonor Barroca and Maria Vargas-Vera. We are also thankful to: Shai-ley Minocha, for her Second Life input at the start of the project; Darrel Ince, for championing the project while Head of Department; to Non Scantlebury and James McNulty, from the OU Library, who made suggestions for the design of the inworld library services; to Ahmad Reeves, who is helping with the Virtual MPhil evaluation; and Graham Roberts, our Second Life land manager. Finally, we would like to acknowledge the development team at TwoFour Communications, UK, particularly Andrew Jinman, Sim Quayle and James Abraham.

REFERENCES

- Agile (2010). Agile Alliance. <http://www.agilealliance.org/home> Last accessed: January 2010.
- Bartholomew, R. (2008). Evaluating a networked virtual environment for globally distributed avionics software development. In *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE)*, Bangalore, India. IEEE Computer Press.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley.
- Castronova, E. (2006). *Synthetic Worlds: The Business and Culture of Online Games*. University Of Chicago Press.
- Daily, M., Howard, M., Jerald, J., Lee, C., and Doug McInnes, K. M., Tincker, P., and Smith, R. C. (2000). Distributed design review in virtual environments. In *Proceedings of the International Conference on Collaborative Virtual Environments (CVE)*, San Francisco, US.
- deepthink (2010). deepthink SLURL. <http://slurl.com/secondlife/Deep%20Think%20East/228/29/34/> Last accessed: January 2010.
- de Freitas, S. (2008). Serious Virtual Worlds. A scoping guide. JISC e-Learning Programme, The Joint Information Systems Committee (JISC), UK.
- Driver, E. and Driver, S. (2009). Immersive internet business value study. Immersive Internet Analyst Report Series 2, ThinkBalm.
- Hall, J. G. and Rapanotti, L. (2009a). Assurance-driven design in problem oriented engineering. *International Journal on Advances in Systems and Measurements*, 2(1).
- Hall, J. G. and Rapanotti, L. (2009b). Requirements analysis in context with POE design. In *Proceedings of the International Workshop on Requirements Analysis, London, 6-7 December 2008*. Pearson.
- Heldal, I., Schroeder, R., Steed, A., Axelsson, A.-S., and Spante, M. (2005). Immersiveness and symmetry in copresent scenarios. In *Proceedings of IEEE Virtual Reality*, Bonn, Germany. IEEE Computer Press.
- IBM (2010). Made in IBM labs: Secure, 3d meeting service now available with lotus sametime. <http://www-03.ibm.com/press/us/en/pressrelease/27831.wss> Last accessed: January 2010.
- Nguyen, T., Wolf, T., and Damian, D. (2008). Global software development and delay: does distance still matter? In *Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE)*, Bangalore, India. IEEE Computer Press.
- Ning (2009). Ning. <http://www.ning.com/> Last accessed: January 2010.
- Otto, O., Roberts, D., and Wolff, R. (2006). A review of effective closely-coupled collaboration using immersive environments. In *Proceeding of the ACM International Conference on Virtual Reality Continuum and Its Application (VRCIA)*, Hong Kong. ACM Press.
- Rapanotti, L., Barroca, L. M., Vargas-Vera, M., and Minocha, S. (2009). deepthink: a second life campus for part-time research students at a distance. Technical Report TR2009/1, The Open University.
- Rapanotti, L. and Hall, J. G. (2009). Designing an online part-time master of philosophy with problem oriented engineering. In *Proceedings of the Fourth International Conference on Internet and Web Applications and Services*, Venice, Italy. IEEE Press.
- ReLive (2008). *Researching Learning in Virtual Environments (ReLIVE08)*, Milton Keynes, UK.
- SL (2010). Second Life. <http://secondlife.com/> Last accessed: January 2010.
- vMPhil (2010). The Virtual MPhil. <http://virtualmphil.open.ac.uk/> Last accessed: January 2010.