Behaviour & Information Technology
Vol. 30, No. 1, January–February 2011, 27–37

Design concerns in the engineering of virtual worlds for learning

Lucia Rapanotti and Jon G. Hall*

Computing Department, The Open University, Milton Keynes, UK

(*Received 18 March 2010; final version received 29 April 2010*)

The convergence of 3D simulation and social networking into current multi-user virtual environments has opened the door to new forms of interaction for learning in order to complement the face-to-face and Web 2.0-based systems. Yet, despite a growing user community, design knowledge for virtual worlds remains patchy, particularly when it comes to an understanding of the particular nature of design in virtual environments, the relationship between virtual and real-world contexts of design, as well as the engineering issues it raises and the management of any related risks. In this article, we explore such issues based on our experience of socio-technical engineering of a novel learning programme for higher education with a substantial virtual component. The project's significance stems 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 learning programme. Of particular novelty is our exploration of the relationship between virtual and real-world contexts of design, indicating when they align and differ, showing when tools and techniques translate and when new tools and techniques may be required.

Keywords: virtual worlds; design; Second Life; problem-oriented engineering

1. Introduction

Virtual worlds have shown 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 have started to emerge for their use both in education (Daily *et al.* 2000, Heldal *et al.* 2005, Otto *et al.* 2006, de Freitas 2008) and in the corporate world (Bartholomew 2008, Nguyen *et al.* 2008, IBM 2009), particularly to foster remote collaboration within the 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 the 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.

Despite the growing interest, little is known about how to design virtual worlds effectively to meet specific institutional needs. Based on our experience, this article makes a contribution towards a better understanding of the design of virtual worlds, with particular attention to the relationship between virtual and real-world contexts of design. The first author has driven an innovation project (Rapanotti and Hall 2009) in the Computing Department at The Open University (OU), UK, part of which has contributed to the development of a Second Life (Second Life 2009) environment for e-Research, a virtual campus for computing academics, their research students and their collaborators, called deep|think (Rapanotti *et al.* 2009, deep|think SLURL 2009). The virtual environment was designed to support academics and research students at a distance, which is the default mode of operation of The Open University. The project's significance stems from the large number of stake-holders involved, the relatively large scale of the virtual world development and the strategic importance of such a development within the OU's e-Research programme.

Given the dearth of specific software engineering approaches for virtual world development, the project adopted a generic framework for engineering design, the problem-oriented engineering (POE) framework (Hall *et al.* 2007a, Hall and Rapanotti 2009a,b), developed by the authors. Of particular relevance to this project is POE's ability to deal with both technical and social elements of a problem and its solution. POE was used both as a constructive framework to guide product design and stake-holder validation and as an analytical framework to address specific design concerns. It is the latter use of the framework that is reported in this article. The other aspects are discussed in Rapanotti and Hall (2009), while a description of the virtual world itself can be found in Rapanotti *et al.* (2009).

---

*Corresponding author. Email: j.g.hall@open.ac.uk

ISSN 0144-929X print/ISSN 1362-3001 online

DOI: 10.1080/0144929X.2010.490922
http://www.informaworld.com

The motivation for this work is to obtain a 'blueprint' for sound design practice, encompassing both engineering and pedagogical perspectives, as it is only from their joined consideration that we can hope for successful systems to be designed, deployed and managed. This article contributes to the knowledge of the design of and design in a real/virtual world fusion, particularly in the context of education, in a way which gives appropriate consideration to software engineering needs, pedagogical requirements and the social and educational context of development.

2. Virtual worlds in higher education

The use of 3D virtual worlds in education is on the increase, due to their ability to evoke a strong sense of presence even in remote participants (Witmer and Singer 1998, De Lucia *et al.* 2009), increase their social awareness and communication (Capin *et al.* 1997) and foster online communities and constructivists and situated learning (Bronack *et al.* 2006, Hollins and Robbins 2008).

Among the existing technologies for virtual worlds, Second Life (Second Life 2009) has proved particularly popular among educators: for instance, it is the *de facto* virtual world of choice for further and higher education in the UK (Kirriemuir 2009). It offers a distinctive combination of 3D simulation with social networking concepts within 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. Also, institutions can purchase and customise the virtual land to suit their specific pedagogical needs – see also the virtual world's comparison chart at Virtual Environments Info (2009). The high degree of customisation, both of land and avatars' appearance, is highly appealing to the users.

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. While reliable data on the actual uptake of virtual worlds in higher education is still lacking and is, in fact, difficult to obtain due to the fast pace of change in this sector, a flavour of the widespread interest within higher educational institutions can be gained by looking at the lists published, for instance, at the Virtual Environment home (Virtual Environments Info 2009) and the Second Life in Education home (Second Life in Education 2009) for world-wide snapshots, or the Virtual Worlds Watch network (Virtual Worlds Watch 2009) for further and higher education in UK.

Despite the growing body of work on the use of virtual worlds in education (see, for instance, de Freitas (2008), Researching Learning in Virtual Environments (2008) for overviews), little is still known about their affordances for learning (Dickey 2003, 2005), and issues of design have received only limited attention, with some emphasis on navigation, wayfinding (Darken and Sibert 1996, Bacim *et al.* 2009) and architectural perspectives (Dickey 2004). A recent study has attempted to distil some design principles for educational virtual spaces based on the empirical data collected by visiting existing educational Second Life sites and interviewing current designers and users (Minocha and Reeves 2009).

Engineering design knowledge for virtual worlds remains unexpressed in the literature. It is this gap that the work reported in this article intends to address.

3. POE as an analytical framework

POE is an emerging framework for engineering design, which is seen as a creative, iterative and often open-ended undertaking of conceiving and developing products, systems and processes. Proposed by the authors, POE has been successfully applied to the design of, and design process improvement for, software intensive systems, particularly in the area of high assurance and mission critical systems (Mannering *et al.* 2007, Hall *et al.* 2007b). The framework is comprehensive and detailed introductions can be found in Hall *et al.* (2007a), Hall and Rapanotti (2009a,b). In this section, we outline only those key POE concepts that are relevant to the project under study and their reporting in this article.

3.1. Process of analysis

POE sees engineering design as a problem-solving process, in which interlocked steps of *exploration* and *validation* are repeatedly carried out: exploration is used for knowledge discovery and representation; validation is for quality assurance and risk management. Both involve stake-holders: *finders* contribute to exploration; *validators* contribute to validation. Quality assurance and risk mitigation are the outcome of stake-holders' communication and interaction. Figure 1 gives an illustration of the basic process pattern and its stake-holders; this 'POE*lite*' version is simply the essence of the more comprehensive process model described in Hall and Rapanotti (2009a). That the process is iterative stems from the fact that validation, in general, contributes to further knowledge discovery and understanding, for which subsequent validation might be required.

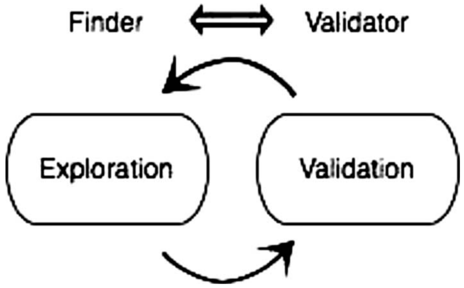

Figure 1. POE*lite*: a reduced POE process pattern and its stake-holders.

3.2. *Problem elements*

Following an engineering tradition (Rogers 1983), POE problems concern the fundamental design question on how a *solution* can be designed to meet some *requirements* in a real-world *context*. Framing a POE problem is then a process of discovery of relevant knowledge pertaining to those problem elements, and from that synthesising the knowledge for the solution.

3.3. *Design concerns*

Design *concerns* are issues that are of particular interest to stake-holders, for which risks have to be suitably addressed, and decisions justified and validated. In POE, design concern is used as an analytical tool as well as to provide a focus for exploration and validation activities. Concerns can be associated both with problem elements and problem-solving process.

We claim that addressing design through concerns contributes to *normal design practices* (Vincenti 1990). Normal design emerges from the recognition that a concern is typical of an engineering discipline and that the standard practices exist to address it. On the other hand, *radical design practices* (Vincenti 1990) involve practices that are related to tackling new problems and for which the degree of success of any design choice is largely unknown, but emerges from the identification of novel concerns or the failing of standard practices to apply successfully to novel problems. As will be seen, the project highlighted some radical design practices related to the nature of virtual world design.

Vincenti's (1990) observation is that, through the accumulation of experience over time, radical cedes to normal, and given the increase in the use of virtual worlds, it is highly likely that our radical practices will evolve to become mainstream, hence our claim of contributions to design normalisation.

4. The study

The discussion in this section is organised around a number of design concerns encountered in the project, with particular reference to learning virtual world that was developed and any related context and requirement knowledge.

4.1. *Context validity*

In POE, context exploration aims at reaching sufficient understanding of the real-world context with which the engineered system will interact, so as to allow the design process to progress towards a fit-for-purpose solution. Context properties yield both affordances and constraints for design, so their correct identification is an essential part of the design process. POE encourages the development of explicit descriptions of the context; these can be of various natures from text to sketches or formal models, to name a few common examples.

Context validity is about making sure that our understanding of the context corresponds to reality, so as to avoid erroneous assumptions on which the designed solution may rely. Therefore, addressing this concern means managing the risk of neglecting or misunderstanding context properties, or making unwarranted assumptions.

The project team was made up of senior academics and senior research supervisors, who were conversant with academic processes and belonging to the target research community; hence the overall risk of making invalid assumptions about the context was limited and appropriately mitigated by various validation activities. The team could both identify relevant context knowledge (as finders) and, to a large extent, validate its accuracy. Further validation was obtained through consultation with academic colleagues and internal regulatory bodies overseeing research degrees including the OU's Research School, which acts as the regulatory body for all research activities including research degrees. On the technology side, validation was also sought through consultation with IT support teams within the university, as well as through internal skill auditing.

The characterisation of the project context is as follows. The organisational context of this project is a UK-based institution, The Open University (henceforth OU), which is a market leader in supported distance higher education world-wide. More specifically, the project was instigated by the then Academic Head of Computing as part of an investment in a range of innovation projects addressing emerging trends in the discipline and our core competencies. The Department offers research (MPhil and PhD) degrees both in full-time residential mode and part-time at a distance mode. Besides access to their supervisors, research students were given opportunities to take part in a structured programme of induction and research training, facilitated by senior academics and highly experienced doctoral supervisors, as well as having access to the wider research community through seminars and other departmental research events throughout the year. At the start of the project, many such events were exclusively residential, with the effect that part-timers found it difficult to benefit from them fully. While the OU had a long tradition of technological innovation in support of distance students, at the start of the project very limited expertise existed within the institution regarding the development and deployment of virtual worlds: such expertise is now growing, with the current project contributing data to feed into a long-term institutional strategy.

4.2. *Requirement validity*

In POE, requirement exploration aims at reaching a sufficient understanding of the real-world need that the engineered system will fulfil, once its contextual properties have been taken into consideration. It is a well-understood fact in software engineering that a poor understanding of requirements is a primary cause of failure of software projects to deliver a fit-for-purpose solution. The requirement is what a designed solution must satisfy and is therefore the main source of validation criteria. As for context knowledge, POE encourages the explicit capture and validation of requirements descriptions.

Requirement validity is about making sure that our understanding of the requirements corresponds to the real need, i.e. managing the risk of addressing the *wrong problem*.

As project instigator and holder of strategic knowledge, the Head of Computing was the main validator in this step, validating through a continuous process of consultation and approval instigated by the project team. The initial requirements were established in negotiation with the Head of Computing, whose strategic objectives guided the project. Among them, the following were particularly relevant to the virtual component of our project: to enhance and develop the Department's provision, and hence grow the graduate market; to promote wider use of technology to support research students at a distance, especially the part-time students.

Owing to their strategic nature, these were considered as very high-level requirements, and part of the design process of the project was to arrive at more specific requirements and a designed solution that could meet them. Hence, *requirement refinement* was an integral part of the subsequent design process, leading to the following operational requirement: to make use of technology to provide research students in distance education mode with a research experience comparable to that of our residential students, including participation in an online structured programme of induction and research training, remote access to seminars and other research events and remote access to a supportive community of peers and academics, particularly in the student's research area of choice.

4.3. *Requirement feasibility*

Requirement feasibility is the process of confirming that the identified requirements can be met within the given context, hence managing the risk that no designable solution exists.

In order to address feasibility, the team conducted an audit of available technology to support distance learners and its use within the Department and the institution at large, particularly to support distant research students. This helped identify a baseline of current practice, upon which the project could be built. It was found that some pockets of excellent practice already existed, but an overall coherent approach to support distant research students through technology was lacking. However, the evidence was sufficient to give us confidence that a design based on a technological solution was feasible, albeit it raised the extra requirement of an accompanying designed user training programme. Training-supported technological innovation from identified best practice is a suggested design practice for this form of project.

4.4. *Solution validity*

In POE, solution exploration aims at the specification of a solution that can ensure satisfaction of the recognised requirements in context.

Solution validity is about making sure that the solution design actually meets the established requirements in context, hence mitigating the risk of designing the *wrong solution*.[1]

The envisaged solution was to provide a rich infrastructure able to support a research degree at a distance, with an MPhil degree chosen as proof-of-concept and a virtual environment as one of the key technological components. The overall infrastructure is comprehensive, encompassing processes, regulations, compliance to national standards, technology and accessibility and user training and skills development. In this article, we focus on design choices surrounding the virtual component, but it is important to appreciate that this is only one piece of a much larger picture.

The main contribution of the virtual world environment to the project's requirements stems from its potential to enhance user participation and to provide a sense of presence and community belonging.

The design was intended to provide a permanent virtual space for students in which to simulate real-life research experiences such as presenting their research with posters or through seminars, attending lectures and workshops, receiving academic guidance and pastoral care from their supervisor, socialising with fellow students, and ultimately developing a sense of belonging to a distinct community.

The risk of a wrong solution was mitigated by mapping known research processes and activities onto design elements of the simulation. Experienced colleagues were also consulted for their expert opinion on Second Life, with secondary risks mitigated by the complementary reading of examples from the literature and participation in relevant inworld demo sessions. It should be emphasised that, given the novelty of the medium, this could only give us some useful pointers, rather than a definite assurance; hence, the risk that the virtual environment would not meet its requirement was only partially mitigated, with any residual risk accepted as part of the innovative nature of the project – this is the nature of radical design.

A long-term programme of evaluation was then set up to collect quantitative and qualitative data on the effectiveness of the technology and its design, to feed within a programme of continuous improvement. The evaluation is underway and is not reported in this article.

4.5. *Solution feasibility*

The concern is about making sure that we have sufficient resource and competencies to carry out and sustain solution development, providing mitigation of the risk of not being able to deliver a solution[2].

On this project, solution feasibility required a range of validation activities and artefacts, from interacting with budget holding stake-holders to ascertain the level of funding to identifying developers with sufficient skills to carry out the work. An innovation fund set up by the Department was used for this project. We realised that in-house expertise in sculpting and scripting[3] was too limited to get the virtual environment up and running, 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 to develop a baseline virtual environment in a relatively short period of time, with a professional look-and-feel and within the available budget. We soon realised that a long-term development view was also needed, due to the particular nature of Second Life. First of all, Second Life is a live open grid, which is updated on a regular basis, with a high turnover of users and simulations: in this respect, it is not so dissimilar to the ever-changing Web, with the added complication that application developers are not in control of the underlying server technology. Moreover, each SIM[4] is very malleable: land owners (including the academic team of this project) can, and do, change 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 affect their physical world, except that it is much easier to do it in software. This phenomenon is particularly acute in Second Life, which, as part of their business model, has to allow users to release their creativity through a thoroughly malleable medium. The need to cater for a changeable world led us to employ a land manager on a long-term basis as well as to instigate a programme of monitoring and continuous improvement of the environment. 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. To mitigate the risks of relying too heavily on external contractors for long-term sustainability, team members also undergo training to develop their own skills and competencies.

In summary, considerations of the pump-priming nature of the project and the justification of its long-term sustainability is another important design consideration.

4.6. *Sound design*

This concern is about making sure that the design follows professional standards and exhibits an appropriate level of quality, hence addressing the risk of delivering a sub-standard solution, i.e. a solution that falls short in non-functional requirements and aesthetics, though it meets all functional requirements.

As already mentioned, established design practices are yet to become widely recorded and accepted. There are, however, many examples of educational environments in Second Life, from which one can learn any pitfalls of a particular decision, and a growing literature on the subject. Before embarking on the actual design the team visited a number of such sites, read the relevant literature and sought advice from within and outside the institution. This complemented the professional expertise in software development and design already possessed by the team. Also, third-party developers were chosen with an established track-record in virtual world development for education.

4.7. Compliance

Compliance relates to existing relevant regulations and organisational strategies and practices, and addresses the risk of misalignment.

Validation was sought through consultation with the OU Research School to check alignment with institutional strategy and regulations on students' research induction and training, and formal approval was received to allow some residential processes to take place online instead. The Computing Postgraduate Research Tutor, who oversees research students and their pastoral care, was also consulted to check alignment with current practices. More general institutional strategies for the use of virtual worlds and to support research students online are still under development, and members of the project team have joined appropriate steering groups, where the project is seen as an opportunity to experiment and disseminate.

The provision of guidance back to, and the involvement in institutional strategic initiatives (and their establishment) is an important part of reflection on and contribution to the future: including it as a design practice establishes another important, but easily overlooked, design norm.

4.8. Asset tracking and management

This concerns a key activity in software development, i.e. the management of software assets, and addresses the risk of losing or compromising assets.

Traditional software engineering makes use of automatic systems and repositories for tracking and version control of program code, but this is not possible when working within a live grid like Second Life for the following reasons. Besides the basic SIMs, which are rented from Linden Labs, everything else is created inworld by generating and/or putting together virtual objects called prims.[5] The current appearance and behaviour of a SIM is the result of the prims that are currently rezzed[6] within the SIM together with their associated scripts, and it is this current state that is preserved and restored by the underlying simulator. Prims only exist inworld, either rezzed in the virtual environment or stored as items in an avatar's inventory, a rather unstructured and unsophisticated repository of virtual assets private to that avatar. There is no tracking of how they change from one state to the next, hence no backtracking to previous states.

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 by storing previous copies within the inventory. 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 is no obvious way to assemble all assets in a single repository and maintain consistent versioning. Therefore, the risk of losing or compromising virtual assets can be high in this type of development. In our project, version control was managed manually and a pragmatic decision was taken that only the latest validated version of each asset should be preserved. This points to the need for scalable tools for asset management.

4.9. Asset ownership and handover

This is concerned with the way software assets are transferred from developer to customer at the end of a software development project and addressing the risk of compromising IP and other rights one may have on the assets.

In traditional software, some form of executable is usually delivered to the customer. Sometimes, as in our project, source code is also handed over so that the customer has full control on the code and may also be able to alter it. Depending upon the contractual agreement, some degree of software ownership and related rights are transferred from developer to customer as part of the handover process.

Second Life emulates, to a certain extent, 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 is related to rights and permissions; for instance, owners may decide who can use or modify 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; while doing so they can establish which rights the new owner may have, with options to modify, make copies or transfer 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 often not easy to disentangle. This, compounded by the fact that virtual assets only exist inworld and with the multi-repository issue discussed earlier, resulted in a major technical headache for the project at handover time, as the contract required all assets to be handed over both in executable and source forms. While scripts associated with prims can be copied and pasted as text outside the grid, the prims themselves cannot (at most they can be transferred from one virtual world to another). Therefore, handover is a process that can only occur entirely within the virtual world, in our case from the developers' avatars to the academics' avatars. The risk of losing control of virtual assets as a result of handover appears very high, as often creator privileges override other privileges and limit the possibility of restoring or changing virtual assets, which may become damaged or obsolete. Moreover, should access to the account of the avatar that owns the asset directory be compromised, then all assets may become unretrievable. In our project, the handover was only partially satisfactory. Although all virtual assets were transferred, not all of them had the right permissions set up; as a consequence some editing rights were lost.

Although this is a concern of particular relevance to Second Life development, the early consideration of the mechanisms for handover, and their contractual recording, is of critical importance in all virtual world development.

4.10. Interaction design

This concerns the way that a design allows users to interact with the software and addresses the risk of users being unable to make any effective use of the software.

It is a well-understood concern in traditional software development, with interaction design focusing on how well bespoke software can be designed to support users in their daily activities (Sharp *et al.* 2007). Many standards and guidelines exist for user interface and user interaction design for conventional software, which are widely applied and understood. However, the same cannot be applicable for virtual worlds. A major difference is that all user interaction is mediated by avatar interaction within the virtual world, so there is a level of indirection between both user actions and avatar actions, and user context and avatar context, which does not exist in traditional software. Therefore, a key question in virtual world interaction design is not just how well the software can be designed to support a user's activities, but also how well the virtual world can be designed to support an avatar's activities and how this translates into a user's activities. The additional risk to be managed is the inability to bridge the real world's/virtual world's divide-and-simulate activities of limited value of the users. In this project, we used the artifice of user scenarios (Alexander and Maiden 2004) to capture real-world research processes and activities, and mapped them into corresponding 'avatar journeys' through the virtual world. This proved effective as a tool to ensure coverage of all relevant activities; the measure of the effectiveness of such a mapping remains the subject of evaluation.

We suggest that further research into scenario-related techniques as the basis of interaction design of virtual worlds would be beneficial to provide some mitigation to the risk of poor interaction design, particularly by linking such scenarios to specific academic processes and activities in real life. It is also important to consider emerging educational theories (e.g. Bronack *et al.* 2006) that have started to shed some light on the effective ways of using virtual worlds in education. These may be intended to not only guide the mapping from scenarios onto virtual journeys, but also to offer learning experiences inworld with no counter-part in real life. For instance, in the project we had to devise ways to help users become familiar with the environment itself. To this end, we used the idea of quest from learning games (Ramondt 2008) to develop an inworld interactive tutorial.

4.11. Developmental testing

This concerns the ability of customers and users to test the software during development and affect the way it is shaped; it addresses the risk of possible software defects due to a misunderstanding of requirements or context properties, or even poor interaction design.

Interestingly, this aspect of software development is facilitated by the new medium. While customers and developers may remain geographically distant, as was the case in our project, their avatars can come together on a regular basis in the virtual world. Given the permanent nature of the Second Life grid, 'customer and user avatars' can experience the virtual world being created for them and, through such an experience, materially affect its design, appearance and behaviour. This can be a very effective mechanism during development, which allows a real team to make material contributions to the virtual end product. It can also be invaluable post-delivery: through such a process a real team can gain such an intimate knowledge of the software, so as to be able to alter and adapt it subsequently to their changing needs. An added bonus is that all this occurs without real travel involved: a broadband connection and the freely available Second Life client software is all that is required for these processes to take place. Therefore, it could be argued that the risk of delivering software that is unacceptable to users could be greatly reduced by continuous inworld developmental testing. Indeed, in our project, this was very successful and convergence towards a design satisfactory to all parties was reached within few weeks.

5. Discussion and conclusion

Even though the use of virtual worlds is seeing an unprecedented growth, with virtual worlds having come to be regarded at the forefront of pedagogic technologies, there is little known about how to design virtual worlds effectively to meet specific institutional needs. Based on our experience, we have considered an engineering bridge between real and virtual worlds in this article, in which real world pedagogic and other concerns are re-evaluated within the virtual context. The context of the article is an innovation project, driven by the first author in the Computing Department of The Open University (UK), in which Second Life contributes to a pedagogic platform for e-Research, supporting both academics and research students at a distance.

The engineering of virtual worlds as social networks for interactive learning is a new engineering discipline that must combine traditional pedagogical thinking with ultra-rapid technological change. The simple transliteration of real-world engineering approaches to pedagogical design into a virtual world provides some basic support for the real/virtual fusion, but the support is limited. This article contributes to the normalisation of the design of real/virtual world fusions, in the context of education, in a way which gives appropriate consideration to software engineering needs alongside pedagogical requirements and the educational context of development.

We have used the machinery of our POE framework and have shown how new design challenges can be met. The detailed contributions of the article are summarised in Table 1, in which we have detailed 18 responses to the design challenges of using virtual worlds as a learning platform. In some areas, such as the context validity concern – by which we check that our understanding of the context is good enough – the subsequent analysis in the virtual world provides no extra difficulty when compared to traditional software projects. In others, such as the asset ownership concern, the virtual world paradigm differs sufficiently from that in the real world – in this specific case because the notions of ownership differ between real

Table 1. Design concerns in the engineering design of virtual worlds.

Name	Concern	Risk	Specific issues arising from virtual worlds	Design practices to mitigate risk
Context validity	Does our understanding of the context correspond to reality?	Making wrong assumptions on which the solution will rely upon	None found	Make explicit descriptions of key assumptions for stake-holder scrutiny
Requirement validity	Does our understanding of the requirements correspond to a real need?	Addressing the wrong problem	None found	Make explicit descriptions of requirements for stake-holder scrutiny. Refine requirements to a sufficient operational level. Review existing best practice Consider training requirements
Requirement feasibility	Can the requirements be met within the given context?	Addressing an unsolvable problem	None found	Expose the solution rationale and any argument that links solution design to requirements in their context
Solution validity	Does the designed solution meet the requirements in context?	Designing the wrong solution	Lack of data and consensus on the pedagogical value of virtual worlds	Identify real-life processes and activities and map them onto elements of design. Instigate a process of continuous monitoring, evaluation and improvement once in operation

(*continued*)

Table 1. (*Continued*).

Name	Concern	Risk	Specific issues arising from virtual worlds	Design practices to mitigate risk
Solution feasibility	Do we have the capability of delivering and maintaining the designed solution?	Designing a solution that cannot be put together	Expertise is still rare. Technology itself can be at odds with organisational culture and/or user attitude. Long-term sustainability to be considered, together with response to rapid change	Evaluate trade-offs between in-house and third-party development. Identify early adopters and instigate a programme of skill development. Ensure sufficient level of funding. Adopt flexible long-term management policies
Sound design	Have we applied the best design practice?	Compromising the quality of the solution	Lack of established design and engineering practices	Take expert advice. Sample existing developments. Engage with the wider community of early adopters
Compliance	Do we meet all relevant regulations and follow established practices?	Breaching norms and regulations	Lack of established institutional practices and norms	Contribute to strategic initiative (and/or their establishment)
Asset tracking	Can we account for all assets and their change?	Losing assets	Lack of tools for tracking and for the version control of virtual assets	Establish processes and procedures at the onset
Asset ownership	Are all asset rights protected?	Losing IP and other rights	Virtual ownership doesn't have a direct mapping onto real-world ownership. Asset handover can be problematic	Clarify IP rights and asset transfer mechanisms at the onset
Interaction and pedagogical design	To which extent is the designed interaction meaningful to users and learners?	Designing meaningless interactions	Virtual-world/real-world divide to bridge	Develop specific scenarios and ways of mapping them onto 'virtual journeys'. Consider emerging pedagogical theories for learning in virtual worlds
Development testing	To which extent are customers and users involved in the development?	Dissatisfied customers and users	User vs. avatar testing	Involve representative stake-holders as early as and as often as possible through in world testing

and virtual worlds – so that a radical perspective of design is necessary. We have identified 11 cases of potential dissonance, of which 8 provide new engineering design challenges, which should be the focus on future work.

Acknowledgements

The authors thank the academic members of the Virtual MPhil team in their institute, namely Leonor Barroca and Maria Vargas-Vera. They are also thankful to Shailey Minocha, for her Second Life input at the start of the project; Darrel Ince, the then Head of Department, for championing the project; Non Scantlebury and James McNulty, from the OU Library, who made suggestions for the design of the inworld library services; Ahmad Reeves, who has helped with the Virtual MPhil evaluation; Graham Roberts, the Second Life land manager. Finally, the authors acknowledge the development team at TwoFour Communications, UK, particularly Andrew Jinman, Sim Quayle and James Abraham.

Notes

1. We gloss over the semantics of *wrong solution* being no solution at all. A wrong solution is simply meant as a delivered artefact that does not solve the problem.
2. The reader will note that solution feasibility is distinct from being able to solve the problem, in providing validation that both problem *and* solution understanding are sufficiently supported.
3. Sculpting 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.
4. SIM 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.
5. Short for primitives.
6. Rezzing means to make prims appear within the virtual world.

References

Alexander, I. and Maiden, N., 2004. *Scenarios, stories, use cases through the systems development life-cycle*, New York: John Wiley.

Bacim, F., Bowman, D., and Pinho, M., 2009. Wayfinding techniques for multiscale virtual environments. *In*: *Proceedings of IEEE Symposium on 3D User Interfaces 2009*, 14–15 March 2009, Lafayette, LA, USA.

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)*, 17–20 August 2008, Bangalore, India. IEEE Computer Press.

Bronack, S., Riedl, R., and Tashner, J., 2006. Learning in the zone: a social constructivist framework for distance education in a 3-dimensional virtual world. *Interactive Learning Environments*, 14 (3), 219–232.

Capin, T., *et al*., 1997. Virtual human representation and communication in VLNet. *Computer Graphics and Applications*, 17 (3), 42–53.

Castronova, E., 2006. *Synthetic worlds: the business and culture of online games*. Chicago/London: University of Chicago Press.

Daily, M., *et al*., 2000. Distributed design review in virtual environments. *In*: *Proceedings of the International Conference on Collaborative Virtual Environments (CVE)*, 10–12 September 2000, San Francisco, USA.

Darken, R. and Sibert, J., 1996. Wayfinding strategies and behaviours in large virtual worlds. *In*: *Proceedings of CHI 96*, Vancouver, Canada, ACM, 142–149.

deep|think SLURL, 2009 [online]. Available from: http// slurlcom/secondlife/Deep%20Think%20East/228/29/34/ [Accessed October 2009].

de Freitas, S., 2008. *Serious Virtual Worlds. A scoping guide*. JISC e-Learning Programme. UK*:* The Joint Information Systems Committee (JISC).

De Lucia, A., *et al*., 2009. Development and evaluation of a virtual campus on Second Life: the case of Second DMI. *Computers and Education*, 53 (1), 220–233.

Dickey, M.D., 2003. Teaching in 3d: pedagogical affordances and constraints of 3d virtual worlds for synchronous distance learning. *Distance Education*, 24 (1), 105–121.

Dickey, M.D., 2004. An architectural perspective for the design of educational virtual environments. *Journal of Visual Literacy*, 24 (1), 49–66.

Dickey, M.D., 2005. Three-dimensional virtual worlds and distance learning: two case studies of active worlds as a medium for distance education. *British Journal of Educational Technology*, 36 (3), 439–451.

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), 26–31.

Hall, J.G. and Rapanotti, L., 2009b. Requirements analysis in context with POE design. *In*: *Proceedings of the International Workshop on Requirements Analysis*, 6–7 December 2008. London: Pearson.

Hall, J.G., Rapanotti, L., and Jackson, M., 2007a. Problem oriented software engineering: a design-theoretic framework for software engineering. *In*: *Proceedings of IEEE Software Engineering Formal Methods (SEFM 2007)*, September, London, UK.

Hall, J.G., Mannering, D., and Rapanotti, L., 2007b. Arguing safety with problem oriented software engineering. *In*: *Proceedings of the 10th IEEE International Symposium on High Assurance Systems Engineering (HASE 2007)*, 14–16 November 2007, Dallas, Texas, IEEE Computer Society.

Heldal, I., 2005. Immersiveness and symmetry in copresent scenarios. *In*: *Proceedings of IEEE Virtual Reality*, 12–16 March 2005, Bonn, Germany, IEEE Computer Press.

Hollins, P. and Robbins, S., 2008. Educational affordances of multi user virtual environments (muve). *In*: *Proceedings of Researching Learning in Virtual Environments (ReLIVE 08)*, 20–21 November 2008, Milton Keynes, UK, 172–180.

IBM, 2009. *Made in IBM labs: Secure, 3d meeting service now available with lotus sametime* [online]. Available from: http://www-03.ibm.com/press/us/en/pressrelease/27831.wss [Accessed October 2009].

Kirriemuir, J., 2009. Early summer 2009 virtual world watch snapshot of virtual world activity in UK HE and FE. *Technical report*, Virtual World Watch.

Mannering, D., Hall, J.G., and Rapanotti, L., 2007. A problem-oriented approach to normal design for safety-critical systems. *In*: *European Joint Conferences on Theory and Practice of Software (ETAPS)*, 24 March – 1 April 2007, Braga, Portugal. Proceedings of Fundamental Approaches to Software Engineering (FASE).

Minocha, S. and Reeves, A.J., 2009. Interaction design and usability of learning spaces in 3d multi-user virtual worlds. *In*: *Proceedings of IFIP Human Work Interaction Design 2009: working conference on usability in social, cultural and organizational contexts*, 7–8 October 2009, Pune, India.

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)*, 17–20 August 2008, Bangalore, India, IEEE Computer Press.

Otto, O., Roberts, D., and Wolff, R., 2006. A review of effective closely-coupled collaboration using immersive CVE's. *In*: *Proceedings of the ACM International Conference on Virtual Reality Continuum and Its Application (VRCIA)*, 14–17 June 2006, Hong Kong, ACM Press.

Rapanotti, L., *et al.*, 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*, 24–28 May 2009, Venice, Italy, IEEE Press.

Ramondt, L., 2008. Towards the adoption of massively multi-player educational gaming. *In*: *Proceedings of researching learning in virtual environments (ReLIVE 08)*, 20–21 November 2008, Milton Keynes, UK, 258–268.

Researching Learning in Virtual Environments (ReLIVE08), 2008. 20–21 November 2008 Milton Keynes, UK.

Rogers, G., 1983. *The nature of engineering*. London: The Macmillan Press.

Second Life, 2009. [online]. Available from: http://secondlife.com/ [Accessed October 2009].

Second Life in Education, 2009. [online]. Available from: http://sleducation.wikispaces.com/ [Accessed November 2009].

Sharp, H., Rogers, Y., and Preece, J., 2007. *Interaction design: beyond human–computer interaction*. 2nd ed. John Wiley & Sons.

Vincenti, W.G., 1990. *What engineers know and how they know it: analytical studies from aeronautical history*. Baltimore/London: The Johns Hopkins University Press.

Virtual Environments Info, 2009. [online]. Available from: http://www.virtualenvironments.info [Accessed November 2009].

Virtual Worlds Watch, 2009. [online]. Available from: http://virtualworldwatch.net/ [Accessed November 2009].

Witmer, B. and Singer, M., 1998. Measuring presence in virtual environments: a presence questionnaire. *Presence*, 7 (3), 225–240.