

# Virtual worlds interoperability - a summary

Copyright 2008 Gareth Nelson ([gareth@litesim.com](mailto:gareth@litesim.com))

This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

## ***Introduction***

This essay is an overview of virtual worlds interoperability, that is the ability for virtual worlds to cooperate in multiple ways. It is assumed that the reader is already familiar with the basics of virtual worlds.

Like the traditional WWW, virtual worlds serve numerous purposes ranging from informational and business use (e Commerce, virtual conferences) through to more mundane socialization and entertainment uses (games, social networking and hobbyist activity). While the medium is more engaging and technologically advanced, the basic categories of uses that virtual worlds can be put to are the same.

However, despite the numerous potential uses virtual worlds can be put to currently, they have traditionally lacked the one vital ingredient which has made the WWW the success it is. They do not easily connect to or link to each other. Each virtual world is (as many have described them) like a walled garden.

Up until recently, the situation of virtual worlds being walled gardens has been an endemic problem affecting the entire industry and causing user base fragmentation amongst other issues which have resulted in the great potential of virtual worlds (or “the metaverse”) to go untapped.

It is my firm belief that when this situation is resolved, we will see the contributions of virtual worlds to the arts and sciences jump up overnight. One need only look at the ever-present usage of the phrase “Google it” to see how much the traditional web has enhanced the laypersons ability to locate almost any information they could desire.

You will find here a very nice summary of how to resolve much of the interoperability issues facing virtual worlds today in this document.

## ***Traditional virtual worlds..... and where the problem lies with them***

Traditionally, virtual worlds tend to be owned by single entities with set purposes. Purposes virtual worlds have been applied for have mainly consisted of entertainment pursuits: games, roleplays, artistic sandboxes, social networking, virtual pets and other mundane uses.

Sadly, this focus on single uses has led to 2 critical issues. Firstly, these worlds are less worlds and more very domain-specific simulations. Secondly, due to them being domain-specific, they tend to be less powerful for those who wish to use them to build more complex works upon them.

Online games have a very obvious set purpose for example: fun and entertainment. But even within this purpose the type of “fun” one might wish to have may be restricted. For example, many MMORPGs require you to complete quests and other tasks in order to advance your character. This says nothing about whether you may wish to remake your own mini game within them using the same platform either.

Social networking within 3D worlds (as seen on sites such as IMVU or even goggle lively) is another highly domain-specific simulation. It is questionable whether it is even appropriate to call such sites worlds.

It is not only unpractical/unfeasible, but in many cases actually impossible or even illegal (as reverse-engineering would be required) to extend these platforms beyond their original purpose. They were not built for user-generated content, or if they were, there are still aspects of the system which the user has no control over due to proprietary source code or network protocols.

What we need is an extendable virtual worlds platform, or at least one which is “extendable enough” that a talented hacker (in the traditional sense of the term) can reverse-engineer portions of it or the whole system legally. Failing that, we could get the required flexibility by allowing all these domain-specific simulations to talk to each other in some manner and thus get the strengths of both specialization and general-purpose simulations. Even better, we can link multiple *general-purpose simulations individually applied to domain-specific problems* and get even greater flexibility, in the same way that HTML is a general-purpose language for marking up hypertext documents but is applied to domain-specific problems through topical websites.

If we can take general-purpose virtual worlds simulations platforms and apply instances of them to domain-specific problems then link them together, then we get a vastly more powerful and flexible platform for information interchange and human interaction than today's WWW.

Should we manage to meet the complex social, legal and technical challenges of this task, we may find that it is the real world and not cyberspace that will receive the greatest benefits as mankind receives another huge boost to our ability to impart and receive knowledge.

## ***The simplest of general-purpose interfaces***

In the previous section I described a system of general-purpose simulations/VWs linked together. Linking together various simulations like this requires that you either have a way to bridge the network protocols that they understand, or that you use the same protocol for all. As our goal is to encourage the highest possible innovation and technological advancement while enabling each individual world to solve specific problems, it would be foolish to insist upon every single virtual world in existence being absolutely identical.

The easiest solution is to instead take the simplest aspect of these disparate virtual worlds that can be shared and share that and that alone. Doing this gives maximum autonomy to individual worlds while allowing some linking between them.

It just so happens that a system already exists for linking wildly separate network protocols running on entirely different computer systems. It's called a URL:

<telnet://example.com>

ssh://example.com

<http://www.example.com/somepage>

secondlife://Baikal/120/78/20

<ftp://test:1234@ftp.example.com>

<mailto:gareth@litesim.com>

The protocols in the above URLs are all wildly different, but the common point between them is that they're all simple strings that specify the protocol, the hostname and even some extra parameters to use with that protocol (such as the “somepage” or the co-ordinates within the Baikal region on Second Life). If you have client software which understands the protocol then these URLs are often clickable in web pages or accessible straight from your desktop on modern GUIs.

URLs are a true general-purpose interface that could trivially be applied to virtual worlds (and in fact have been already). However, they have a weakness: you still need client software for each protocol, possibly even each service. You may also need a separate account for each service. If we want to make our virtual-world linkups as seamless as possible then URLs are insufficient.

There are other forms of linking which consist of visible (within the 3D simulation) portals to other worlds or other regions of that particular world (this is one of the main features of the Croquet project for example), but they are all dependent upon shared network protocols. If you want to go to another world and it does not use the same protocol, neither URLs or virtual portals will help you get there.

The only thing that will in these situations is to download yet another client, possibly create an account on the other virtual world service provider and login separately. Once you've got this new client and new account setup of course then your URLs might do just fine, and of course creating new accounts on different websites has not held back growth on the WWW.....

.....except you don't generally need a new account to view a different website. You might need an account to setup account-specific features on that website (to post pictures on flickr.com for example), but to simply view the homepage you don't need a new browser and a new account. In virtual worlds, we have a situation where casual “browsing” across different worlds is essentially impossible.

A more general-purpose solution is required, one that does NOT require users to download a new client and create a new account for each separate virtual world they wish to explore. There are many contenders for this catch-all solution, but I do not believe they will ultimately be able to unite enough virtual worlds to create a casually browsable metaverse in the short term as-is.

What is flawed in these catch-all solutions is that they generally have no concept at all of differences in the protocol layer between virtual worlds. One such solution is OGP (Open Grid Protocol) from Linden Lab™. OGP is at the time of writing a protocol consisting of a set of web services which guide the Second Life™ viewer software to a specific simulator host and provide basic identity information (first and last name and User ID).

Since these general solutions have absolutely no concept of differences in protocol between virtual worlds, they will fight and compete with each other to be the dominant standard, or simply become too fragmented to be useful. The correct way to unite disparate virtual worlds is rather more messy than many software engineers would like: protocol bridges and multiple protocol handlers within clients.

### ***The meta-metaverse***

The “meta-metaverse” is the term I shall (only half-seriously) coin for the practice of using proxy servers, client-side mods, protocol bridging and other techniques to unite separate virtual worlds. Running at a meta layer above all these separate virtual worlds, the meta-metaverse will be the combination of domain-specific hacks combined with general-purpose protocols that enable a true seamless virtual world browsing experience.

Quite simply, there is a lot of things that virtual worlds share in common with each other that can be abstracted away and managed by another protocol. If this can be pulled off, we can write bridges that share a “mother tongue” in another protocol while translating the experience of various virtual worlds into a form that any arbitrary client can understand. When combined with clients that have multiple protocol handlers, this should enable a swiss-army knife combination of virtual worlds simulation platforms, protocols and clients to suit nearly all situations. The numerous combinations available in such a scenario alone represent a whole new level above simply creating a more immersive or high-

bandwidth user experience.

With such a system in place, it would be feasible to produce bridges for protocols other than virtual worlds protocols: HTTP, SMTP, IRC, even DNS all come to mind. What if you could “see” DNS records or IM Second Life™ users via SMS text messages?

URLs play a big role here too, as they specify which kind of bridge to use when accessing a remote service. A well-written bridge could be designed to either co-operate directly with virtual world service providers and allow guest users, or it could automatically sign up first-time users and fill out account settings to closely match the user's main account.