Virtual worlds are at a stage of development today similar to where the online community was in the early 1990s, when proprietary systems competed for market share, before the World Wide Web introduced common open standards. At present, most virtual worlds (such as Second Life) are proprietary, although some important open source efforts exist (for example, OpenSimulator), as do early efforts at standardization.

Most standards activity to date has surrounded interoperability, and discussion about virtual world standards is often based around such worlds’ graphical look rather than how to represent the content as a whole. This isn’t surprising, given that most virtual worlds are graphical, and the technology grew out of video games. Certainly, many use cases require the ability to represent high-fidelity graphics, but others involve rendering on low-end enterprise machines, in browsers, or even on mobile devices.

However, we should not assume that virtual worlds are graphical. Indeed, by focusing on their visual aspects, we miss the opportunity to adopt a more generic approach. For instance, visually impaired users can play audio versions of immersive games (such as AudioQuake; www.agrip.org.uk/download) by employing sounds to create the equivalent of echo location; other audio games are even playable over the telephone. Other work annotates graphical virtual worlds for visually impaired users (http://services.alphaworks.ibm.com/virtualworlds) and even creates virtual guide dogs (www.virtualguidedog.com/news.htm) — although these latter approaches tend to rely on the original creator calling each object by a sensible name.

When considering an audio world, the graphical elements no longer take center stage; rather, a scene’s semantic elements — the position and types of doors and chairs, for example — become paramount.

Three main approaches exist for virtual world standardization: interoperability, infrastructure, and semantic markup languages. Most standards efforts to date have concentrated on virtual worlds’ graphical aspects rather than their semantic content. Here, the author uses an example audio-only virtual world to reveal the advantages of defining any virtual world in semantic terms, so it can be realized in various ways — for example, by graphics or audio. Examples of semantic markup already used in conjunction with virtual worlds indicate that this approach is promising.
We believe this opens up a better way to represent any virtual world — defining it in semantic terms, then letting the client render it in whichever way is most appropriate to the user and the platform he or she is using. The world will naturally need to include far more than just its objects’ semantic elements, but managing how those objects are represented to the user is perhaps the most crucial part of any virtual world.

**Toward a Semantic Virtual World**

Standards for virtual worlds aren’t new. The Virtual Reality Markup Language (www.w3.org/MarkUp/VRML) dates back to 1994. In 2006 and 2007, with virtual worlds riding on the hype curve, the IETF Virtual World Region Agent Protocol (VWRAP; https://datatracker.ietf.org/wg/vwrap/charter) standards group considered avatar interoperability, and the Virtual Worlds Roadmap effort (http://metaverseroadmap.org) was started, then abandoned when initial corporate interest in virtual worlds waned. In 2010, IEEE started its own thread of standards development (www.metaversestandards.org).

In furthering these efforts, the virtual world community could adopt three main types of standards:

- **Interoperability standards** provide ways for different virtual worlds to exchange information — for example, by defining standards to move avatar information and assets between worlds. As mentioned, this is where most virtual world standardization efforts have been focused.

- **Infrastructure standards** define common internal representations and application programming interfaces to be used across the infrastructure (such as HTML/HTTP for the Web). Vendors and providers adopt these standards but develop their own implementations (Internet Explorer, Chrome, Firefox, and so on).

- **Semantic markup standards** define standards in terms of the role each object plays within the environment; its visual (or other) representation is stored separately, even outside the virtual world.

Here, I focus on semantic markup standards. To understand how semantic markup might work with virtual worlds, let’s briefly look at some examples of virtual world platforms, standards, and authoring tools that already use a semantic approach.

**Vastpark and IMML**

The 3D virtual world VastPark (www.vastpark.com) uses an Immersive Media Markup Language (IMML; www.vastpark.org/projects/vp/wiki/What_is_IMML). IMML is encoded in XML, which defines how the world is laid out, how it’s lit and seen, what avatars exist, objects’ behavior in the world, and communication with the outside world.

IMML itself says nothing about what anything looks like. Instead, it identifies a link to a graphical representation of each object, focusing on where that object is in the scene and how the user can interact with it. Although the visual look might be important in some applications, users can now optimize it to the power of their computer. At runtime, the client looks within its object inventory to find the 3D representation of each object (“autumn sky box,” “table,” “chair,” and so on) and places and renders it in the world. In theory, if a Vastpark client had a different 3D file linked to “chair” and “table,” one user might see a modernist scene and another a gothic scene, but that scene would behave in the same way. VastPark’s Metaforik service (http://vastpark.org/wiki/vp/What_is_Metaforik) lets different clients automatically choose different 3D files for each object depending on capability — so a client running on a high-power PC might see a more detailed object than one running on a mobile device or in a browser.

**CityGML**

The CityGML standard (www.citygml.org), developed by the geographic information systems community, lets planners mark up a city layout in a simple, platform-independent way. CityGML allows for rendering the view at the most appropriate level of detail and, like IMML, captures position and behavior and can bind to 3D models separately. So, again, we have a way of separating out a space’s layout and meaning from its graphical representation.

**PIVOTE and Medbiquitous Virtual Patient**

At Daden, we initially developed the PIVOTE (PREVIEW Immersive Virtual Operational Training Environment) project for a paramedic...
training simulation in Second Life. The “traditional” way to build such an exercise would involve the in-world Linden Scripting Language and embedding text in images or note cards. However, this approach doesn’t scale well and is hard to maintain.

Instead, we adopted the ANSI Medbiqitous Virtual Patient (MVP) standard (http://groups.medbiq.org/medbiq/display/VPWG/Home), used internationally by medical schools to share Web- and PC-based medical training exercises, and applied it to virtual worlds.

We define the exercises on the Web using MVP markup. Each user action in the virtual world triggers, via a Web service, an action in the MVP exercise definition, which generates a response that’s passed back into the virtual world to the user. The exercise author must still create 3D assets and scenes within the virtual world, but they don’t contain any exercise logic or content and are simply bound by an ID to the MVP actions. All the exercise’s non-3D content (text, images, or video) and its structure are authored, stored, and managed on the Web. The resulting system moves the scene’s behavior from the virtual world to the Web, where the author can more easily manage it and where nontechnical staff can more easily maintain content and author new scenarios. This also preserves the training content if the target virtual world changes.

Although PIVOTE doesn’t address the issue of a virtual world’s appearance, it does show how we can abstract the meaning of a virtual world experience from the virtual world itself.

Beyond the original application, we’ve also used the PIVOTE system to author exercises as varied as teaching young people about knife crime and for civic emergency management (see Figure 1). PIVOTE-based projects also won the UK’s Times Higher Education Award 2009 for Outstanding ICT project and the 2010 US Federal Virtual World Challenge for skills building.

In an extension of PIVOTE, Nabil Asif from the University of Surrey proposed a “kitchen designer” approach to virtual world scene creation. This approach uses a Web-based tool to lay out objects from a set of inventories on a 2D grid. Inventories are based on object types (furniture, plants, and so on) or use cases (such as a hospital, office, or factory). Once the user is happy with the layout, he or she presses the “build” button and the system builds the scene in the virtual world. The prototype system for this (Object Positioning and Location, or OPAL) uses a modified form of IMML for storing the layout and MVP for storing behaviors. Figure 2 shows an example OPAL scene in a Web editor and in Second Life.

Evaluating the Approach

From the descriptions of IMML, CityGML, PIVOTE, and OPAL, we can clearly see ways in which we can represent a world semantically — in terms of its layout and behaviors — rather than in terms of its visual look. Such an approach does have challenges, including:

- the need to keep object naming in the virtual world in sync with object naming in inventory and asset files, and
- the need for accurate placement of objects on the Web screen.

Considerable advantages also exist:

- the virtual environment can be rendered across a wider range of devices and platforms;
- the approach simplifies scene layout and behavior authoring;
- users with different interface (and even aesthetic) needs can receive custom views;
- the approach improves accessibility for those with disabilities (particularly visual);
the definition of the environment can be stored, converted, transferred, exchanged, and sold more easily;

• the definition of the world can be generated procedurally; and

• the approach helps autonomous avatars deal more easily with the virtual environment.

Several approaches to virtual world standards are being pursued, but semantic markup languages hold promise for separating scenes’ meaning from their graphical (or audio) rendering. Although infrastructure standards (which can include a semantic element) might eventually provide a universal standard for virtual worlds (and are likely to be wider in scope than just the interaction and representation problem considered here), semantic markup languages represent a flexible way of capturing a virtual world’s content and even intent right now, without forcing a common graphics standard.

References


David Burden is the founder of Daden, a UK-based virtual worlds and virtual character solution provider. He has worked with virtual worlds since the mid-1990s and has a special interest in the use of artificial intelligence and autonomous avatars within virtual worlds. Burden is a member of the Institution of Engineering and Technology and is a Chartered Engineer. Contact him at david.burden@daden.co.uk.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.