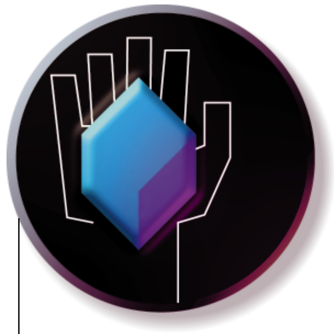


World vision

Benjamin Woolley takes a fresh look at VRML 2.0 — using Moving Worlds technology, it lets you use 3D models and simulations on the net.

When I last wrote about VRML 2.0 (the second version of the standard for using 3D models and simulations over the internet), I got into a spot of bother. I described the decision by SGI and Netscape to declare the Moving Worlds technology as the best basis for the new standard as "pre-emptive".

The two companies had announced the support of just about the entire 3D industry (excluding Apple, who is now on-side, and Microsoft) before there had been much discussion about the alternatives. My remarks were posted to the VRML newsgroup, where they provoked some sharp criticism (and a little support which was, for some reason, offered anonymously). It seems that a few members of the group did not like the suggestion that the VRML community was being, or could be, manipulated. Each proposal for VRML 2.0 would be assessed on its technical merits alone.

In the event, Moving Worlds was voted in as the new VRML standard by VAG, the presiding VRML Architecture Group. Microsoft's ActiveVRML, Moving Worlds' main contender in a field of six proposals, attracted a large negative vote. Obviously, VAG members felt that Microsoft's hold over the internet would develop very nicely without help from them.

Since the final draft of the standard was formally adopted on 4th August, VRML 2.0 has had a chance to get a toehold on the web, helped by a substantial presence at this year's Siggraph show in New Orleans. So this month I thought I would put all past disagreements behind me and have another, more thorough look at VRML 2, its capabilities and its future.

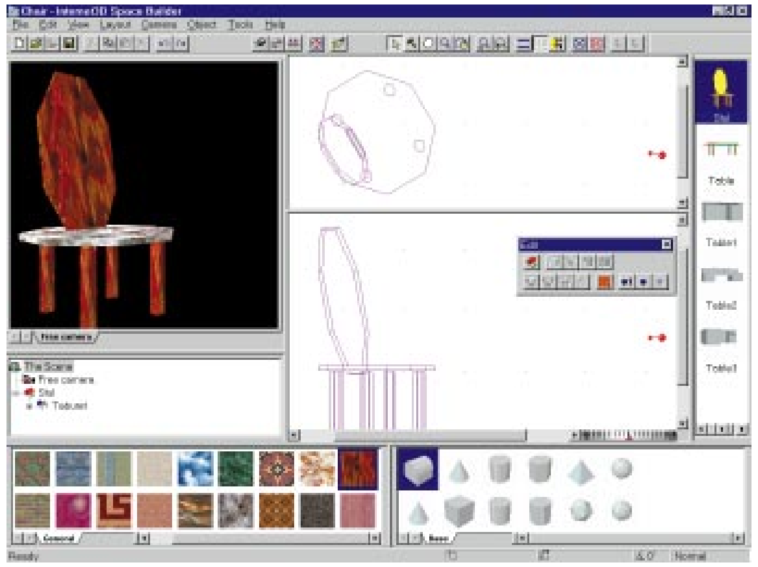

Chair - Internet3D Space Builder
File Edit View Layout Camera Object Tools Help

Fig 1 Paragraph's Internet 3D Space Builder

A better world

The standard is ambitious, promising to provide "a richer, more exciting, more interactive user experience than is possible within the static boundaries of VRML 1.0." There are five major areas of improvement: "enhanced" static worlds, interaction, animation, scripting and prototyping.

The enhancements to static worlds include the ability to put in backdrops, fog and bumpy terrain. Rendering scenes with fog may turn out to be too processor-intensive for all but the most powerful systems so, for a while, I don't expect to see it used much. But the backdrop facility which places a bitmap, such as a landscape, into the scene's background could prove to be a useful and simple way of adding a little more colour and character to a world.

The key to the interaction improvement is the concept of the "sensor". There are various "geometric" sensors that are triggered by events in space, and a sensor triggered by events in time. When a sensor is triggered, it can invoke some other node to be executed (a node is the VRML term for a programming command — see the *PCW* May edition of this column). An example of a geometric sensor is the ProximitySensor node. For this, you define a box-shaped region in space. If the user enters this space while navigating through the world, an event is triggered (for instance, an object in the vicinity becomes animated).

A particularly important enhancement to the VRML sensory environment is the introduction of sophisticated collision detection. A collision node prevents the user, or more precisely their "avatar", entering either specified geometry or all geometry in the scene. In particular, you can ensure that the user does not plough into uneven terrain instead of walking over the top of it.

In the field of animation, the third area of VRML 2.0 improvements, there is a whole new class of nodes called "interpolators", which can be used to alter an object's colour, position, orientation and size as well as other features.

Objects can also be controlled and given behaviour using scripting, the fourth main improvement. Interestingly, the VRML 2.0 specification does not specify which programming language should be used for scripting. The standard specifies that the language is one supported by the browser being used to view the world. This, of course, currently means Java but perhaps one day a developer will come up with an alternative that is tailored to 3D animation and simulation?

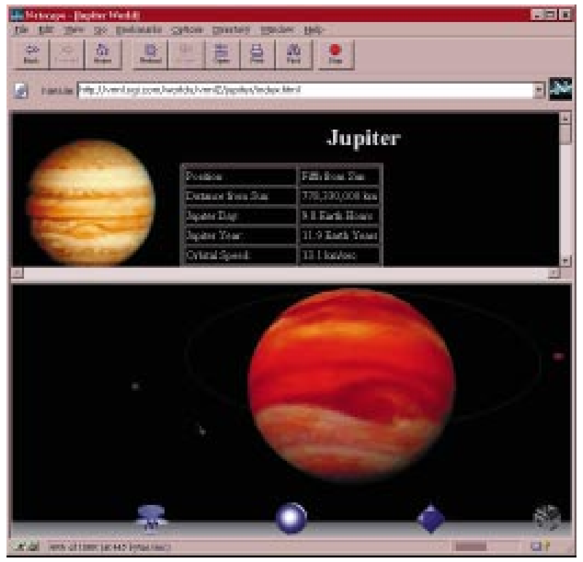
The final area of improvement is "prototyping", also known as "extensibility". VRML 2.0 allows new nodes to be created out of existing groups. You imagine this will typically be used to create nodes for complex objects. Since it is possible to pass parameters and event information to and from these prototype nodes, they can be controlled just like any other.

These and other enhancements have turned VRML from a basic 3D scene description language into a sophisticated animation and simulation programming tool. This should mean that, as promised, it can provide "a richer, more exciting, more interactive user experience". But in addition, it could also mean that creating these user experiences will be much more of a complex business. The VRML specification contains concepts and jargon that all but the most competent programmer will find daunting.

None of this is the fault of the standard's designers. They have tried very hard to make the full specification accessible and understandable. You can download it from the VRML home page (vag.vrml.org). It is a 1.5Mb file that has been compressed using the Unix "tar" format and you will need a utility like WinZip version 6.1 to decompress it. You will find it very well laid-out in HTML format, with convenient links for jumping between the various sections. There are a few tutorials, one based on a Siggraph '96 session (at www.sdsc.edu/siggraph96vrml/) and a couple available through SGI's VRML server (vrml.sgi.com/experts/vrml2tutorials.html). Of course, by the time you read this, there may be more.

For those with neither the time nor inclination to tackle such complexities, it is worth trying out some authoring tools that are beginning to emerge. You should discover a list of those online at the starting point for all VRML work, the VRML Repository at /www.sdsc.edu/vrml/. At the time of writing, only two tools were listed that supported VRML 2.0: Internet 3D Space Builder and Virtual Home Space Builder, both from Paragraph.

I tried Internet 3D Space Builder (Fig 1) and can report that it is one of the neatest, nicest 3D apps I have yet downloaded from the internet (from Paragraph's web site at www.paragraph.com). I was using a beta version that had no documentation and did not support all of VRML 2.0's features like animation but in its basic design it worked like a dream, allowing me to build worlds out of primitives and a small collection of more complex objects (mostly office furniture) simply by dragging them into the scene. A preview window showed what the result would look like in a browser and even allowed me to drag textures straight onto the surfaces of objects.



Position	Fifth from Sun
Distance from Sun	778,330,000 km
Jupiter Day	9.8 Earth Hours
Jupiter Year	11.9 Earth Years
Orbital Speed	13.1 km/sec

Fig 2 SGI's Jupiter demo: the lower panel in the window provides a 3D tour of Jupiter and its moons, while the top panel delivers the facts

It's a small world

Before having a go at building your own world, you might want to see what others have achieved so far. At the time of writing, there was little to see. Unlike Java or Shockwave, you sense a reluctance among content providers to use VRML and you can understand why.

The hardware is not yet in place to make realtime 3D a credible form of communication. Many people still have 486s, most have 8Mb of RAM or less and hardly any have 3D acceleration, or connections faster than 28.8bps. This means that the simplest VRML world or object behaves as if its batteries had run down. VRML tends to look ugly, as well, because the detailed textures are too heavy on resources.

Nevertheless, there are a few demonstration worlds around (see the Jupiter example in Fig 2), hinting at the riches to come. I viewed them using the beta 2 version of SGI's CosmoPlayer, which was, at the time, one of only three browsers listed by the VRML Repository as supporting VRML 2.0 (Netscape's first version of Live3D is a subset of VRML 2.0). I also tried a couple of scenes created using a Doom-to-VRML 2.0 converter. The results were incredibly slow to load and run, but suggested one possible source of material that would look good once 3D accelerators become more commonplace.

There remains some debate about whether VRML is the way to go with 3D on the internet. Various companies are touting alternatives. According to the graphics industry newsletter *Wave*, there is growing interest in using the so-called "DIS-Lite" standard as an alternative. DIS (Distributed Interactive Simulation) is a protocol developed by the American Department of Defense for networked simulations of battlefield operations. The companies that are working in this field, like Mak (www.mak.com), see a lite version of DIS as being the most effective way of building up a new generation of internet simulations and games.

The VRML Architecture Group may have voted on its vision of the 3D future but I sense that when it comes to the wider industry, the jury is still out. The time when VRML enjoys the same sort of global acceptance as HTML, or the same level of commercial support as Java, will be like a complex 3D world downloading onto a 486 via a v34 modem — slow in coming.

PCW Contacts

Benjamin Woolley, writer and broadcaster, can be contacted at **woolley@illumin.co.uk**. His home page is **www.illumin.co.uk/woolley/**