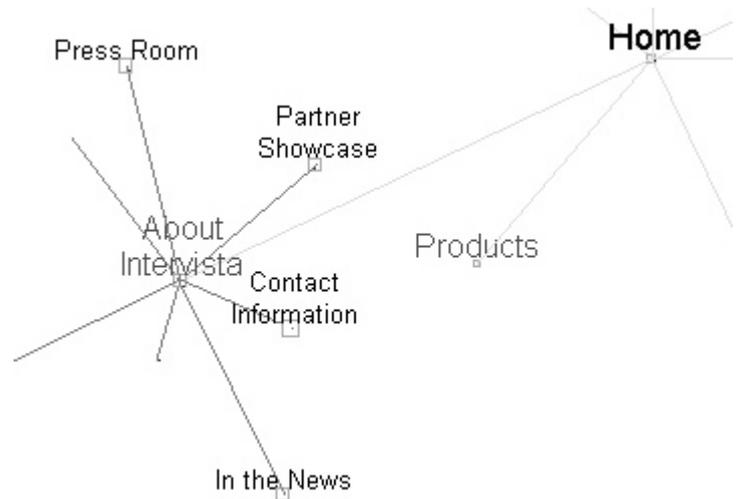


Visual XML

Authoring, Publishing and Viewing Structured Information

Dan Ancona
5.19.98
7.16.98 revised

God is not afraid of our collective imagination.



Overview

Visual XML is a way of describing websites and other structured information spaces in XML and publishing them in VRML. The images on the right are what a site, a page/node and links designed with the current VRML proto implementation look like. This whitepaper will describe how it works, some of the features, and some impacts and implications of three dimensional information spaces, including the benefits to end users. The terms information space, web space, and web site are used interchangeably in this paper, but all refer to structured information: nodes of information that are related or linked together. Examples of structured information spaces include presentations, many types of document, and websites.



The exact implementation of Visual XML has yet to be determined. The VRML could be created on the server side by a Perl script, or on the client side using ActiveX controls or Javabeans. VXML could be a standalone format (similar to RDF), or it could function as a 3D stylesheet. One of the main objectives of VXML is to give authors and designers an easy way to experiment with conveying information in 3D, using a simpler and more direct syntax than straight VRML.

Visual XML is intended to pave the way for the eventual replacement for HTML. It is intended to be a general front end for XML. The web has many problems, and lacks much of the functionality imagined by its conceptual but unimplemented predecessors like Vannevar Bush's Memex and Ted Nelson's Xanadu

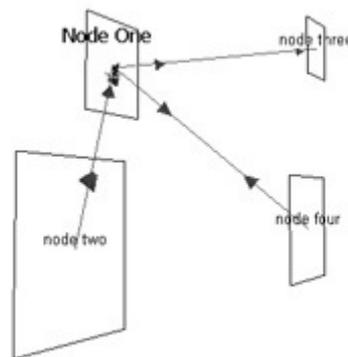
project. VXML addresses a subset of this missing functionality and brokenness with the current web:

- Relationships between nodes (web pages, data, or information) exist, are designed, and are implied, but users almost never see them and even if they could, they can't change them.
- General navigation problems: Web users generally have almost no idea how big a web site is, how much information is there, or what types of media are being used. They also have only rudimentary ideas of where they've been and where they can go.
- Totally representational and totally abstract worlds both have their problems. Mcubed combines the best of both.
- Collaborative document creation with current interfaces is basically impossible. Most people still print out pages and link in their changes with a red pen.
- Media integration on a 2D page hardly makes sense, when even existing hardware is capable of more. 2D pages are a medium to be integrated, not a medium for integration.
- It makes more sense to have 2D objects in an intelligently designed 3D space than to have 3D objects inside 2D windows.
- All the problems with HTML: It doesn't degrade well. It's exceeded its designed capacity. Writing software to deal with it is painful. It isn't extensible. XML is really attacking these problems; Mcubed is just a general way of displaying XML.

VXML offers benefits to end users of the information space. Most classical 3D visualizations have been used to display complex data, but VXML is designed to permit the display of something far simpler: relationships. This is accomplished by using the simple visual linking structures described in linkspace.

The VXML protos are 100% pure ISO standard VRML. VRML is backed by a consortium of more than 35 companies that have made the commitment to interoperability. The VXML protos have only been tested in Platinum technology's WorldView VRML browser, and they use some spec-compliant browser extensions currently available only in WorldView 2.1. More information on WorldView is available in References. The screenshots shown here are from working demos, which will be available on the web soon.

The rest of this whitepaper will talk about the details of how VXML works, including samples of VRML code and what the corresponding XML might look like. Non-technical readers may want to briefly scan these parts of the whitepaper and skip to the Impact and Future sections.



Site Space Description

This is what a sample site space looks like at a distance. This is the first thing that a visitor to a website would see when they visit the site. Since it's just a framework for the site, the download size of this structural representation is small.

In a directory or on a site, this file would typically be saved as home.wrl. A 3D-only web browser would look for this file the way that 2D web browsers and servers typically look for index.html in a directory or on a site.

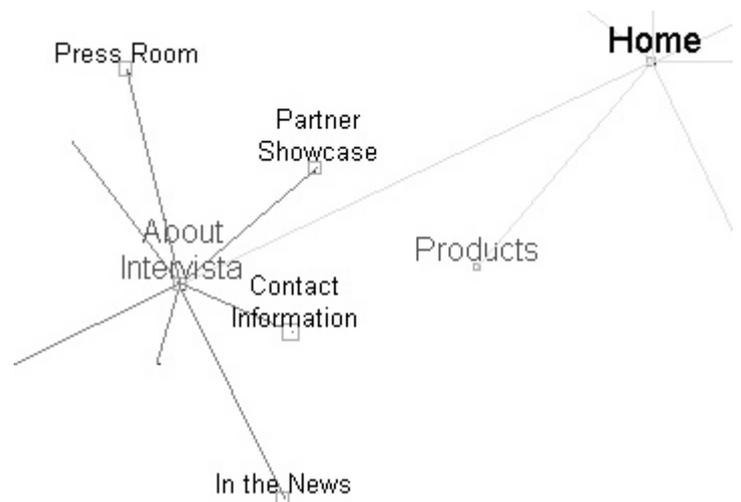
Mcubed separates site design from page design from data, and yet all of these *can* be lumped together if the author desires. Although it horrified old-school SGML hands and created vast angst for site developers when HTML started to be used for serious applications, it was this very lumping together of style and data in the first place that allowed HTML to be adopted so rapidly and so widely. The problem wasn't that HTML allowed this, it's that there was no facility to do it any other way. With VXML, affordances are built in so that it can work just as a 3D stylesheet. Content and design can be separated if the author so desires.

A VXML file format could easily be filtered into a 2D page description language, either print or screen based. The main benefits to the end users of VXML

This is what the VRML looks like for placing a node in the site space. This is just an instantiation of a PROTO, which basically means that the hard part of doing the VRML has already been done. Even this first preliminary pass at the creation of this PROTO will try to do the right thing in its default settings.

This is what the corresponding XML might look like.

The nodeUrl field in the VRML or in the XML determines what is loaded when the user navigates (either via a link or via free navigation) into the node's space. What that might look like is described on the next page.



```
NodeHolder {
  ID "n0"
  position 0 0 0
  vptitle "welcome"
  titleText ["Welcome to Intervista"]
  nodeUrl ["n0.wrl"]
}
```

```
<node ID="n0" position="0 0 0" vptitle="welcome"
titleText="Welcome to Intervista" nodeUrl="n0.wrl" >
```

Node Space Description

This is what a page space or node looks like.

It's roughly equivalent to a page in HTML.

There are five main things an author can put in a node space:

- sound
- image
- shape
- text
- movies

These basic node building blocks could be easily extended using Java that knows how to process the XML and turn it into VRML. Anyone wanting to create an extension could do so by hacking the appropriate VRML, writing some Java code to do the translation out of the corresponding XML, and putting it on the net so that other non-Java/VRML capable authors could use it.

The lower figure shows how the page can be freely navigated. The text on this page uses Intervista's WorldView's feature called PopupText, which allows high quality text to be always facing the viewpoint. The other geometry can be examined normally.



This is what the VRML looks like for the page above. You'll have to imagine the sound.

This is no more difficult and far more powerful than DHTML.

```
img {
  size 0.6 0.6 0.6
  src ["logo.gif"]
  position 0 1.03 0
}
```

```
sound {
  src ["greetings.wav"]
}
```

```
shape {
  size 0.06 0.06 0.06
  position 0 -1.1 0
  src ["logo.wrl"]
}
```

```
text {
  position 0 1.35 0
  pointSize 14
  textColor 0 0 0
  string ["Welcome to Intervista's homepage"]
  style "BOLDITALIC"
  justify "CENTER"
}
```

And it's even easier in XML. This is what I imagine part of the XML might look like for this node space.

```
<nodespace>

<sound src="greetings.wav"/>
<shape src="logo.wrl" scale=0.06 position="0 -1.1 0"/>
```

<text position="0 1.35 0" size="14" color="0 0 0">
Welcome to Intervista's homespace
</text>

</nodespace>

Linkspace

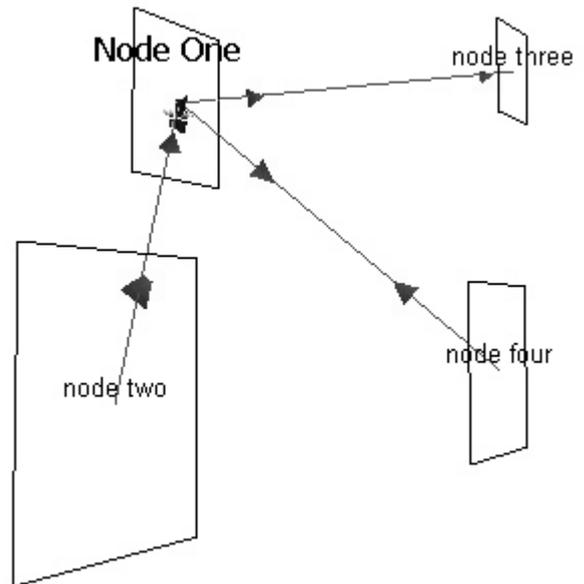
Perhaps most important is what VXML means for links.

In 1945, an engineer named Vannevar Bush wrote an essay titled "The Memex." In this essay, he presciently described a machine that functioned much like today's web. However, with the clarity of thought only available because nothing even remotely like the Memex had ever been implemented, he also described lots of linking features that are nearly impossible to implement using basic HTML. Ted Nelson, who coined the term hypertext, also has proposed many different linking structures and further defined Bush's ideas.

I propose calling these linking structures Bush-Nelson links in their honor. XML is a method of implementing Bush-Nelson links. Project Mcubed is a way of visualizing that implementation.

The image to the right is four simple nodes connected by three bidirectional jump links. A user positioned at one point in the space is smoothly navigated to the end point when they click on the blue link trigger. If these were another type of link- bidirectional collapsing links, for example- when the trigger is activated, the two nodes involved could move to alongside each other.

When Ted Nelson saw an early demo of this at Hypertext 97 in April 1997 (the floating homepage demo, which I also showed at the VRML BOF at SIGGRAPH 96), he went bananas.



This is what the VRML looks like for a link. All I've done so far are bidirectional jump links, but there are *lots* of other possibilities for linking structures:

- links that fire automatically (useful for demos)
- links with exits that can be traversed at varying velocities
- invisible links
- links that disappear or appear when text or a bit of geometry is moused over or otherwise interacted with
- multihead/multitail links (think about a hub at the geometric center of all the connected nodes)
- collapse links, that bring the nodes they connect closer together when activated

These are just the ones I've thought about so far. Implementing any of these using VRML shouldn't be too difficult.

Of course, links are easy to describe in XML, too.

Just the node names can be given in the XML if the tool that publishes the VRML from the XML is even remotely intelligent.

Plus, links can be stored out of line with XML. Different arrangements of links can be applied to the same raw information. A visual implementation of the Xpointer specification would allow links to address a document at any granularity.

```
biLink {
  ID "L0"
  startPoint 0 0 10
  endPoint 9.511 0 3.09
}
```

```
<biLink startNode="n0" endNode="n1"/>
```

```
<autoLink startNode="n1" endNode="n2" delay="0:05.00"/>
```

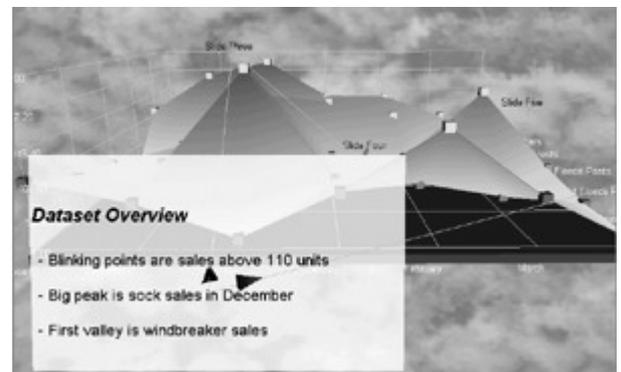
```
<collapseLink>
  <node ID="n3" weight="0.6">
  <node ID="n4" weight="0.3">
  <node ID="n5" weight="0.4">
</collapseLink>
```

Impact

Georeferencing, Spatial Self-Organization and Advironments

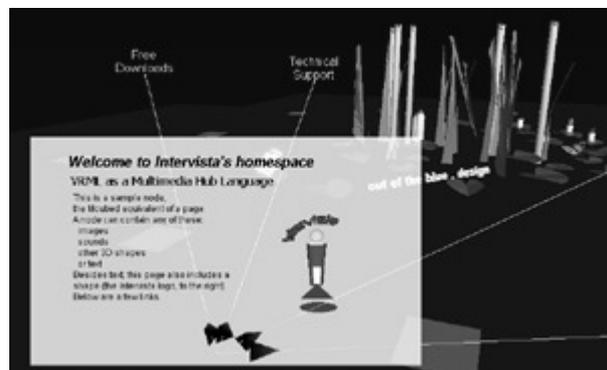
Impacts of this work are expanded possibilities for standardized georeferencing of metadata, spatial self-organization of the web in forms like web rings, advertising, and the effect that this interface could have on web communities.

By georeferencing, I'm referring to the general act of placing data at specific locations in space, not just the act of associating data with geographic locations. Examples of the former are the many attempts at using representational spaces to organize data, such as file folders in Windows or the desk and hallway approach of Magic Cap. A good example of the latter is being carried on at UCSB's Digital Alexandria project. The problem with representational spaces until now has been that real architectural spaces usually make for dreadful ways to access data.



Bush-Nelson links implemented in VRML overcome this by allowing meaningful navigation among the links regardless of their positions in space. Data can be georeferenced, but still navigable by association. VXML is a tool for allowing actual information architecture to occur, if that term is taken to literally mean the placing of information in space.

The substrate space that the data is referenced to can be either representational (as in the butterfly advironment example) or abstract (as in the surface chart example).



Spatial self-organization refers to the rather recent occurrence of structures like web rings. Until now, they've lacked a visual interface, but such an interface (as well as other structures) is quite possible using Mcubed.

Another interesting aspect to VXML is the fate of the ad banner. The image above is of a VRML world created by the design firm Out Of The Blue. Its an example of what they call an Advironment. Trapping such a gorgeous, immersive piece of work inside an 100x400 pixel banner is a shame, and Mcubed provides a solution to this. With georeferenced Mcubed nodes and links, the data could be arrayed inside the Advironment world just as easily as that world could be included in a nodespace.

The main short term end-user benefits of VXML are the usability, visual impact, flexibility and extensability of the web and information spaces. But the main eventual benefit of this interface may be to interactive web community sites. Frustration with such sites was one of the main motivating factors for this work in the first place; I want to be able to annotate other people's comments, link them together, see the whole conversation and where the most interesting and most dense conversastions are happening. More details and a demo of how this might work is forthcoming.

Future

This is the beginning of a work in progress. I'd like to get input and commentary from a wide variety of people on how to procede from here.

Some outstanding issues:

- **Extensibility:** I think this might point to a general solution of the extensibility problem, but I haven't resolved the details of how it might work by doing

a sample implementation yet. But since VRML is a rendering language, and not a markup language per se, it is far more flexible and open to extensibility than HTML ever could be. For now, the (probably java) code to take a piece of extended Mcubed XML and render it in VRML could simply be pointed to as a resource from inside the XML. A good first demo of this might be to do a simple table implementation, where the handler code determines the positions of all the elements and adds any lines or borders.

- **Degradability:** Authors using VXML will be able to make pages that should print out well, either to a 2D screen browser using DHTML or to paper using Postscript. Out of line links could be added in the margins in print or stored in a separate window or page using DHTML. Converting information to a format readable by a handheld device shouldn't be too hard, either. A thought experiment: imagine 3D, 2D, print, and handheld versions of this whitepaper.
- **Collaboration and Enabling Community:** One of the main applications I see for this technology is for enabling community and collaborative work. This ranges from multi-user and avatar scenarios, to improved interfaces to the sprawling community sections that many sites are offering. But beyond offering a better interface to software like WellEngaged, I see the possibility of offering a better and more interactive interface to the web itself, based partially on the client side as well as the server. I'd like to be able to annotate any piece of text on the web and share those annotations with friends or coworkers.

The VRML files and protos used to create the screenshots and demos will soon be available. Please [contact the author](#) for more information or a live demo.

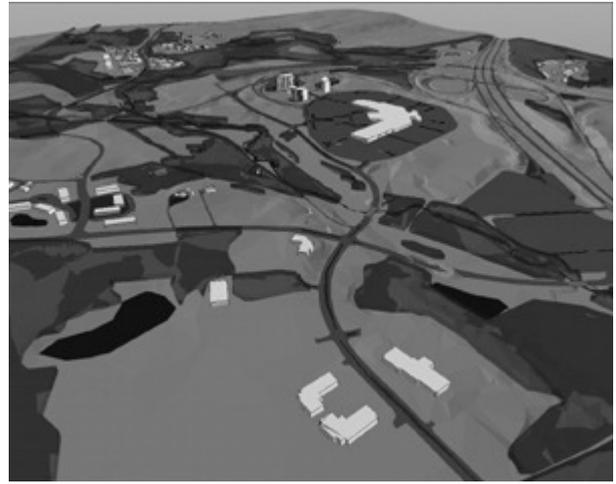
History

Pretty much every project I've worked on for the past four years has been somehow related to Mcubed. I've realized this in retrospect, and I'm psyched; four years ago I picked a really hard problem, and now I've finally got a toolkit to go about finding some solutions to it. This is a sampling of the most relevant projects. For now, only stuff I've been working on is listed. For the larger context of this work, see some of the annotations in References.



Owings Mills, MarylandFall 1994 - Spring 1995
For IATH using PolyTRIM (Centre for Landscape Research software)

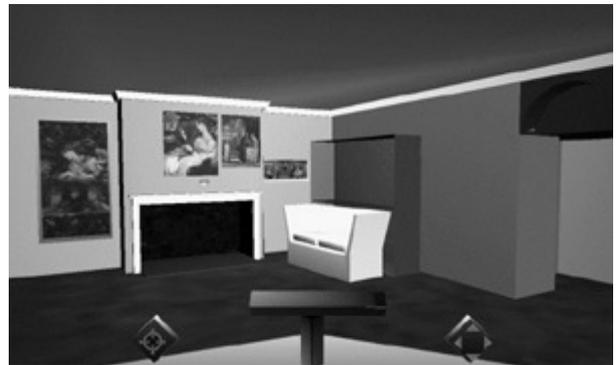
A large scale landscape visualization. It was my first real time work and luckily no one was around to tell me that you couldn't move 10M polygons on an Indy in real time. We settled for the rotten frame rates.



Rossetti RoomSummer 1995

For IATH using hand-coded VRML

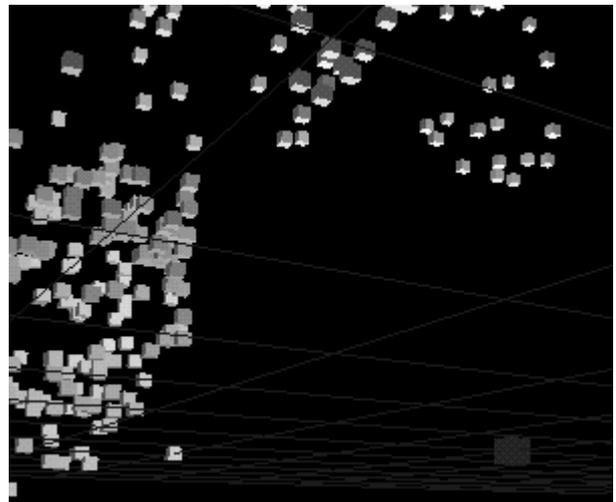
My first VRML1.0 world, created by a CGI script. To the extent that you could click on the images in the world and bring up a page of information about the painting, it was an information space.

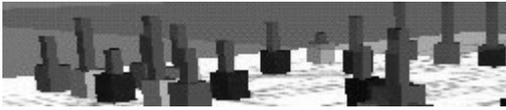
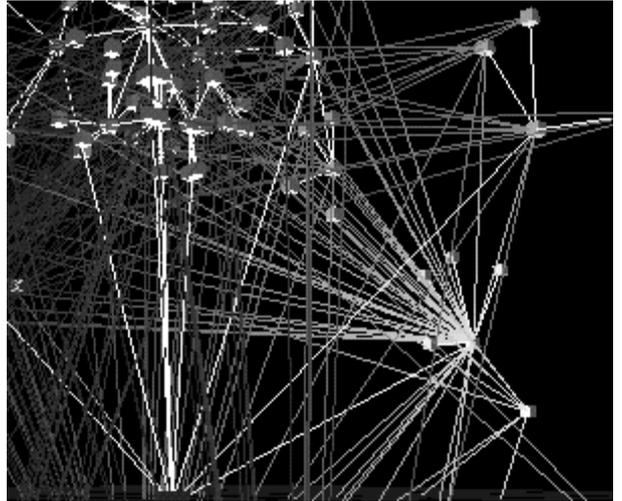
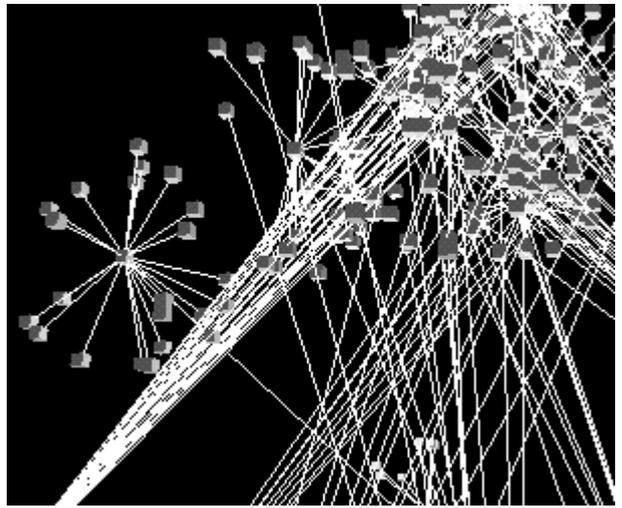


Invisible CitySpring 1996

For my undergraduate thesis, using perl-generated VRML

My first attempts at site space visualization.

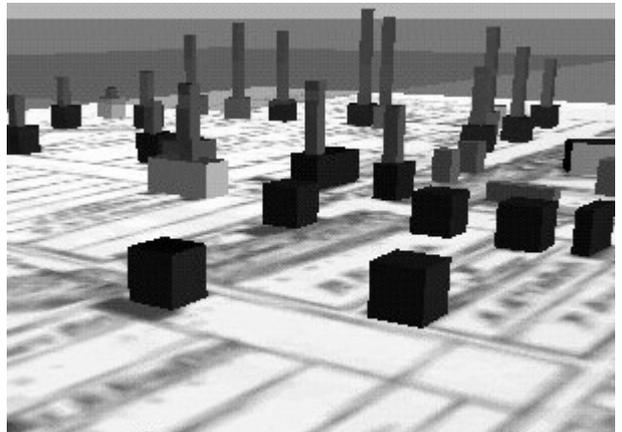




Urban Visualization Prototype Summer 1996

For IATH, using Cosmo Worlds

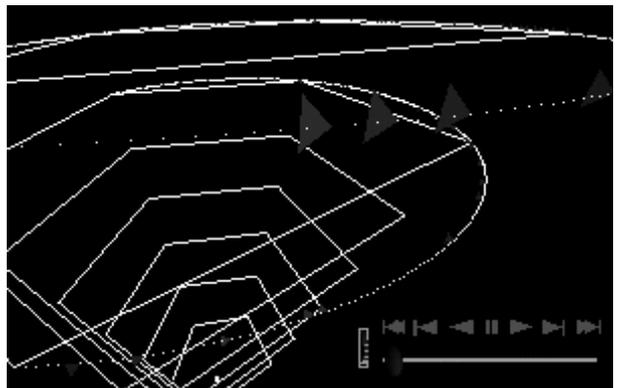
My first georeferencing prototype. The idea was to associate data (household income, political affiliation) with the various houses. If I'd understood SGML at the time, I would've understood how to make the plumbing for this work.



Dante's Inferno SGML Viz Spring 1997

For IATH, using perl-generated VRML from SGML

My first SGML work, a big breakthrough for me personally. The lines in this image represent the 34 cantos of Dante's Inferno, and the points indicate individual lines. The triangular markers represent occurrences of tags that were selectable via an HTML form. The resulting VRML world could then be



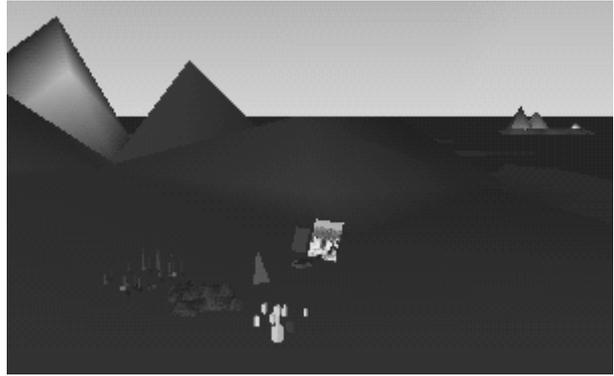
navigated via a VCR widget.



Representational Website Interface Spring 1997

For IATH, using Cosmo Worlds

The first time I tried mixing representational and abstract information in the same space. It was still a map, but it was a step in the right direction.



[no image available]

AEP Presentation World Summer 1997

For IATH, using Cosmo Worlds

My first Powerpoint in Cyberspace attempt. It was pretty cool looking but unfortunately, the file and all the screenshots that went with it were destroyed in the June 1997 disk crash.

Author

Current contact information for
comments and inquires regarding this work:

Dan Ancona <dan@ultrablue.com>

work 415.543.VRML x 265

cell  415.806.9773

page  415.807.1052

<http://www.ultrablue.com/dan/>

This work will be presented for discussion
in the enterprise working group meeting at
SIGGRAPH 98, and possibly at the Web3D
Roundup. See vrmf.org for more information.

Bibliography

(annotations, organization & further details coming soon)

Lipkin, Daniel *Integrating XML and VRML: A Technical Discussion*

www.xml.com

Intervista

Feed's Memex Document

Feed's Xanadu Document

Carey, Rikk and Bell, Gavin *The Annotated VRML Specification*

Light, Richard *Presenting XML*

St. Laurent, Simon *XML, A Primer*

Acknowledgments

Tons of people have already helped with this project so far: questioning my reality, giving me server space, raising me, teaching me a thing or two, and getting it when I needed someone to. A partial list, in no particular order: Jim P., Julie, Sara, Tony, Clay, Myron, Chuck, Marisa, Linda, Len B., Virtual Real Estate, www-vrml, Taylor, most of the biancanauts at one time or another, IATH, and of course, Mom & Dad.