Chapter 2: Connective, Collective, Corrective: The Birth of VRML

The World Wide Web is the quintessence of collaboration; as a tool, it inspires the coming together and juxtaposition of disparate pieces into an integrated whole. The software foundation of the Web has evolved similarly - sharing has been the driving force behind the rapid development and propagation of Web-based technologies. First CERN created "libraries" of the World Wide Web source code; NCSA Mosaic has always been available (in its UNIX incarnations) in source code form, so that anyone could freely modify the functionality to suit specific needs. The presumption is that these enhancements would in turn be returned to the community - as does happen - and that would broaden and accelerate the entire developmental process of the Web. Within the Internet, and specifically within the Web, *a resource shared is a resource squared*.

In October of 1993, I downloaded NCSA's Mosaic program, along with their World Wide Web server application. The server, NCSA *httpd*, allows a UNIX workstation to act as a Web server, that means it can respond to requests made of it by Web browsers.

Both Mosaic and httpd are relatively easy to install on a UNIX workstation, and within a few hours I had my very own Web site. I spent many hours that fall surfing the Web - small enough that it seemed possible to visit every site - and it quickly became clear that the global hypermedia system we'd been promised for years by Ted Nelson had finally come to pass. Unfortunately, Nelson's vision for *Xanadu*, a comprehensive hypermedia system, never came to pass. In development for almost 20 years, "Project Xanadu" collapsed into a wreck of finger pointing and name calling after Autodesk Corporation withdrew the project's financial backing in 1993.

Amazing as it was - I had an epiphany which has never really ended - I immediately sensed that the Web needed a three-dimensional interface. The text and pictures made for a pretty place, but was it really useful? If I had the Library of Congress, with hundreds of millions of volumes in its collection, how could I browse through it on a screenfull of text? The Louvre was one of the first major Web sites, but even it was reduced to images glued to a Web page.

Architecture is art; space and place are essential to our civilization, our language, and our being. If we were really building the global info-organism - something I recognized from the moment I got onto the Web - we'd need to make it as sensually rich as human experience is. Text is a very recent thing; the printed word has five hundred years of history behind it. Compare that with half a million years of human language. What would the Web be if we couldn't recreate the cave paintings of Lascaux? Or Chartes Cathedral? Or Frank Lloyd Wright's Fallingwater?

We are creatures of sensation; here in the West we depend upon our eyes to tell us most of our stories. But we have ears to ear, and skin to touch, space to move through. Bringing that to our Web would make it more human, and more like ourselves. If we want our global mind to be a mirror for our being, shouldn't it be perfectly clear?

I knew we could bring the Web out from the flat text and images into a new sensuality which would be more fluid, expressive, and emotional. I set to work.

Fortunately, the Web, built upon source code that I could download and compile on my own computer, lent itself to rapid modification for specific purposes. Within a few months, I had the basic designs for a three-dimensional Web browser completed - but only in my head. I needed some help to write the actual code because a 3D Web browser would need a new language - the equivalent of HTML - to serve as the bedrock of cyberspace. HTML is well suited for text, and can, when pressed, even work with images. Cyberspace defines the layout of complex objects in a three-dimensional environment; that was completely beyond HTML's capabilities. I knew I'd need to create a new language, one which could speak to the Web and cyberspace simultaneously. I could design the language all day long, but I didn't know how to implement it.

Enter Tony Parisi. In December of 1993, Tony moved to San Francisco with his wife, Marina Berlin, to start a new life in the promised land of Northern California. Tony is seven days older than me (we see a lot of things the same way as a result), and has a strong background in desktop software. At Lotus Development, he helped write the Microsoft Windows version of *Lotus 1-2-3*. When I met him, he was developing new computer languages and user interfaces as a Principal Engineer for Belmont Research Corporation.

As Tony says, he stepped off the plane and into cyberspace. On New Year's Day 1994, sitting among the packing boxes in his new apartment, I gave him the full indoctrination: about the Web (best thing since sliced bread), about virtual reality (the essence of VR is communication and experience, not reality and immersion), and about the importance - as I saw it - of bringing them both together. We're both excitable Italian-American kids; before we knew it, we were working out the details of the design of a 3D interface to the Web.

At the time, Tony thought he'd humor me. It wouldn't take very much effort, and it it would be a neat hack, *if* we could pull it off. In mid-January, my day job as a consultant vaporized when the contracting company that suddenly collapsed - this happens all the time in Silicon Valley - and I found myself with some free time.

I had my hands on another important piece; a very nice 3D *renderer* (a renderer draws 3D objects onto the computer screen) called *Reality Lab*, from a British start-up company named Rendermorphics. A few weeks before I met Tony, I was introduced to Servan Keondjian and Kate Seekings, the founders of Rendermorphics. Although I had no money to pay for a Reality Lab license, they believed in my still-forming vision for a 3D Web, and let me use their rendering library at no charge. (Rendermorphics was acquired by

Microsoft in February of 1995 and Reality Lab is now a core component of Microsoft Windows.)

For his part, Tony wrote a *parser* (a computer language translator), which converted from a text file into a set of objects that the computer could understand and manipulate. As I defined this language, he learned a lot about three-dimensional graphics - polygons, normals, shading, and lighting - and I learned a lot about parsers. It was a great collaboration.

By February 14th, 1994, we had created an application which would go across the Web to fetch a 3D object - a banana, of all things! - which would then be displayed in a window on the computer screen. I could move my mouse over the banana, click on it, and - ABRAHADABRA! - NCSA Mosaic launched and loaded my home page on the Web. We had successfully created a 3D interface to the Web. Even more, we could click on a link in Mosaic and have it fetch a 3D object, then Mosaic wourld launch our application, in which the object would be displayed. We could go from the 2D Web in HTML to the 3D Web and back again!

We named our embryonic little program *Labyrinth*, because it took a twisty place that had no space -the World Wide Web - and gave it depth.

Later in February, while searching the Web pages at CERN about the World Wide Web Project (all of the people developing Web software pooling their knowledge and resources), I noticed that Tim Berners-Lee had a number of pages talking about using virtual reality with the Web. I sent him electronic mail - excitedly sharing what Tony and I had done - and in March I found myself invited to the First International Conference on the World Wide Web, at CERN, in Geneva, to present our work.

The air in Geneva that May seemed charged with electricity. The First International Conference on the World Wide Web gathered up the world's Web developers into a four-day whirlwind of conferences, lectures, and informal meetings. Everyone shared their own work, and took delight in everyone else's. Most of the attendees had academic backgrounds and concerns - this early on, the Web had no commercial focus. Despite diverse backgrounds, we all had something in common - the sense that something amazing was taking place in our midst. *This was it.* We'd been dreaming of the Web for years - a way to share our knowledge, everywhere, all at once. Not once did anyone suggest that the Web would soon change the whole world. That was a given.

The conference was filled up months in advance. Even so, people came to Geneva hoping to sneak into one of the sessions. It was standing room only as Tim Berners-Lee gave his keynote speaking eloquently about the need of a constitution for cyberspace. He was followed by Dr. David Chaum who talked about privacy and secure electronic voting, then demonstrated his *DigiCash* system publicly for the first time. The emphasis on security and seemed quite appropriate. *Finally*, we had something worth protecting.

CERN had just one machine set up for electronic mail access! With 350 webheads, that created a tiny crisis. I came to CERN an hour before the conference opened each morning, so I could grab my turn at the terminal set up for that purpose. One morning, as I walked in, I saw conferees from Japan talking with Italians, French, and Germans in a murdered English. For a wonderful moment, I felt like I'd been transported to Starfleet Academy. This made *sense* - the nations of the Earth working together to create an infrastructure for sharing knowledge.

On the second day of the conference, Berners-Lee and David Raggett, one of the primary forces behind HTML's development, held a "birds-of-a-feather" on "Virtual Reality Markup Languages and the World Wide Web." (Birds-of-a-feather sessions, often called "BOFs", are informal meetings of individuals interested in a particular topic.) Raggett had given serious thought to a platform-independent standard for 3D in the Web, and his proposals helped to shape all of the work that followed. The acronym he coined - "VRML" - stuck, although "markup" was later modified to the more appropriate "modeling."

At that session, I met Brian Behlendorf, the UNIX system administrator for WIRED magazine, and the technical brains behind HOTWIRED. He, too, seemed very interested in getting a real 3D Web language specified and into people's hands. He'd just finished a course on computer graphics at UC Berkeley; over the course he'd come to realize the power 3D graphics gave to human communication. Brian wanted to see VRML happen, and sweet-talked WIRED into volunteering their considerable connectivity and computing resources to help create a standard for a VRML.

On the final day of the conference, at the "Advanced Topics" session, I presented a paper simply titled, "Cyberspace". I'd hastily rewritten some of my slides to use the acronym VRML wherever appropriate. In the demonstration that followed the session, people got their first taste of a 3D Web; I showed my cyber-banana, linked to the paper on cyberspace. It seemed appropriate.

The Web teaches an open philosophy of sharing and development. There are no barriers, no prejudices (ideally), just an open forum for the discussion and development of ideas and applications. We wanted to repeat the success of the Web, so we approached the Internet and Web communities, saying, "come help us design the 3D language for the Web - come help us design VRML." We set up a list-server (a program that forwards mail sent to it to a list of "subscribers") on WIRED's computers, and a Web site which outlined our focus and goals.

I expected that only fifty or one hundred people would be interested in such an esoteric effort. I was way off base. Within a week, we had well over two thousand people who had subscribed to the mailing list we set up to coordinate the effort. We'd struck a chord, and the whole Internet began to vibrate in response. (That week I got more electronic mail than any other period in my life. One morning I woke up to find 750 messages in my mailbox!)

The development time Tony Parisi and I had put into Labyrinth had helped us to define the issues important in a VRML; all that work (just a man-month between us) could be set aside, for a more complete solution, something which could be as expressive as a professional 3D designer would want. Labyrinth was a good beginning, but little more than a demonstration of what the eventual capabilities of VRML should be. While *Jurassic Park* might not be possible on most of 1994's computers, we knew we needed something with commercial potential, and room to grow, otherwise we'd be considered some "hacker's project", garnering little interest outside of academia.

I don't believe in reinventing wheels. The last thing the world needs is another computer language - something else people have to adopt and learn and master. Wouldn't it be more appropriate, I thought, to adapt an existing computer language, one designed by graphics professionals? We could take it and modify it to understand the Web, to link with it. I posed this question to the members of the community forming to define VRML. With so many experts in computer graphics - some with twenty years experience or more - finding an existing solution that could do the job well should be easy enough.

In fact, we soon discovered several candidates that would fit the task. The Object Oriented Geometry Language (OOGL) from the Geometry Center at the University of Minnesota, the Cyberspace Development Format (CDF) from Autodesk, and the Manchester Scene Description Language (MSDL) from the University of Manchester in Great Britain are but a few of the languages presented as potential VRML nominees.

The first suggestion I heard - from Clay Graham, a virtual architect of enormous talent and growing reputation - turned out to be the winner. A programming library, *Open Inventor*, from Silicon Graphics (a high-end workstation manufacturer), used for rapid prototyping of 3D applications, had a scene description language and file format rich enough for commercial quality applications. It could grow - you could add your own features to it, and most importantly, it had been time-tested in commercial production environments. *It was debugged.*

We debated the merits of each of the candidates for a month, but a consensus quickly developed around Open Inventor. I traveled down to SGI, where I was introduced to Rikk Carey, the of Open Inventor's principle designers, Paul Strauss, and Gavin Bell. Gavin had been active on the discussion, serving as an incredible resource, answering questions about Open Inventor, proposing extensions to it, and speaking, on his own, about SGI's plans for Open Inventor.

The Web community is a community committed to open standards. We could not adopt a standard owned by any company. SGI had to be committed to putting Open Inventor's file format into the public domain, without restriction, so that the Web community could build upon it, free from the fear of lawsuits or royalties. The folks at SGI understood this. They wanted to see Open Inventor, even as VRML, succeed. We all knew that if it did, SGI would make a lot of hardware sales, and that SGI would have the inside track on VRML development. At the same time, many man-years of effort had gone into Open Inventor.

SGI had made a serious investment in their technology. To just hand it over to the Web community seemed a questionable idea at best, and lunacy at worst.

But Rikk believed. He worked out the legal particulars within SGI, and later that summer, the subset of Open Inventor that became the core of the VRML 1.0 specification was placed into the public domain.

Now that we had a language, we had to face a number of difficult decisions. How would we extend this language to meet the requirements of the Web? How could we produce a scene description which would have consistent performance across a wide range of computers? How could we allow for interactivity?

All of these questions, fraught with pitfalls, obsessed the VRML community throughout the summer and into the fall of 1994. Perhaps the most important consideration, interactivity, was completely left out of the first specification - distributed simulation is among the most difficult of all problems in computer science. The problem is this: if I'm on a computer running Microsoft Windows, and you're on a Apple Macintosh, and I "hand" you an object in cyberspace, what do you know about that object? The two computers are very different, and really have no language in common. We'd need an equivalent of HTML for behavior; a generic way of describing how things happen. We had zero possibility of addressing this problem in the next year or five.

In the end, the VRML community took its cue from the Web. The Web had no interactivity, but it was still very useful, and quite perfectly of captivating the imaginations of the Internet's users.

On the 17th of October, 1994, at the Second International Conference on the World Wide Web, in Chicago, Tony Parisi and Gavin Bell presented the draft specification for VRML 1.0. It needed a number of small refinements, but covered all of the bases. People could begin to design tools - browsers and editors - that conformed to the specification, and knew that these tools would interoperate.

The VRML session was another standing-room-only event. Mark Andreesen, now vice president of the recently formed Netscape Communications Corporation, sat in the back of the room, wildly scribbling notes. People from NASA and NCSA realized that they already had Open Inventor capable machines, and were a stones-throw away from VRML. People talked about future directions, argued out points of architecture and design, and the session's alotted time came and went, so the eventevolved into the second VRML "birds-of-a-feather" meeting, concluding in dinner for all of the VRML die-hards.

After the fall Web conference, Paul Strauss and Gavin Bell set to work on QvLib, a "quick" VRML parser written in C++, using SGI's base of Open Inventor source code. This was another major gift from SGI; they turned over source code developed by SGI employees for use by the Web community. QvLib reads in the VRML files (that's what a parser does) and produces a set of objects which can be manipulated by the computer.

In December of 1994, QvLib was made available on several computer platforms. The stage was set - *anyone* could now implement a VRML browser.

One of VRML's biggest assets came along accidentally. There are many different 3D scene description languages, none of which interoperate. It's necessary to convert from one to another as you move from application to application. Every time a conversion happens, some important data gets lost. Yet, a designer will often use several different programs when creating a 3D scene, so 3D file conversions a very common. Companies began to look to VRML as a "metafile data format" or universal interchange format for 3D data. Using VRML, a designer could create an object in Caligari trueSpace, place it into a scene in 3D Studio, and render it in SoftImage, all without converting it from one format to another.

In January of 1995, Robert Wiedeman of Template Graphics Software - who resold Open Inventor for non-SGI platforms - convinced SGI that they should build a VRML Web browser. SGI would design the browser and TGS would translate it to other platforms; that began the development of WebSpace. At the same time, Tony Parisi formed his own company, Intervista, and began development of his own VRML browser, WorldView.

VRML was gathering a head of steam, fueled by the 2.2 billion dollar marketing engine of Silicon Graphics, who sought and got endorsements from Ford, Netscape, Digital, and many other companies, hoping to make VRML a Web *fait accompli*. Each of these organizations saw how VRML could be used to sell cars, Web servers, or computers - there wasn't much hard sell. Cyberspace sold itself.

In fact, the press reacted so favorably that several reporters "jumped the gun" and announced the existence of WebSpace and VRML before the worldwide launch date of April 3, 1995. On that date, termed "Day Zero" by Kevin Hughes, most of America learned about VRML for the first time. SGI made a world-wide press announcement and front page (of the business section) all over the country.

The first weeks after that announcement were a heady time; reporters clamored to know everything about the technology that seemed destined to evolve into cyberspace. NEWSWEEK ran a 2 page article on a technology just under 2 months old. But, now that VRML browsers like WebSpace were ready for prime time, I didn't need to talk about what it would be like, I could sit a reporter in front of a computer and *show them how VRML worked*.

The overwhelming response and welcoming of the technology made it clear that VRML was at the right place at the right time with the right approach. Two weeks later, in Darmstadt, Germany, at the Third International Conference on the World Wide Web, VRML evolved from an embryonic technology into a hotbed of Web development. I gave the keynote speech on "Developer's Day", and later fielded questions from researchers interested in making VRML browsers, or VRML Web sites.

A Tale of Two VRMLs: The Tale of VRML 2.0

The first version of VRML, developed in the embryonic days of the Web, had a number of shortcomings; most serious among these, an almost complete lack of interactivity. In 1994 it simply wasn't possible to constructing interactive content on one platform - be it SGI, or Macintosh or PC - in such a way that the content run would on another platform, under a different operating system & microprocessor.

Just four weeks after "Day Zero" all of that changed forever. Sun announced the availability of the *Java* programming language, specifically designed to provide cross-platform compatibility for interactive Web content. It seemed too good to be true - a language truly-tailor made for our needs. Better yet, Java had no innate 3D capability, so VRML had something to offer it, as well. It seemed like a marriage between the two - pairing VRML's visualization capabilities with Java's interactivity - could provide the best of both worlds, and *real* virtual reality, rather than just a walk-through of an unchanging universe.

Tony and I went down to Sun in early May, to meet with the team; Kim Polese, James Gosling, Arthur van Hoff, and a whole host of others. Everyone seemed excited and anxious to make the marriage happen. But one question kept on popping up - how could we change VRML?

At this point, the *www-vrml* mailing list formed the virtual "town square" of the community. Thousands of individuals, highly connected, began to act in concert to work through the myriad technical issues facing us. This *connective intelligence* led directly to our vote on the VRML 1.0 specification. Yet, with all of VRML's shortcomings - and no obvious way to remedy them - VRML 1.0 seemed to have no future. The mailing list could suggest changes in the specification, but the list itself - an amorphous body of individuals - had no ability to translate these suggestions into something concrete.

I'd recognized this situation back in the winter, and had begun to float the balloon of a VRML Conference, a one- or two-day affair where the best minds in VRML could gather, hash out the basics of a next-generation VRML, then refine it through the mailing list. While the idea of such an event had gotten some support, Rikk Carey offered a counter-proposal: why not set up a permanent, self-appointed body to guide VRML into its next generation. Eight, or perhaps ten individuals, dedicated to the task - and fully answerable to the community - could do the job succinctly, without becoming mired in political disputes or "religious" wars about the "right" way of doing things.

It seemed a good idea; in late July, Rikk and I hand-picked eight of our fellows - including Gavin and Tony - to form the VRML Architecture Group (VAG). We had a mandate, and our reputations served to quell the concerns of a community which might otherwise think that VRML had been "taken over" by a cabal of sinister engineers.

At SIGGRAPH '95 in Los Angeles, VRML began to enter the spotlight in computer graphics. As I walked the show floor, I found the IBM booth *completely* given over to

VRML - an auspicious beginning. A few days later, at our second annual SIGGRAPH BOF - now becoming the major "fleshmeet" for the world-wide VRML community - the VAG stepped forward to introduce ourselves to the community we purported to represent. We hoped that public scrutiny it would keep us honest, and, two weeks after the show, we packed our bags for Half Moon Bay - a beautiful seaside town about 40 miles south of San Francisco - for three intense 12-hour days of introductions, presentations, proposals, counter-proposals, assignments, and action items.

Despite the nobility of our attentions - and the approval of a VRML community keenly watching our ever maneuver - design by committee has never proven either efficient or effective. At our next meeting, in October, Rikk and Gavin stunned the VAG by offering up their own version of a complete VRML 2.0 specification - including limited support for interactivity. It set the VAG into turmoil, and we left the meeting arguing amongst ourselves about the merits of their approach - but each of us had a growing realization that the VAG could not create VRML 2.0; the longer we avoided admitting it to ourselves, the more irrelevant the VAG became, and the more uncertain the future of VRML.

I went to Holland for the month of November, to evangelize VRML and speak at some conferences. While there I received an invitation to speak at a Web conference in San Francisco, the day after I returned from my travels. Curiously, I also received email queries from some folks at Microsoft who asked if they could send me a "white paper" about their own work in VRML. I found a package waiting for me when I returned from Europe, a dense document describing Reactive Behavioral Modeling Language (RBML) and its integration with VRML 1.0. It seemed very academic, but - in the light of later events - proved highly relevant.

On the morning of 4 December 1995, I found myself horrifically jet-lagged and addressing a room of twelve hundred people, who had just heard from Marc Andreesen on the glories of the Web. I took a different tack, and suggested that sensibility and sensuality could add more to our Web experience than ubiquity and uniformity; in other words, an interface that disappeared into simplicity might be the best guarantee of the Web's success.

Less than 30 minutes after I walked offstage, I found myself ushered into a press conference where - to my absolute surprise - I found Eric Shmidt, CTO of Sun, Tom Jermoluk, President of SGI, and Marc Andreesen of Netscape onstage together, all extolling the marriage of VRML and Java as "the platform for twenty-first century computing". I was stunned and speechless. Somehow SGI had managed a marketing coup with its biggest - and most feared - competitor, Sun, and had arranged the marriage which we'd been working toward for nearly a year. SGI announced their *Cosmo* line of products - a Java-enabled VRML browser, and tools for high-end VRML production. SUN pledged to integrate VRML with Java as part of the standard. And Netscape pledged to support both.

I ran into Rikk Carey just a few minutes after the press conference ended. He quipped, "Mark, welcome to the wedding." I could only nod, for it seemed as though my wildest dreams had come true - the companies at the heart of the Web had adopted VRML as

their own, and - more than that - sought to extend it in a way that would make it fundamentally useful.

And so it would have been, but for the fact that three days later - Pearl Harbor Day - Microsoft swept in for a surprise PR attack, and announced their renamed-at-the-last-minute *ActiveVRML* proposal, really just RBML with new window-dressing. They touted it as a better methodology for crafting interactive virtual worlds, built from years of research at Sun - where, in a belt-tightening, it had been axed in favor of *Oak*, the forerunner of Java - only to be rescued by Microsoft's Advanced Research Group.

Now we had two contenders for VRML 2.0, each claiming themselves as the legitimate heirs to the original specification. As neither the VRML community nor the VAG had been involved at any level in the creation of either specification, neither specification had any inherent claim to legitimacy - but they could argue about it, and they did - endlessly. Throughout the middle of December, leading up to VRML '95, our pure-VRML technical conference, both camps waged a war of words; sometimes academic (autonomous behaviors in RBML are better because interactions between components needn't be designed in advance), sometimes political (Silicon Graphics has got a choke-hold on VRML and won't let go), and sometimes just plain paranoid (Microsoft is evil and Bill Gates wants to rule the universe). We had a short VAG meeting during VRML '95 where even Rikk and Gavin exchanged harsh words - and they worked with each other! The fabric of VRML had begun to tear under the pressure of so many multi-billion dollar agendas, and we had no consensus on any way out of the mess.

I looked back into history recognized that we had traveled this ground before - at our beginning. We had originally thought of multiple, competing specifications as an advantage, rather than a problem; why couldn't we assume that attitude again, and let the best specification win? It took a some convincing - and a bit of arm-twisting - to get the other VAG members to agree, but, at the start of January, the VAG issued a Request for Proposals (RFP) for VRML 2.0, defining a list of requirements that had to be satisfied in a next-generation VRML, including those qualities already present both in Cosmo - now renamed "Moving Worlds", and ActiveVRML.

A month later, we had ten candidates, among them, proposals from Sun, Apple, IBM, a German technical institute, and, of course, Silicon Graphics and Microsoft. We spent another month debating the merits of the proposals - using *www-vrml* as the forum for discussion. Through all of this the VAG evolved from an organization which creating specifications to an organization which worked to maintain the integrity of an open specification process. But - given the money at stake - the discussion period inevitably became political. Silicon Graphics rounded up supporters - particularly Sony and Paragraph - making their efforts appear more community-based; in a very beneficial side-effect, this also opened the "Moving Worlds" specification to input from parties outside SGI, something which radically improved it. Within a few weeks most of the VRML community had rallied behind it.

What we'd witnessed - and SGI had unwittingly engendered - was the birth of *collective intelligence*; some of the best minds in simulation, in Java, and computer graphics offered up their own pearls of wisdom, creating a "Stone Soup" of incredible flavor. And so, when all the votes had been tallied, over a thousand individuals had expressed a preference - and about 70% of them preferred "Moving Words". Once again, we had the beginnings of a specification, but - once again - months of hard work lay in the details.

Throughout the spring and summer of 1996, most of the VAG worked on the technical details of VRML 2.0; even Microsoft stepped back up to the table again, humbled by their defeat in the specification wars, but holding true to their public commitment to support VRML 2.0 - whatever that might turn out to be.

I struck out on a different project, determined to protect the future of VRML. We'd nearly seen two years' effort lost in the closing days of 1995, because of a dangerous lack of infrastructure in the VRML community; we had no buffer between ourselves and the commercial interests, any of which might have a vested interest in "driving" VRML development. The VAG, as an unofficial body, had no power to reign in those who sought divisiveness in the community; it could only offer patient wisdom against the onslaught of press releases and marketing dollars. In short, VRML needed a backbone, so that it could stand up to any of the forces which sought to manipulate it.

In July, as Rikk and the other VAG members put the final touches on the 2.0 specification, I completed another engineering task - that of creating a VRML Consortium. Consortia are the "meeting places" for commercial communities, a space free from comptetive values, where organizations can coordinate their activities. Consortia exist because every member has a vested interest in the success of every the whole - and every company working in VRML has a vested interest in the success of VRML as a standard. The VRML 2.0 specification, representing thousands of man-hours of work from a world-wide team of engineers and computer scientists, was too important to lose to in-fighting or competitive marketing. At SIGGRAPH 96, in New Orleans, the final draft of the VRML 2.0 specification and the VRML Consortium were introduced *together* - two halves of the same whole. Incorporated in December 1996, the Consortium boasts over 50 members at this writing, and remains open to any organization - academic or commercial - wishing to assist in ensuring the success of VRML.

Built to Last

The VRML Consortium creates an infrastructure for VRML, based around four basic areas; standards and specifications - ensuring that VRML becomes a stable, world-wide standard; conformance and compliance - so that browsers and tools inter-operate effectively; research and development - helping VRML to meet the needs of all the communities using it; and marketing and education - spreading the word, then providing resources for those who want to learn more. The Consortium represents a transition from the amorphous collective intelligence of the community into a finely-honed *corrective intelligence*, capable of recognizing its own shortcomings, and willing to reshape itself as it learns from its mistakes.

The VRML community has begun to differentiate - much like the fertilized egg as it matures, different areas in the VRML community serve different functions. *Working Groups* are the heart of VRML development. Composed of any willing individuals, these groups are chartered around a specific application area - such as Java, or database integration, or object management - they establish a succinct set of goals, then submit their work to the *VRML Review Board* (VRB) for approval. The VRB - the official incarnation of the VAG - is the "Supreme Court" of VRML, charged with maintaining the coherency and utility of VRML. The VRB in turn reports to the Consortium's Board of Directors - composed of member organizations - who oversee the overall direction of VRML, both technically, and in the marketplace.

If this all seems very much more formal and bureaucratic than the pirate ad-hocracy of VRML's earliest days, that's because VRML is now big business. Hundreds of millions of dollars in investment and product sales fundamentally rely on its success and stability. All of that money has engendered its own brand of conservatism - VRML style. The stable economic ecology created by the Consortium helps both the small companies and the big players, leveling the ground, and preventing any earthquakes. It's a community of companies, and, in that, not very different from the community of individuals who came together in mid-1994 to make VRML happen.

Enough history! This book is about the future - and how you can work to shape it. What you need to know from all of this is that VRML is *stable*; what you're learning here won't be obsolete by the time you finish reading it. VRML 2.0 - created by a world-wide community - has been built to last.