

# The Birth of MacroMind

Marc Canter

When we started MacroMind we had a vision of a world where PCs and other consumer electronic devices featured highly interactive computer graphic and music interfaces. This world was our version of the scene from *Blade Runner*, where the hero dives into the photo—navigating through the pixels at x1,000. We knew that the digital future would need tools and we were toolsmiths, experienced in producing development tools for videogames.

But when we started MacroMind, we didn't know anything about a GUI or mouse. We didn't know anything about the PC industry or selling software. All we knew was that the Macintosh had a whopping 128k of RAM and built-in audio and video cards—and that's all we needed to know.

We were videogame guys from Bally-Midway in Chicago. Our company was first called Chicago Software, until we decided three days later to name it after one of the characters in my co-founder Jay Fenton's game—GORF. We were working on its sequel Ms. GORF together when Jay and I, along with our other co-founder Mark Pierce, incorporated MacroMind in April 1984.

One thing we knew—that the world needed end-user “tools” that could be used by artists, musicians and designers—to create this “stuff”—this combination of graphics, text, music and interactivity that we knew was possible. Digital video was still a far away dream, but we knew what the potential of computer music and graphics was—as we all had been working with it—throughout the late 70's in Chicago.

But the only way to control computers to do musical or visual things back in 1983 was to program it with C, FORTH or some other programming tool.

So we set out to create videogame development tools that could be used by artists or musicians. We called ourselves a software rock & roll band and were represented by the Wm. Morris talent agency. We likened ourselves to modern day poets, rock stars or painters, crafting our tools like great gunsmiths or watchmakers before us.

This notion of an end-user tool that was the modern day violin/paintbrush is what was behind our original product SoundVision. But our publisher insisted that we separate the concept and create two products: MusicWorks for computer music (which became the MIDI and sampling software business) and VideoWorks for graphics and animation (which evolved into multimedia animation and authoring systems.)

MusicWorks was released in Oct. 1984 and featured the first music score, real-time timeline. It also had a piano keyboard timeline and an Overview window. As the music was playing, the end-user could plop down quarter notes or eight notes or draw on the piano scroll. This was radical stuff back in 1984 and drew the attention of Alan Kay, Steve Jobs and the Macintosh community. This was back before MIDI existed.

But within 12 months there were over 10 other Macintosh music products, and there were fewer than 100,000 Macs out there! Because such an inordinate number of programmers were also musicians, the Macintosh GUI was like a magnet for everyone who EVER dreamed of a real-time interactive music interface.

VideoWorks was released in June 1985—and that turned out to be our saving grace. By Xmas of '85 we had baked a tiny basic scripting language into the VideoWorks time-

line and licensed it to Apple, to produce guided tour training discs which would ship with every Macintosh. VideoWorks' timeline synchronized multiple channels of moveable sprites with an audio channel. Any sprite could be moved at any frame, and as thousands of frames zipped by, designers found themselves with a dynamic world that could combine and move design elements and navigation controls along motion paths around the screen. A scripting system was interlaced with this synchronization mechanism enabling creative people to decide where a hot-spot was, what action would happen upon clicking on that hot spot and to completely redefine what menus were.

Each multimedia menu took on a character of its own, and thus was born "multimedia authoring." Hypercard was around at the same time, but because it was based on a non-time-based card system, it lacked the ability to author the dynamic nature of multimedia (but DID have database capabilities!) By re-defining what menus were and creating entirely new worlds of education, entertainment and visualization, multimedia designers had in their hands a tremendous powerful new kind of tool which could take them into any realm. Empowering normal, non-technical people to author multimedia was what we stumbled onto, not by design or intent. We discovered multimedia authoring because that's what the market needed, what the world needed—at that point in time (mid-late 80's.) We just looked out around the landscape and saw what was possible, and sat down next to the people who were building these multimedia thingies—and we then gave them what they wanted and needed.

This was the secret to Director. We WERE multimedia developers. We kept our company going, doing gigs for Microsoft, Ashton-Tate, Lotus and Borland. We practically invented the concept of the floppy disc (later CD ROM) marketing and training disk. We did our first projects in 1985—and had to give up the practice in 1988 when the VCs entered into our lives and insisted that we not compete with our own customers.

So because we had a direct connection to the animators, designers and musicians who were using our tools, we knew exactly what features and capabilities they needed. We also put our programmers into our trade show booths and had them demo to potential customers. This made them understand how people were seeing and approaching their software. These were some of the reasons that Director evolved as it did.

But what was really going on was that we were participating in the first golden era of software development—which started with the release of Lotus 1-2-3. Pagemaker, Photoshop, Painter, Filemaker, Quicken, Sidekick, MORE, Quark Express, Word and Word Perfect. Productivity software took the power of microcomputers and put it into the hands of non-programmers. That was a big deal back in 1986!

Now in 2003 we look back and laugh. But we're not done yet.

## The New Paradigm of Tools

We've barely moved forward since those days of word processors and spreadsheets. Since 1988, the only major new software categories that have been created are email, IM, browsers and blogging tools. For all the flack, noise and supposed development, we have barely innovated the initial concept of what a tool is.

And we've lost track of what some of the potential of digital media is.

Doug Engelbart talks about an augmented computer experience which improves the way human beings interact and work together. Don Norman talks about an invisible computer and activity based approach to user experiences. Nicholas Negroponte has talked about bits and the digital convergence. Only 3% of the world's fiber infrastructure is currently being utilized.

I have (over the past 11 years since I left Macromedia) done applied research in scalable

content and developed various implementations of multimedia personalization. Whatever you call it, there is a new paradigm of tools coming which will jump forward as far from where we are today as did that jump from assembly language programming to productivity tools in the 1980s.

Tools today are still fundamentally based upon shrink wrapped product mentality. It goes on a truck to a warehouse, where it is distributed through a two-tier system to retail locations, where consumers come in cars and pay for the product—upfront. They then take it home, by themselves, read the manual, by themselves, and attempt to use the product. That product outputs traditional formats for traditional distribution systems—and you'll have to BUY any additional upgrades, improvements or retrofits to new technology. That all seems pretty archaic to me.

In 1997 Dave Winer had a content management system called Frontier which had a web site framework and series of utilities surrounding it. Dave found that he needed various communications and data flow control, so he started to develop a sophisticated set of capabilities and functions that worked with a browser, on-line servers and a simple, outline based scripting language. He invented a protocol mechanism for sending and discovering any sort of XML data (XML-RPC.) He was in a zone.

As I sat next to Dave and watched this object database shuffle, filter and sort data, I wondered at how direct, simple and easy it was to manipulate (what I thought were) complex data structures and procedures. To my delight I saw a simplistic system of customization and templating that could be easily adapted to many of the things I wanted to do!

At that point I knew that these huge web creation engines (like Vignette, Broadvision, Interwoven, ATG had) were in fact just grown up versions of Dave's Frontier system. But these huge systems were relegated to building traditional web sites and ecommerce (partially because of their cost and complexity), and were not pushing the envelope as to what was possible.

But Dave (and a company named Pyra) stumbled onto something that WOULD push the envelope—blogging. By 1999 Dave had a hosted blog service called EditThisPage just about at the same time that Pyra released Blogger. These early blogging tools were the first of what I call this new paradigm of tools, where everything the end-user needs is right there in front of them—in the browser page.

As the blogosphere evolved, Dave released Radio Userland, which put the blogging tool on the client's machine and integrated a new aggregator with the tool. Thus was born the next step of this paradigm, kind of picking up where Napster (and the other P2P clients) leave off.

News aggregators were made possible because Dave and Netscape had developed a syndication standard called RSS and all of a sudden any blogger could have their blog's RSS channel subscribed to—off a list right next to the NY Times or the BBC's channels. Other standards like OPML and the MetaWeblogAPI came out and this all fueled what we know as the blogosphere today. These standards combined with XML-RPC led to the rise of news aggregators and a world of 100,000's of RSS feeds and blogs.

This blogosphere has enabled new kinds of communities and interaction, all based upon the simple premise of personal publishing. As Anil Dash declared it, this was the era of "micro-content," where smart, distributed chunks of content would usher in an era of customizable, sustainable ecosystems. New kinds of micro-content "browsers" would come into being, enabling whole new kinds of navigation and browsing. I had often dreamed of going beyond just linking that T B-L gave us. I've often wanted a 2-way link, or a video or audio link or a link that expressed what sort of link it was. I discovered (through the blogosphere BTW) that the dream of semantic web was coming into reality.

The blogosphere is based upon the principle of self expression as a new kind of amateur journalism, whether you are reporting on the Battle for Baghdad or in your backyard. What I began to see was how the technology, spirit, ethics and process of the blogosphere can be applied beyond to much larger areas of influence.

The syndication and alternative distribution systems that drive the blogosphere will give birth to much more than just amateur journalism.

Jonathan Peterson in his Corante.com blog “The Me in Media” writes about new trends in personal publishing and how individuals one day will generate the majority of on-line media. The belief in ‘narrowcasting’ is something that has inspired me for years, and we have put a NARROWCASTING button into our new tool WebOutliner—even though we don’t really know what narrowcasting is and what it can do.

But we’re confident that our end-users will tell us. Just like we learned from the early days of VideoWorks, we’re planning on evolving with the market—just as soon as we figure out what it is! This process of iterative design was the secret behind Director and something that can’t be formerly engineered, scheduled or Gant charted.

## Open Standards

When I was a young entrepreneur I was introduced (by Dave Winer) to a book which became my marketing bible —Marketing Warfare—by Reiss & Trout. This book talked about erecting a new hill (which represented a new product category) and erecting barriers of entry on top of that hill, holding off your competition with machine guns and hand grenades. This is how we captured and held onto the nascent multimedia authoring tools business.

In fact that lock-in strategy of ours has done pretty well for Macromedia. 15 years later they have 97 percent of all the browsers running their Flash format.

But in 1994 a guy named Doc Searls said: *“It shouldn’t be warfare, it should be a conversation,”* and thus a new trend was born, which was personified in a book called the ClueTrain manifesto. In this book Doc and his fellow Cluetrainers tried to explain that if you don’t listen to and work with your customer, you’ll be gone.

And in fact they’re not your customers anymore. They’re your partners, because without them, you’re nothing. This new refreshing approach to how software is bought and sold has given birth to a new open world of standards and source code.

Open source and open standards have evolved over the past eight years, providing LAMP (Linux, Apache, MySQL, Python/Perl/php) and an NEA kind of environment for all to use: No one controls it. Everyone can use it. And Anyone can improve it.

The Internet has blossomed into a rich garden of innovation and from that garden is springing a new paradigm for tools. It’s a World of Ends—where the distributed power of people working together can conquer over the entrenched powers and infrastructure.

In fact it’s a new kind of infrastructure that we’re building.

Many people ask me how to make money in the open source world. I always point to the story of Jabber—which is both an .org and .com. We plan on offering our WebOutliner as “open source” and will ask only one thing from those who use the code: you have to link to us and give us credit.

We plan on making money selling our software based upon the principle of “pay for using.” The WebOutliner will allow you to use it for free—as long as you save your files onto our servers. But if you want to save your files onto your OWN servers or local drives, then we’ll ask for \$ 40. You won’t be ABLE to upload your media onto servers at all. You can dabble with the tool as much as you want, but the moment you start relying upon and caring about what you’ve created; then THAT’s when we think we can ask for money.

**Everything we need has been invented,  
now it's time to get it all to work together**

OK—so all these trends are lining up, we've got open standards and lots of open source code (and more coming—thanks to Mitch and Andy!)

One of the most exciting evolutions I see coming is how the technical and social standards established within the blogosphere will spread around the world. This process will happen if for no other reason than because ISPs need to sell bandwidth and there's almost an unlimited amount of it. So they use software to differentiate their offerings.

The same goes with consumer electronics vendors and system operators. In fact there's almost an endless amount of entities that can benefit from these new kinds of tools and the functionality and content they enable.

But (of course) the best stuff will come from the digital artists and musicians (and others) who just wanna play with it. There's so much to do! Contributative and open galleries, collaboratories, Wikis, forumne entirely news, social networks and systems will be created to enable new forms of creativity.

If you can imagine a new kind of interactive TV show which originated on someone's PC, and which narrowcast itself out to a limited number of people—well that's just ONE thing that's gonna happen!

Virtual garage sales will enable folks to sell off their stuff—and go around eBay. New kinds of tools will enable the creation of new kinds of micro-content, while also providing a built-in on-line community of fellow micro-contenters.

One of the ways that this could happen is what Seb Paquet calls Structured blogging. This idea of micro-content can be applied to lots of other kinds of media and data types. A person's persistent digital identity profile can be thought of as micro-content. A media library or a collection of photos (albums) or music or video (playlists) are also micro-content types. These are the obvious new kinds of micro-content.

But there are also some rather revolutionary new kinds of micro-content. A conversation (whether it be from an IM or chat session, email interchange or message board or forum) can be micro-content. A review of a movie, record or restaurant can also be micro-content. And an entire on-line community can be encapsulated as micro-content.

Tim O'Reilly talks about new kinds of distribution systems. What we have to imagine is what happens when new kinds of tools create lots of new kinds of micro-content which enable new kinds of on-line communities.

This is what our company—Broadband Mechanics—is doing, and we hope that as many new tools and technology as possible will enter into this ripe arena. The idea is that all of us together can equal what Microsoft is working on, which they call Longhorn.

It's possible that an inter-connecting world of micro-content servers and RSS aware tools can create a distributed, open source, web services based People's Mesh.

Longhorn and Apple's iLife will be the litmus we will compare our People's Mesh to. The goal would be to equal their functionality, but have it free and open for us all to use.