CAREER AND STYLE

William Bricken
January 1999

My current professional goal is to use the tools of Computer Science from a humanistic perspective -- to access powerful symbolic tools while maintaining control over those tools. My technical work has evolved through concept, implementation, and pragmatic commercialization.

Cross-disciplinary

My career has followed my heart, through *multidisciplinary studies* in Psychology, Education, Mathematics, and several branches of Computer Science (Logic Synthesis, Visual Languages, Computer Graphics, Artificial Intelligence, Intelligent Tutoring Systems, Human-Computer Interaction, Software Development, and Mathematical Foundations).

My Diploma of Education is in Mathematics and Counseling. The Stanford PhD combines Research Methodology, Computer Science, Artificial Intelligence and Educational Psychology; the dissertation addresses the modeling abstractions and pragmatics of computational tutoring systems. As a professor, I've worked in Departments of Social Psychology, Education, Industrial Engineering, Computer Science, and Software Engineering. My technical publications are in journals of Educational Technology, Computer Graphics, Artificial Intelligence, Visual Languages, Innovative Computing, and Human-Computer Interaction. My research addresses the reconstruction of the formal foundations of computation, based on spatial rather than textual representations of programs.

As well I've been straddling the academic/industrial gulf over the last fifteen years, working jointly in academic and in industrial research labs. Advanced Decision Systems conducted research for the Department of Defense; I worked on projects emphasizing distributed computing, maintenance training, program semantics, intelligent interface, cockpit design, and parallel logic processing. As Director of the Autodesk Research Lab, we explored innovative uses of CAD, including object-oriented interface, fractal curve generators, spatial algebras, and virtual reality (VR).

The University of Washington (UW) Human Interface Technology Lab (HITL), where I was the Principal Scientist, focused solely on immersive virtual reality. VR incorporates many hard problems in computer science: real-time distributed processing, multiple concurrent and interacting participants, fast sensor fusion, accurate multi-sensory display, computer graphics design and construction, and an exotic interface. The VEOS system that I designed and helped to implement was one of the most functional, state-of-the-art VR operating systems around in the early '90s. It incorporates interactivity tools which are based in physiological, psychological, and computational research. And it addresses many of the technical issues of web-based interactivity.

For grounding in physical reality, I've personally designed and built a large house, along the way mastering dozens of specific construction skills (such as landscape and architectural design, rough and finish carpentry, masonry, glazing, plumbing, electric, and lots of hauling and digging). The architectural design addressed functional, acoustic, air, heat, visual, and sound flow patterns. The garden was planted with native plants harvested from the nearby forest; the fireplace was faced with lava rocks collected from the local beach; the windows were installed after the walls so that they could be situated for specific views.

Educational Innovation

I carried innovative educational techniques into my classrooms at Seattle University (SU). Each class mixed lecture and participation. AS a teacher of skilled software professionals, I focused on facilitation of learning while insisting that all students take responsibility for their own learning. Each student choose between three grading options: standard, goal-oriented contract, or direct negotiation with the instructor.

My educational philosophy is constantly being refined. I believe in *embodiment*, that education must be placed in a physical and visceral context for somatic as well as conceptual learning. Learning is deeply connected to *empowerment*, to that which students find relevant and functional in their lives. Empowerment requires that the curriculum itself be jointly designed by student and faculty, that students have direct control over and responsibility for their educational experience. And I strongly believe in *exemplification*, that the teacher must demonstrate what is being taught, and, more importantly, that how something is taught must be consistent with what is taught. So I tend to avoid constructed toy problems in favor of current technical issues, as untidy and as difficult as they may be.

Learning is embodied in action; learners are empowered when responsible for their own learning; teaching should exemplify both.

Within the high tech industry, *metaskills* (learning how to learn, tolerance of ambiguity, handling constantly shifting content, just-in-time skill acquisition, finding rather than remembering information) are far more important than content skills. Teaching and learning is for the long-term, with performance goals located in the five year range rather than the five week range.

Recently, I have been working on an idea for addressing the general absence of *ethical grounding* in Computer Science by developing a *Computer Humanities* curriculum. The idea is to refocus CS on the person, to see computational hardware as a rather obstinate tool that is in its infancy, poorly developed, and horribly designed. Computation is quite foreign to cognition (I am adamantly not a Cognitive Scientist), and our software tool developers have forgotten that tools serve people (as opposed to the dominant perspective that a person is free to invest time in operating computers).

One new idea -- originated by Mitch Kapor -- is to view institutional software (large accounting and transaction systems, operating systems, shipping and tracking, etc.) as national infrastructure, similar to our roads, airports, telephone lines, and broadcasting frequencies. This does not mean that the government manufactures software, just that commercial software developers will be assured national usage, with fees to match, at the cost of open standards.

Fundamentally I believe that knowledge itself has been developed redundantly in multiple academic and professional communities. Educators must work toward simplifying and condensing the diversity of notations across disciplines. My junior year in college turned out to be influential: I was taking Differential Equations from the Math department, Inorganic Chemistry from Chemistry, and Mechanics from Physics. At one point, all three were teaching the same material (the chain rule for differential equations) using different vocabularies and different notations. I realized then that "multidisciplinary" simply meant having a unified perspective while avoiding the knowledge elitism which maintained artificial boundaries and hidden information. Today in CS, one of the deepest problems is that programmers are holding onto esoteric knowledge (such as C programming hacks/tricks) in order to preserve intellectual territory. The computer industry is as yet too immature to see the uniting threads that have the potential of making computation essentially simple.

Technical Innovation

I thrive in new and young research programs, having contributed to the founding of a couple of research labs (both still struggling), and a couple of entrepreneurial start-up companies (both failed).

My team fielded the first *publicly accessible immersive VR system* at SIGGRAPH'88. The intent was to make this technology freely available (modulo having to wait in line for an hour) to non-technical folks, to evaluate its effect on the naive user. The results of over 500 subjects yielded two central phenomena which almost everyone experienced: freedom and relevance. Folks delighted in the lack of constraints in a digital environment. It made them feel like superheros. And almost everyone said that the technology was relevant to their work and their lives. However, we also observed the very powerful suggestive capabilities of VR, it taps directly into an emotional commitment.

I moved back into academia to study the psychology and ethics of VR technology, helping to establish HITL at UW. The immediate task was to build the lab infrastructure. For this I negotiated with many high tech companies, securing $3M in donated equipment. I selected a small group of graduate students to admit into our program, and I provided classes and tutorials on VR software system design.

The technical challenge was to build a VR operating system that permits rapid configuration. VR raises so many significant research questions, and other systems at the time took literally days to reconfigure. The VEOS system which we built is reconfigurable in real-time. We could pose a question (What if your arms were made of rubber and could stretch to reach distant objects?) and have an experiential answer within minutes.

I was the Chief Scientist for a start-up company that invested nearly $1M to further develop interactive VR software. The culmination of that work was a state-of-the-art VR system that supports multiple participants and complete environmental interactivity. The company failed to raise second round funding, primarily because the market for VR equipment had not yet ripened. Of course, it is in full flourish today in the form of video gaming.

My technical fascination is with *experiential formal systems*, a project that I’ve been working on for over 20 years. Spatial math uses a system of formal representation and transformation that is visual and manipulable, the opposite of the symbolic abstraction that constitutes almost all current formal approaches. This work was initially quite difficult to support, since it challenges so many conventional assumptions. Basically, I was told that I was “plainly wrong”. So after failing to gain support for the ideas in lectures, I focused on implementation as proof of principle. That work found support from Paul Allen’s research group, Interval Research Corporation, where I developed algorithms under trade secrecy. Secrecy means that I have not published over the last few years, although I have generated one book and dozens of publishable internal technical reports.

In attempting to validate the desirability of the techniques of spatial mathematics, I sought the most difficult pragmatic problems in the application area of logical deduction. These hardest problems occur in the optimization and validation of large semiconductor circuits. I've succeeded in demonstrating world-record performance for benchmark circuit optimization using a diagrammatic formal system to achieve Boolean minimization. These tools use “void-based” reasoning, essentially using void-substitution (erasure) in place of symbol manipulation. The ultimate goal is *reconfigurable computing*, the union of hardware specification languages and software programming languages into physical circuitry that can be programmed and modified in real-time. This could provide hardware acceleration for any software algorithm using only one reconfigurable chip. This project is motivated by the performance demands of multimedia computing, which currently require specialized, and largely non-redundant, circuitry to accelerate functions such as audio, 3D graphics, videoconferencing, and compression.

In 2001 I co-founded Bricken Technologies Corporation (BTC) in order to commercial the boundary mathematics application to semiconductor design and optimization. BTC provided the opportunity to spend three years intensively programming. I loved the opportunity to develop novel, high-performance algorithms for exceptionally difficult optimization problems. The company was formed, unfortunately, just after the dot-com bubble burst, so attracting funding proved to be very difficult. We finally closed the corporate doors early in 2005. Building software in start-up companies is arduous, due to limited time resources. Errors are too costly to be acceptable, and I thrived in these difficult circumstances. I deeply enjoy the challenges of high performance programming because I enjoy designing architectures and algorithms that require extremely high quality. I've developed a suite of unique skills that apply widely, and I like to use them directly to solve business problems.

Weaknesses

I speak only English (unless you consider programming as a foreign language). Generally I work on long-term future technologies with a formal basis (this includes formal verification, new models of inference, virtual reality, reprogrammable hardware, and advanced educational technology). This disposition leaves me somewhat abstracted, a visionary rather than a pragmatist. I also work with technically unpopular concepts, preferring radical change to gradual improvement.

That said, I have an excellent reputation managing technical teams, and have been fortunate enough to lead teams that have developed nationally acknowledged software capabilities.

Potential Contributions

Computer Design:

- -- advanced VR systems
- -- software tools for the design of interactive experience
- -- logic engines (serial and parallel)
- -- logic synthesis
- -- intelligent tutoring systems
- -- computational pedagogy and ethics
- -- complementary skills in both technology and education
- -- deep knowledge of current technology and of technological trends

Fluent in LISP; competent in Mathematica, Prolog, and Pascal; reading familiarity with most other computing languages, including hardware design languages such as Verilog and VHDL. I've worked with most computer graphics development systems circa 1995 (Wavefront, Alias, Swivel, Director, AutoCAD), and all common operating systems. Deep knowledge of the design and architecture of most AI-based tutoring systems (Mycin, Emycin, Pixie, Teiresias, Scholar, Sophie, West, Guidon, Buggy). Substantive knowledge of logic synthesis algorithms and techniques.

I produce around 50 pages of error-free code per year, since 1982.

Internet Information and Knowledge Systems

-- GUI design and interface programming
-- multi-sensory display and programming
-- design of interactivity
-- artificial intelligence
-- agent technologies
-- knowledge organization and representation
-- theory of ontology and knowledge modeling
-- pedagogical strategies
-- inferential knowledge-bases and computation

My approach to the design of web-based experiences is first from effective educational tools and technologies, emphasizing accessibility, ease of use, and maintainability. In internet technologies, content is king. The technology itself should be invisible, requiring a minimum of cognitive load for both learning and usage. Next most important would be CS facilitation and intelligence augmentation, how to help a person to solve problems and gather information more effectively. I believe that an intimacy with technology and technological change is mandatory, however, third, I'd emphasize defensive postures (technical and psychological) for controlling rampant technological change.

My approach to the technical infrastructure of the internet is formalist. I'd use elementary mathematical modeling to attempt to control the chaotic structure of web information, and code-generating programming techniques in order to buffer the human user from configuration complexities. Lately, I've been thinking about how to construct a semantically meaningful web-based interaction, and how to contribute to the development of a semantic web facility.