Bruce L. Horn, Ph.D.

2440 Western Avenue #505
Seattle, WA 98121

(650) 207-2401 (cell)
blh@acm.org

Education

1985-1993	Ph.D.	Computer Science, Carnegie-Mellon University Dr. James Morris and Dr. Jeannette Wing, advisors.
1991	M.S.	Computer Science, Carnegie-Mellon University
1988-1990		Research Assistant, University of Oslo, Norway Worked with Dr. Kristen Nygaard, co-inventor of Simula and the field of Object-Oriented Programming.
1977-1982	B.S.	Mathematical Sciences, Stanford University

Professional Experience

1/2012-present *Member of the Technical Staff*
Wavii, Inc.

I am currently working on part of the natural language pipeline that involves time-tagging of events specifically and temporal relevance more generally.

4/2011-12/2011 *Principal Research SDE, Natural Language Feature Team*
Microsoft, Inc.

I was responsible for building libraries for our Finite State Transducer optimizing compiler and runtime engine for use within the Bing online and offline infrastructure.

6/2010-4/2011 *Principal Research SDE, Natural Language Engineering*
Powerset, Inc., a Microsoft Company

In this role I worked with the researchers and engineers in the NLE group. Specifically, I was responsible for two projects: the first (primarily by myself) developing new technology for optimization of tail queries using extremely high-quality lexical and semantic resources, and the second (with another researcher) building a fast, flexible, high-quality, semantically-driven text summarization system.

1/2010-5/2010 *Principal Research SDE, Semantic Initiatives*
Powerset, Inc., a Microsoft Company

I worked with a small group of researchers, led by our Chief Scientist, to identify and specify unique applications of our natural language technologies in the development of the Microsoft Bing search engine. We were quite successful in this endeavor and were able to kick off several new efforts within Powerset/Bing, as well as consult on and influence existing projects at Microsoft.

5/2009-12/2009 *Principal Development Manager, Reference and Answers*
Powerset, Inc., a Microsoft Company

I managed the Reference and Answers group, a group that consisted of both the backend (crawling, indexing, and serving up documents from natural-language queries) and the frontend (web interface, runtime, and user experience) of both Powerset.com and bing.com/reference, an application component of Microsoft's web search service.

Our group was responsible for the entire pipeline for web search: importing content, indexing it on our servers using our proprietary hybrid NL/keyword approach, handling queries, and serving up the results pages and answers for the reference application. We provided the user experience in bing.com/reference/ and displayed Wikipedia results in our own republished format.

We worked closely with the NL Engineering group, which I previously managed. The NLE group was responsible for the fundamental natural language processing of both the content and user queries.

5/2007-5/2009 *Director, Natural Language Technologies*
Powerset, Inc., a Microsoft Company
San Francisco, CA

I managed a growing group of computational linguists, semanticists, and tool developers. Our group's focus was the natural language interpretation and encoding of documents and end-user queries for the Powerset semantic search system.

The NLT Group was responsible for the entire Natural Language pipeline, from tokenization and sentence breaking, morphological analysis, syntactic analysis, through semantic transforms and finally to the creation of linguistic facts that were then stored in our semantic index. We were also involved in the maintenance of critical tools including finite-state transducer compilers and the ambiguity-preserving syntax engine. The semantic platform was the keystone of the company’s technology.

Early in my tenure at Powerset I took over and merged two groups to form the Semantic Platform group, which built tools to support NL development and testing.

I managed or participated in a number of key projects, including:

- the porting of our Prolog transfer system to C++;

- lexicon server and client optimization;

- semantic transfer system refactoring and extensions;

- and development of a flexible cloud-computing infrastructure.

In addition, my direct technical duties involved:

- Designing and developing initial versions of our end-to-end web-based NL pipeline testing and regression tool;

- Maintaining and upgrading code for our Ruby-to-NL tool interfaces;

- Writing Ruby packages for translating from Prolog, examining, displaying and extracting semantic facts from our transfer system;

- and moving most of our software packages into our continuous build and integration system.

1/2005-7/2006 *System Architecture Consultant, WildPackets, Inc.*
Walnut Creek, CA

I worked with the WildPackets development team in the design of new versions of their flagship product, OmniPeek. My focus was on developing extensions to the OmniPeek feature set to include powerful forensics tools; network baselining using advanced statistical methods; and anomaly detection.

My deliverables at WildPackets included:

- The design of a improved architectural framework for a distributed server-based packet capture and analysis system;

- Evaluation of third-party network analysis systems and technologies;

- Strategies for post-capture analysis;

- and a new architecture, called Events, Activities, and Monitors, for network baselining and anomaly detection.

1999-2004 *Cofounder and Chief Technical Officer, Marketocracy, Inc.*
Los Altos, CA

I co-founded Marketocracy for the purpose of finding the world’s best investors to manage a family of mutual funds. I created the first version of, and led the continuing design and implementation of a Web-based software framework for allowing individual investors from all over the world to manage realistic model stock portfolios. The model portfolio managers with the best track records were then chosen to run Marketocracy’s real funds.

Marketocracy’s web-based simulation provided leading-edge tools for mutual fund management, analysis, real-time trading, and forums for our worldwide community. The site was written using the WebObjects framework, in Objective-C, using a distributed Frontbase database system.

At Marketocracy, I:

- Designed and implemented the prototype system which was instrumental in conveying the viability of the model-portfolio concept to Marketocracy’s initial investors.

- Presented business and technical plans to capital partners.

- Hired critical employees, including the Chief Information Officer and key engineering staff.

- Conceived of, designed, and led the implementation of the real-time web-based stock trading system and simulation that is the fundamental basis for the company;

- Contributed significantly to the specification and writing of most of the patents in Marketocracy’s patent portfolio;

- And maintained team morale by facilitating communications between senior management and other employees in the company.

I left day-to-day operations in spring of 2002 after completing the site design and backend trading system. I served six years on the Marketocracy Board of Directors, leaving in December 2005.

1997-present *Founder and President, Ingenuity Software, Inc.*
Mammoth Lakes, CA

I founded Ingenuity Software to develop new and innovative information management systems for personal computers. The first such system, iFile, provided a traditional Finder-style desktop system combined with the ability to apply and search on extended attributes to each piece of information in the system, and to provide automatic and manual categorization facilities, iFile also extended the desktop metaphor to other kinds of objects that traditionally are managed with special-purpose applications, such as email programs and address books. By unifying all kinds of information onto the desktop, iFile made it convenient to link, search, and cross-index contacts, email, and documents in an efficiently integrated system.

The iFile system went through beta-test on early Macintosh OSX systems, with ports to the latest Mac OS, Windows and Linux planned. The underlying technology can be used in a wide variety of applications, and is available for acquisition or licensing. Patent number 7,275,063 describes some of the iFile technology.

1984-1997 *Computer Science Consultant*
Palo Alto, CA; Pittsburgh, PA; Oslo, Norway

• I designed and implemented a prototype Postscript LaserWriter spooler for Adobe Systems, Inc., which was subsequently used by Apple for their LaserShare product.

• I designed and implemented a multimedia client application using Telenor's h.263 video codecs for Eloquent, Inc.

• I consulted at Apple on novel research topics in their Advanced Technology Group, and worked on LiveDoc, an experimental system that automatically structured documents.

• I provided expertise in system design, user interfaces, and graphics algorithms to Maya Design Group. Helped with the design and implementation of their experimental Hyperfax system.

• I provided consulting services to companies and industrial research laboratories in Norway and the United states on programming language issues and reactive interfaces.

1981-1984 *Apple Computer, Inc.*
Cupertino, CA

At Apple I was one of the principal designers of the original Macintosh computer. I conceived of, designed, and implemented the Finder, the Macintosh's visual mouse and window-based operating system shell, which was used for the design for the Lisa Filer. Features first seen in the Macintosh Finder—icon-based, drag-and-drop file handling, direct filename editing, and double-clicking to open documents in their associated applications—have since become an industry design standard.

I was also responsible for the creation of several parts of the ROM-based system. One of the most important was the Resource Manager, a virtual memory/object-oriented database package, which was one of the two cornerstones of the system software, and whose concepts are also widely copied in the industry.

I created the Dialog Manager, a Window Manager extension that defined mouse-sensitive fields and provided hooks for automatic updating of various text, graphic, and control fields within a window. The Resource Manager and Dialog Manager were used to separate the language-specific parts of applications from the language-independent parts, thus easing the task of building software localized to particular languages.

Finally, I began the design of an object-oriented operating system based on ideas from Smalltalk but optimized for a platform that would support traditional programming languages as well as an intrinsic object-oriented system language. These ideas have since appeared in a variety of new operating system designs from Apple and other companies.

1980-1981 *Central Institute for Industrial Research (SI)*
Oslo, Norway

At SI I implemented the Smalltalk-78 programming language and environment on a Norwegian microcomputer, the Mycron-2000, under an agreement with Xerox PARC. I designed and implemented a screen-oriented debugger for the 8086 microprocessor, rewrote an assembler to assemble 8086 code, and wrote many diagnostics and utility programs to help in the implementation and the debugging of the 8086 Smalltalk interpreter, primitives, and memory manager. This system was adapted and extended for use in a variety of industrial applications by startup companies based in Norway.

1974-1981 *Xerox Palo Alto Research Center*
Palo Alto, CA

At PARC I worked in the Software Concepts Group (formerly the Learning Research Group), where Smalltalk and the modern graphical user interface was created, and was involved in various aspects of the system over the years. I implemented a music note-capture system and a digital terrain map projection system in Smalltalk, and was involved in several group projects.

I helped develop a portable 8086-based multiprocessor Smalltalk machine, called the NoteTaker. For that project, I designed and implemented much of the ROM firmware (display, keyboard, mouse routines, A/D and D/A conversion routines, touch-screen routines, multiprocessing support and interrupt handling) and assisted with the debugging of the prototype hardware.

I was primarily responsible for the implementation of the Smalltalk-76 kernel in microcode on the Xerox Dorado, an early bitmap-graphics-based personal research computer developed at PARC. This system was the one demonstrated to Steve Jobs when he made the famous trip to PARC to see Smalltalk.

Personal Overview

California native. Hobbies include cross-country and nordic downhill skiing, mountaineering, tennis, swimming, bicycling, flying, studying cognitive processes and the structures of subjective experience, and environmental and system studies. Competent Norwegian speaker and can understand Swedish and Danish.

Selected Publications and Patent Applications

Patent continuation of 7,275,063, “StickyPath.” Patent number 7,840,619 issued November 23, 2010.

“Internet-based system for identification, measurement and ranking of investment portfolio management, and operation of a fund supermarket, including "best investor" managed funds.” Patent number 7,509,274 issued March 24, 2009.

"Computer system for automatic organization, indexing, and viewing of information from multiple sources." Patent number 7,275,063 issued September 25, 2007.

"System and method for creating and maintaining investment portfolios." Patent application number 20,030,093,353, May 15, 2003.

Newcomer, Joseph M; and Horn, Bruce. "Inside Windows Regions." Dr. Dobbs Journal, Volume 18, Issue 3. March, 1993.

Myers, Brad A.; Smith, David C; and Horn, Bruce. "Report of the 'End User Programming' working group" In "Languages for Developing User Interfaces", Brad A. Myers, ed. June, 1992.

Horn, Bruce. "Properties of User Interface Systems and the Siri Programming Language". In "Languages for Developing User Interfaces", Brad A. Myers, ed. June, 1992.

Horn, Bruce. "Constraint Patterns as a Basis for Object Oriented Programming." ACM Sigplan Notices, Volume 27, Issue 10. October, 1992.

Assorted stories in http://www.folklore.org, Andy Hertzfeld's website from which his book, "Revolution in the Valley" was created.