The Honorable Thomas S. Zilly

UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF WASHINGTON
AT SEATTLE

TIM and PENNY PATERSON, husband and wife and the marital community thereof, Plaintiffs, v. LITTLE, BROWN AND COMPANY, a Massachusetts state corporation, TIME WARNER BOOK GROUP, a Delaware state corporation, HAROLD EVANS ASSOCIATES LLC, a New York state limited liability company, HAROLD EVANS, and DAVID LEFER, Defendants.	)	No. 05-CV-01719-TSZ EXPERT WITNESS REPORT OF GARY J. NUTT

I. INTRODUCTION AND QUALIFICATIONS

1. I have been retained by Davis Wright Tremaine LLP to prepare a preliminary report of my opinions to be expressed at trial, along with the bases for these opinions as required by Federal Rule of Civil Procedure 26(a)(2)(B).

2. I have been asked to offer my opinion regarding declarations from Mr. Tim Paterson and Professor Lee Hollaar regarding the *Tim and Penny Paterson vs. Little,*

Brown and Company, et al. case. To accomplish this, I examined the materials listed in Exhibit A to this expert witness report.

3. I have been a Professor in the Department of Computer Science at the University of Colorado at Boulder since 1986. Between 1981 and 1986 I worked in industry as a computer engineering product developer, manager, and executive. I was a researcher at the Xerox Palo Alto Research Center and Bell Laboratories from 1978-1981. I was an Assistant Professor of Computer Science at the University of Colorado from 1972 to 1978. Since 1970 I have studied many different aspects of computers including computer hardware, operating systems, networks, distributed systems, application software, and distributed software. I have been a legal consultant since 1995, providing consultation on cases related to computer forensics, trade secrets, and intellectual property. As a legal consultant I have acted in the capacity of consulting expert, testifying expert, court-appointed expert, and special master. Exhibit B is a copy of my *Curriculum Vitae*.

II. OPINIONS AND BASES THEREFORE

A. Expert Report Summary

4. Dr. Gary Kildall created the first successful Operating System (“OS”) – Control Program/Monitor (“CPM”) – for a microprocessor.

5. CP/M defined a market that stimulated independent software development, inexpensive development platforms, and the proliferation of inexpensive application programs for inexpensive computers.

6. Dr. Kildall developed the Basic Input/Output System (“BIOS”) approach to designing small computer operating systems. This design was used in 86-DOS and its descendants.

7. At the time that Mr. Paterson developed the initial version of 86-DOS, he had no credentials for designing an OS, and used Dr. Kildall's CP/M design to direct the implementation of 86-DOS.

8. Computer scientists agree that the system call interface implicitly reflects aspects of the internal design of the OS. Thus, adopting a proprietary OS interface implicitly means adopting aspects of the OS internal design.

9. 86-DOS used, and depended on, the CP/M OS programming interface.

10. By replicating the implementation of the visible CP/M interface, one necessarily must replicate the semantic actions associated with functions on the interface. Thus, 86-DOS necessarily replicated significant aspects of the CP/M design.

11. Sir Harold Evans' use of the term "clone" was qualified and accurately describes how 86-DOS compared to CP/M.

12. Mr. Paterson is unable to prove that 86-DOS did not copy algorithms, data structures, and other trade secrets and confidential information incorporated in the CP/M program.

B. Background

13. In 1980, most of the value of Digital Research, Inc., ("DRI") was reflected in its products, the most commercially successful of which was CP/M. Dr. Kildall and his company, DRI, had made a major investment in (1) designing and developing the CP/M operating system for the 8080 (and Z80), and in (2) establishing a significant market for the product. People who wanted to write application programs using the 8080 or Z80 required an OS to manage the execution of the program, and to provide certain services such as file management. Like other operating systems, CP/M exported a "programming

interface" (or "system call interface") consisting of a number of system functions that could be called from the application programs. Independently, the OS provided a "command interface" by which a human user could transmit commands and directives to the OS to control its behavior, e.g., to choose a program to execute, or to copy a file.

14. The effort to design and implement an OS is substantial. Dr. Kildall worked on the design and implementation of CP/M for two to three years before deploying the product.

15. The success of the product depended on attracting programmers to write application programs that would rely on the system functions provided by CP/M, thereby enticing customers to buy application programs, such as games or word processors, which depended on CP/M, and which, in turn, would be delivered with popular microcomputers.

16. Because programmers could not write programs that depended on CP/M unless they were provided with documentation describing how to use the system functions, DRI published a copyrighted description of the system function syntax and semantics. In doing so, DRI was providing programmer documentation to enable the market for its OS – which by then was a major commercial product – not inviting other parties to implement their own version of the OS using DRI's intellectual property.

17. Mr. Paterson's and SCP's sole reason for adopting the CP/M interface was to enable programs that depended on the CP/M interface to switch to microcomputers that used the 86-DOS OS, implemented on 8086-based microcomputers.

C. Microprocessor Technology Circa 1975

18. Intel introduced the 4-bit 4004 4-bit microprocessor chip in 1971 and the 8-bit 8008 microprocessor in 1972.[1] These microprocessors targeted embedded systems applications (in which the microprocessor typically replaced subassemblies implemented with electronic logic components). These microprocessors were not capable of supporting their own development environments, so it was necessary to implement their programs on a separate, larger computer. Intel provided specialized integrated development environment (IDE) machines on which software for their microprocessors could be developed. The Intellec 4 IDE was used to develop software for the 4004, and the Intellec 8 supported programming for the 8008.

19. In April 1974, Intel introduced the 8-bit 8080[2] microprocessor ("the 8080"). Intel upgraded the Intellec 8 to the Model 80 to provide an IDE for developing software for the 8080.

20. By the end of 1974, the 8080 began to appear in personal computers aimed at the hobbyist market: "In January 1975, *Popular Electronics* magazine's cover featured a picture of the Altair 8800 computer – the world's first microcomputer which used the new Intel 8080 processor …").[3]

21. Programmers quickly realized that the 8080 was sufficiently powerful to execute a much broader class of programs than the earlier microprocessors, e.g., word

[1] "Microprocessor Quick Reference Guide," Intel press materials, *available at* http://www.intel.com/pressroom/kits/quickreffam.htm

[2] *Id.*

[3] *History of the Microcomputer Revolution, Segment 5 – The World's First Commercially Available PC* (KPBX FM 91.1 radio broadcast 1995).

processors and electronic game programs. Hobbyist groups rapidly emerged to exploit the 8080-based microcomputers.

22. The cost of the Intellec 8 Model 80 was far too high for the average hobbyist programmer, however. Instead, many worked at a company, or attended college, where they were permitted to use large, corporate computers to develop their own programs. These programmers began to develop "cross compiler environments" in which one could prepare application programs on the large computer which could be executed on the 8080. For example, Mr. Bill Gates and Mr. Paul Allen are alleged to have used a PDP-10 computer at Harvard to develop a Basic Program translator for the Altair 8800.[4]

23. By the beginning of 1975, when the Altair 8800 was introduced, there was no operating system of any kind for microprocessors. While there were large machines (such as IBM and Digital Equipment mainframe computers) with sophisticated operating systems, no one had adapted OS technology to microprocessors.

D. Dr. Kildall Creates CP/M – The First Successful Operating System For Microprocessors

24. In 1973, Dr. Kildall began to develop CP/M, an operating system for microcomputers. He founded DRI in 1974 to market and develop CP/M and related products.

25. Dr. Kildall was among the first – if not the first – to build a successful operating system for a microcomputer. Between 1975 and 1979, CP/M was the most widely used OS for 8-bit microcomputers.

[4] *History of the Microcomputer Revolution, Segment 6 – What Good is a Computer Without Software?* (KPBX FM 91.1 radio broadcast 1995).

26. After Intel introduced the 8080 microprocessor, CP/M soon appeared in personal computers such as the Altair 8800.

27. Prior to the introduction of CP/M, microprocessors were not suitable for use as a general-purpose computer on which one could develop software. Instead, a programmer would have to purchase an expensive IDE to develop programs for a microcomputer, or use a cross development environment on a large computer. CP/M enabled a programmer to purchase a microcomputer with CP/M, and to use that microcomputer to develop its own software, all for an investment of less than $100 in addition to the cost of the microcomputer (compared to the cost of over $10,000 for an IDE; or hundreds of thousands of dollars for a large, general-purpose computer such as the PDP-10 or an IBM mainframe).

28. Dr. Kildall's development of an OS for the 8080 made it possible to create software, save it as files, manage the files, print information, input information, etc., all on an 8080-based microcomputer.

29. In developing his OS, Dr. Kildall used an Intellec 8 Model 80 that was not configured with a mass storage device (such as a floppy disk) – Intel may not have offered that option in 1975. Dr. Kildall acquired an 8" Shugart floppy disk drive, and had a friend (Dr. John Torode) develop a controller to adapt the floppy disk drive to the Intellec 8 Model 80. This enabled the Intellec 8 Model 80 to have sufficient mass storage capacity to save and retrieve files. It also represented a prototype of an inexpensive 8080-based microcomputer. While Dr. Torode developed the hardware controller, Dr. Kildall developed CP/M using an 8080 simulator on a mainframe computer (a Digital Equipment PDP-10 system). Once the controller was completed, Dr. Kildall installed the CP/M OS on

the Intellec 8 Model 80. By June 1975, there was an operational version of CP/M for the enhanced Intellec 8 Model 80. By 1976, DRI was selling Version 1.3 of CP/M, and publishing various forms of user documentation for the system, e.g., the OS general description for Version 1.3.[5]

30. The idea that one could simply borrow the OS technology from large computers such as the PDP-10 and apply it to an 8-bit microprocessor, was not technically feasible at the time CP/M was created. Large machines had extensive mass storage devices (hard disks), 32-bit or larger memory words and CPU functional units, very large executable memories (compared to microcomputer memory sizes), CPU operating modes, network devices, and proprietary system software costing millions of dollars to develop.

31. The fact that Dr. Kildall created an operational OS on an 8080-based microcomputer was a significant milestone in computer technology. The additional fact that it enabled the creation of a new occupation of independent programmer, and established an entire market in which one could use an 8-bit microcomputer as a general purpose, programmable computer is a remarkable scientific and commercial achievement.

32. Professor Hollaar has failed to recognize the significance and importance of this contribution. Incredibly, he even argues that Dr. Kildall's work is no more significant than reusing technology that is used on mainframe computers.[6,7,8]

E. SCP Enters Microcomputer OS Market Using CP/M Design and Interface to Develop 86-DOS

[5] *An Introduction to CP/M Features and Facilities*, Digital Research, 1976.

[6] Declaration of Professor Lee A. Hollaar ("Hollaar Decl.") in support of Plaintiffs' Response to Defendants' Motion for Summary Judgment ¶ 14, at 4.

[7] *Id*. at ¶ 16.

[8] *Id*. at ¶ 44, at 10.

33. By the time the Intel 8086 was introduced in June 1978,[9] microcomputers with 8-bit microprocessors were well-established as an inexpensive program development environment, and as a platform on which a broad spectrum of application programs could operate in conjunction with CP/M.

34. Prior to 1978, Seattle Computer Products ("SCP") produced various hardware components for computers. In 1978, SCP developed and began marketing a single-board computer based on the Intel 8086 microprocessor.

35. To help establish itself in the marketplace, SCP elected to bundle its hardware with an OS that programmers and users could use interchangeably with CP/M:

> "…[A]n operating system becomes useful when other people write applications for it.
>
> So in order to make it as easy as possible for other people to write applications, since Seattle Computer had no leverage or market share that would interest people, I chose to develop 86-DOS with an interface that was translation compatible with the CP/M application program interface."[10]

36. Professor Hollaar agrees that SCP's only option was to copy the CP/M system call syntax and semantics: "Paterson found it necessary to use aspects of CP/M so that translation of CP/M application programs to QDOS was possible. If he had changed the numbers for the operating system function calls, or the format of the file control block, or the ending delimiter for a string to be written to the console, it would have made it exceedingly difficult for developers of CP/M applications to move them to QDOS."[11]

[9] "Microprocessor Quick Reference Guide," *supra* note 1.

[10] Deposition of Tim Paterson ("Paterson Dep.") at 47, attached to Plaintiff's Response to Defendants' Motion for Summary Judgment.

[11] Hollaar Decl. ¶ 52, at 12.

Thus the admitted SCP market strategy was to use the CP/M design and interface to entice customers to buy SCP 8086 boards with 86-DOS.

F. Mr. Paterson Had No Credentials for Developing a Commercial OS

37. Mr. Paterson's first professional computer experience was as a part-time computer repair technician at The Retail Computer Store (while he attended undergraduate school). A computer repair technician is typically responsible for diagnosing and correcting errors in computer hardware, and sometimes computer software. Mr. Paterson took this position in about November 1976 for about a year and a half, where he remained until he graduated from college in June 1978.[12]

38. After earning his B.S. degree he went to work for SCP as a hardware design engineer.[13] During 1978-79, Mr. Paterson participated in the creation of the SCP 8086-based single board microcomputer that began shipping in November 1979. [14] Designing and prototyping a single board microcomputer is a hardware design activity that normally involves few software tasks.

39. Mr. Paterson recalls that his only experience in writing OS code prior to developing 86-DOS was in a college class project in either his senior year (1977-78) or in graduate school (1978-79). He states:

> "The description of the operating system was actually provided by the instructor; that is there was a list of possible…term projects that we could do and the projects were fairly specific and one of them was to write an operating system and operating system was to have the functions with what we called the P, the function V, start process, I guess an end process or something like that and I/O…but that's what I wrote an operating system with those,

[12] Paterson Dep. at 17.

[13] *Id.* at 20.

[14] Tim Paterson, THE RIGHT PLACE … THE RIGHT TIME 43 (1995) (unpublished).

> I believe, five specific functions and then additional work to sort of test and improve it."[15]

40. By Mr. Paterson's description, the OS functional interface for the assignment consisted of only five functions and they were part of the assignment specification. Mr. Paterson had no part in designing the system interface or OS requirements.

41. Sometime during summer 1979, Mr. Paterson discussed with DRI the possibility of obtaining a license to a 16-bit version of CP/M ("CP/M-86").[16] Mr. Paterson alleges that someone at DRI told him that CP/M-86 was scheduled to be released by the end of 1980.

42. Mr. Paterson states that by February 1980, SCP decided to create its own OS that would mimic the CP/M:

> "… I sought to make the application program interface (…) compatible with CP/M to enable automated translation of 8-bit programs into 16-bit programs. …, since no one had yet developed an operating systems on 16-bit chips, there was no point of reference but CP/M."[17]

43. At that time Mr. Paterson still had no prior experience in OS design, and no professional programming experience of any type. His only related experience, then, was as a part-time computer repair technician, a computer hobbyist, and as a college student.

44. Mr. Paterson testifies that he created 86-DOS during April-July 1980, using only copyrighted CP/M documentation and his experience as a user of a Cromemco microcomputer that used the CDOS OS. Mr. Paterson describes CDOS as "… a CPM

[15] Paterson Dep. at 35-36.

[16] Paterson, *supra* note 14, at 44.

[17] Declaration of Tim Paterson ("Paterson Decl.") in support of Plaintiffs' Response to Defendants' Motion for Summary Judgment ¶ 5, at 2-3.

look-alike and so CPM programs ran with C-DOS [sic]," adding: "we did not use CPM provisional research. We bought this look-alike from Crememco [sic] or it came with the computer one or the other."[18]

G. Dr. Kildall Developed CP/M BIOS Design, and SCP Used the Design in 86-DOS and its Descendants

45. CP/M was originally designed on the Intellec 8 Model 80, which was configured for its specific set of devices (including a custom floppy disk). Because other computers using an 8080-compatible microprocessor would use different devices – e.g., a different controller for the floppy disk – the parts of the CP/M that handled each new type of device had to be modified before CP/M could be used with these computers. Other parts of the OS, however – e.g., the file manager – did not require modification.

46. In CP/M Version 1.3 (circa 1975-76) Dr. Kildall recognized that he could design CP/M to *separate the part of the OS that managed the devices into a module that was distinct from the remainder of the OS*. This was the Basic Input/Output System (BIOS) that had a fixed programming interface to software functions with different implementations to manage different types of keyboards, paper tape punches, screens, floppy disks, etc.[19] This design enabled Dr. Kildall to be able to quickly adapt CP/M to a new 8080-compatible computer by altering only the BIOS portion of CP/M without having to alter the remainder of the OS. This was an innovation unique to CP/M, one which still exists in contemporary computers using Intel microprocessors. There is no doubt that 86-DOS reused this idea, and that it remained in MS-DOS and subsequent Microsoft Windows operating systems.

[18] Paterson Dep. at 37.

[19] Digital Research, *supra* note 5, at 1.

H. 86-DOS Used – and Depended on – the CP/M OS System Call Interface

(a) The CP/M System Call Mechanism

47. An OS call mechanism differs from a conventional function call in that it does not require that the calling program be *statically linked* to the OS code. Instead system calls use a *dynamic linking* technique that enables application programs to be prepared for execution explicitly linking the OS functions to the program until run time. Briefly, dynamic linking uses a form of indirect function reference, e.g., by creating a table with addresses of each of the system functions. Application programs invoke system functions using the table index rather than the actual function address (as is the case in static linking).

48. Both the 8080 CALL and the 8086 INT instructions are designed to invoke functions using these dynamic, indirect function references. The 8086 INT differs from the 8080 CALL in that it supports a larger table of address functions than is used in the 8080 (which supports only 8 indirect references to system functions).[20]

49. Professor Hollaar argues in his declaration that Dr. Kildall simply copied system call instructions from large computers. He appears to have confused the trap instruction used on IBM mainframes and DEC PDP-10 minicomputers with the 8086 "INT 21" instruction.[21] In the case of larger machine OS call mechanisms, the IBM mainframes and the PDP-10 incorporate processors with supervisory and user modes of execution, whereas the 8080 and 8086 execute in only a single mode. The SVC and UUO

[20] Professor Hollaar notes that Mr. Evans imprecisely refers to 8080 CALL as INT. This is immaterial, and in any event, Mr. Evans' description is otherwise accurate. By using the CALL instruction as an OS system call, the result was the same as if the 8086 INT instruction were used to simulate a CALL, albeit with a larger set of target function addresses.

[21] Hollaar Decl. ¶ 14, at 4.

instructions on the respective large computers are user mode instructions that switch the processor to supervisory mode and then branch to an OS function that will execute with the processor in supervisory mode. Comparing these trap instructions with the 8080 CALL instruction or the 8086 INT instruction misses the essential point of dual mode operation. In his declaration, Professor Hollar may have been thinking of the INT instruction used in the 80386, which behaves more like the SVC and UUO instructions.[22]

(b) Use of the CP/M Operating System Call Interface in 86-DOS

50. It is an undisputed fact that Mr. Paterson used the CP/M system call interface in implementing 86-DOS. As cited earlier in this declaration, Mr. Paterson states:

> "… I sought to make the application program interface (…) compatible with CP/M to enable automated translation of 8-bit programs into 16-bit programs. …, since no one had yet developed an operating systems on 16-bit chips, there was no point of reference but CP/M."[23]

51. Mr. Paterson is clear that, as a "tiny company," SCP believed that it would be doomed to failure if it attempted to create an OS and to establish it as a predominant OS choice. [24] Mr. Paterson knew that to develop a new OS from scratch would require many staff-years and a level of expertise that neither he nor SCP possessed; as well, he knew that the success of the company would depend on its ability to capture a significant share in a highly competitive marketplace. Mr. Paterson stated: "I felt that CPM [sic] translation would significantly promote the adoption of DOS by others in the computer industry, so I

[22] Hans-Peter Mesmer, THE INDISPENSABLE PC HARDWARE BOOK: YOUR HARDWARE QUESTIONS ANSWERED 81 (2d ed. 1995).

[23] Paterson Decl. ¶ 5, at 2-3.

[24] Paterson Dep. at 47-48.

made it a primary design requirement."[25] Professor Hollaar confirms this position.[26] So instead of creating an entirely new OS, then, Mr. Paterson chose to re-implement the functions that were carefully defined as the programming interface to the CP/M OS.

I. Paterson's Use of Translation Compatibility Illustrates 86-DOS's reliance on CP/M's Design

52. Within the context of Intel 8080 and 8086 systems, "translation compatibility" is a technique for creating a source code program in one language using the source code of a program written in different language so that the created program performs the same function as the original program, although expressed in a different language. Mr. Paterson used translation compatibility to enable third party programs that relied on their interaction with CP/M to be translated so that they had exactly the same interactions with 86-DOS. For translation compatibility to be successful, 86-DOS is required to replicate the proven and dominant CP/M OS interface and behavior – i.e. the essential CP/M design. As a consequence Mr. Paterson did not need to possess any deep knowledge of operating systems or to expend any resources on a general design of a new operating system. Mr. Paterson has freely admitted that he did not want to design an interface; rather he wanted to use one that had proven itself in the market. He also states that CP/M was the only acceptable choice.[27]

53. Mr. Paterson has repeatedly claimed that he used "translation compatibility" to adapt third party source programs to 86-DOS. He states: "You could translate that [CP/M 8080 source] program with the translator, producing 8086 code and in the process, *the detailed mechanisms of how you would have asked the operating system to perform a*

[25] Paterson Decl. ¶ 6, at 3.
[26] Hollaar Decl. ¶ 52, at 12.
[27] Paterson Decl. ¶¶ 5-6, at 2-3.

task, such as saving a file to disk, those individual instructions would be translated into 8086 instructions and that sequence of 8086 instructions that came out of the translation process would in fact cause 86-DOS to also save a file."[28] (Emphasis added). In other words, the translation program effectively changed symbolic machine operation code ("opcode") names from 8080 or Z80 symbolic assembly language to 8086 symbolic assembly language without changing the semantics of the call in the application program. If the original Z80 program called a CP/M function, such as conin, then the translated 8086 program would use the appropriate 8086 opcode and function number to call the OS function named conin in the 8086 version of the program. (Mr. Paterson provides a detailed example of this in his writings; the example also illustrates dynamic linking since it uses a table index – a number rather than a function name – to invoke to a system function.)[29] Hence if the resulting 8086 program were then translated into machine language on a machine with 86-DOS, it would call a function named conin, exactly corresponding to the function it would have called on an 8080 machine with CP/M. Translation compatibility, then, depends on 86-DOS having the same OS function, with the same syntax and same semantics as CP/M, meaning that the application program could not distinguish between using CP/M in the 8-bit world versus using 86-DOS in the 16-bit world.

J. In Replicating CP/M's System Call Interface, 86-DOS Replicated Significant Aspects of the CP/M Design

[28] Paterson Dep. at 48-49.

[29] Paterson, *supra* note 14, at 46.

54. Both Mr. Paterson[30] and Professor Hollaar[31] state that the system call interface can be thought of as a simple set of labels that map to varying implementations. Professor Hollaar even states that the system call interface is no more significant than the shift pattern in an automobile with a standard transmission. This is a completely naïve position in OS design. The system call interface defines (1) the names of functions exported by the OS, (2) the function calling sequences, (3) the semantics of invoking the function, (4) a specification of the effect of calling each function, and (5) sometimes important internal design constraints.

55. According to Dr. Butler Lampson, one of the leading OS researchers in the past 40 years, the choice of functions is the essence of system design: "The most important hints, and the vaguest, have to do with obtaining the right functionality from a system, that is, with getting it to do the things you want it to do. … *Defining interfaces is the most important part of system design. Usually it is also the most difficult*, since the interface design must satisfy three conflicting requirements: an interface must be simple, it should be complete, and it should admit a sufficiently small and fast implementation."[32] (Emphasis added). This explains why it took Dr. Kildall (a person with a Ph.D. in computer science and many years of system programming experience) 2-3 years to design and implement CP/M, yet it only took Mr. Paterson (a person with limited software experience and essentially no experience creating an OS) two staff-months to create a "translation compatible" OS.

[30] Paterson Decl. ¶ 9, at 3.

[31] Hollaar Decl. ¶¶ 32-34, at 8.

[32] Butler W. Lampson, *Hints for Computer System Design*, *ACM Operating Systems Review*, 15, 5 (October 1983), 33-48, a*vailable at* http://research.microsoft.com/%7Elampson/33-Hints/WebPage.html.

56. Paterson attempts to minimize the importance of the system call interface by saying: “I did use CP/M as the model for the specific disk functions since I had decided to attempt translation compatibility. The functions themselves, such as the facilities to open, close, read, and write, are present in any operating system.”[33] Such adoption, however, implied that 86-DOS would not only have to replicate the syntax of each system call, but also create OS functionality that replicated the internal semantic behavior of each system call. To replicate such function semantics, 86-DOS must also represent the aspects of the internal OS state that an application program might perceive in CP/M in order to ensure that 86-DOS’ OS function execution behaves the same as the CP/M function execution.[34]

57. Mr. Evans correctly observed that Mr. Paterson rewrote the implementation of the functions, without doing any of the design work. Even years later, Mr. Paterson appears not to recognize this crucial distinction between design and implementation, conflating his discussion of design and implementation/mechanics. Mr. Paterson does not seem to recognize that Mr. Evans’ statements generally address design, and in some cases, do not refer to the implementation at all.[35]

K. Mr. Paterson’s Knowledge of CP/M Belies Claim That He Never Looked at its Algorithms and Data Structures

58. Mr. Paterson worked with Cromemco CDOS for an extended period of time, both at The Retail Computer Store and at SCP, and he likely knew how to disassemble an object code version of CDOS using the debugger and other tools. Even if

[33] Paterson Decl. ¶ 12, at 4.

[34] CP/M’s file control block is an example of such internal state. Even though the 86-DOS file system was implemented differently from CP/M, application program execution behavior could depend on the existence and state of a file control block.

[35] Paterson Decl. ¶ 11, at 4.

Mr. Paterson never had access to CP/M source or object code, he did have access to the object code of a CP/M "look-alike" (CDOS), and thus had the capability to disassemble CDOS object code to inspect internal algorithms and data structures. If CDOS was, indeed, a copy of CP/M, Mr. Paterson's inspection of the operation and design of CDOS would have familiarized him with the operation and design of CP/M.

59. Given Mr. Paterson's broad knowledge of the internal workings of CP/M, and the fact that this knowledge could not have been obtained by only reading a manual, Dr. Kildall is most likely correct that Mr. Paterson somehow looked at CP/M's algorithms and data structures.

L. Mr. Evans Use of "Clone" Accurately Describes How 86-DOS Compared to CP/M

60. Professor Hollaar states that "… there is no precise definition of 'clone'…"[36] suggesting that in his understanding of the word, any author, including Mr. Evans, could use the word clone to describe any two entities with similar characteristics. Mr. Evans makes clear that the 8080 and 8086 are different hardware, and that in his use of the word "clone," he argues that 86-DOS is a "clone" of CP/M for reasons relating to the substitution of 86-DOS as an OS that supports the execution of software written to use the CP/M OS. In Sections H, I and J, I have explained the deeper technical rationale that supports Mr. Evans' use of the term in the context of OS programming interfaces.

M. Mr. Paterson is Unable to Demonstrate that 86-DOS was not a Clone of CP/M

61. Mr. Paterson has testified that he does not possess a copy of the source code for any version of 86-DOS prior to Version 1.25.[37] Mr. Paterson produced a paper copy

[36] Hollaar Decl. ¶ 27, at 7.

[37] Paterson Dep. at 27.

of the Version 1.25 86-DOS source code. That source code listing shows that there were at least 20 versions of 86-DOS between December 29, 1980 and March 3, 1982 (the date of Version 1.25). Mr. Paterson has testified that there were additional versions prior to December 29, 1980, starting no later than his first working version in August 1980.[38] However, he is unable to produce source code that conclusively proves that he did or did not copy any aspects of 8-bit CP/M in these early versions of 86-DOS.

III. LIST OF PUBLICATIONS AUTHORED IN THE PAST TEN YEARS

62. *See* Exhibit B, *Curriculum Vitae.*

IV. COMPENSATION

63. I am being compensated at a rate of $350 per hour, and my compensation does not depend on the outcome of this case.

V. LIST OF CASES IN WHICH I HAVE TESTIFIED AS AN EXPERT

64. *See* Exhibit B, *Curriculum Vitae.*

DATED this 5th day of June, 2007.

By Gary J. Nutt
Gary J. Nutt

[38] *Id.* at 34.

CERTIFICATE OF SERVICE

I hereby certify that on June 6, 2007, sent a copy of the attached document to the following:

D. Michael Tomkins
Dietrich Biemiller

s/ Bruce E. H. Johnson
Bruce E. H. Johnson

EXHIBIT A

<u>Materials Used</u>

1. "Microprocessor Quick Reference Guide," Intel press materials, *available at* <u>http://www.intel.com/pressroom/kits/quickreffam.htm</u>.

2. *An Introduction to CP/M Features and Facilities*, Digital Research, 1976.

3. *History of the Microcomputer Revolution*, (KPBX FM 91.1 radio broadcast 1995).

4. Complaint, *Paterson v. Little, Brown & Co.*, No. 2:05-CV-01719-TSZ, (W.D. Wash. filed Feb. 28, 2005).

5. David H. Ahl, THE FIRST WEST COAST COMPUTER FAIRE, THE BEST OF CREATIVE COMPUTING, 98-127 (1980).

6. Harold Evans, *et al.*, THEY MADE AMERICA (2006).

7. Declaration of Professor Lee A. Hollaar in support of Plaintiffs' Response to Defendants' Motion for Summary Judgment.

8. Butler W. Lampson, *Hints for Computer System Design, ACM Operating Systems Review*, 15, 5 (October 1983), 33-48, a*vailable at* <u>http://research.microsoft.com/%7Elampson/33-Hints/WebPage.html</u>.

9. Hans-Peter Mesmer, THE INDISPENSABLE PC HARDWARE BOOK: YOUR HARDWARE QUESTIONS ANSWERED 81 (2d ed. 1995).

10. Stephen P. Morse, *et al.*, "*Intel Microprocessors: 8008 to 8086*," in Daniel P. Siewiorek, *et al.*, COMPUTER STRUCTURES: PRINCIPLES AND EXAMPLES (1982).

11. Tim Paterson, THE RIGHT PLACE … THE RIGHT TIME 43 (1995) (unpublished).

12. Declaration of Tim Paterson in support of Plaintiffs' Response to Defendants' Motion for Summary Judgment.

EXHIBIT B

Curriculum Vitae: Gary J. Nutt

2736 Third Street, Department of Computer Science, CB 430
Boulder, CO 80304University of Colorado
(303) 818-4344Boulder, CO 80309-0430
(303) 443-5436 (303) 492-7581

SUMMARY OF INTERESTS, SKILLS AND EXPERIENCE

Broad perspective on software and network technologies, reached from over 25 years of research experience and 10 years of commercial experience. Technology specialist in operating systems and software; distributed systems; soft real-time systems (including digital CATV technology), wireless and mobile computing; and internet content delivery.

PROFESSIONAL POSITIONS

November, 1986 – present
Professor
Department of Computer Science
University of Colorado
Boulder, Colorado

Teaching and research in distributed and network systems, operating systems, real-time systems, collaboration technology, multimedia, visual programming, and computer science curriculum development. Chair of the Department of Computer Science (1995-97); this academic unit had about 30 faculty and 30 research and support staff. Founder and Director of the Center for Software Systems Science (1988-94); the purpose of this center was to organize the systems and software faculty and staff, providing an interface to the software industry. Co-founder and Director of the Colorado Open Systems Consortium (1992-93); the members of this user organization were CIOs and their designates in Information Systems organizations in the Colorado Front Range area. Industrial consultant and legal expert.

September, 2000 – April, 2001
Senior Scientist
Emerging Technologies Group
Inktomi Corporation
Foster City, CA

[*I held this position while on leave from the University of Colorado.*] Technical responsibility identifying intellectual property and for building the corporate patent portfolio, including writing invention disclosures and working with corporate and outside counsel. Focused on technologies related to applied distributed systems, internet caching (terrestrial and satellite), streaming media (audio/visual) data management, and wireless computing.

April, 2000 – August, 2000
Vice President, Engineering
Bookface.com, Inc.

San Francisco, CA

[*I held this position while on leave from the University of Colorado.*] Bookface technology was focused on scalable infrastructure, security and digital rights management for text-based content. Worked with the CTO and CEO prior to first round financing. Responsible for engineering management, recruiting, external technology interactions (patents, security audit, software acquisition, co-location facility, technical advisory board), and general executive duties. . Staffed engineering group (opened a development office in Boulder). Drafted a 46-page patent specification for Electronic Media System and Method, that was submitted as a provisional patent in June, 2000.

March, 1984 - November, 1986
Vice President, Colorado Technical Office
Interactive Systems Corporation
(Headquarters in Santa Monica, California)
Boulder, Colorado

Established and managed a successful cost center responsible for operating system, network and application software products. Helped formulate the business and marketing plans for the network and application products. This business unit was eventually split and acquired by two large companies. Corporate Officer. Reported to the President of the Corporation.

April, 1981 - March, 1984
Engineering Director
NBI, Inc.
Boulder, Colorado

Built and managed an organization of 85 hardware and software engineers. Responsible for all hardware and software development for a product line of workstations and servers. Reported to the Vice President of Engineering.

June, 1980-April, 1981
Senior Member of the Technical Staff
Bell Laboratories
Denver, Colorado

Responsible for technical assessment and direction of data communications services for office information systems in PBX environments. Member of a corporate task force studying ISDN services.

July, 1978 - June, 1980
Member of Research Staff
Xerox Palo Alto Research Center
Palo Alto, California

Studied application of computers and networks to office information systems. Co-authored the first widely-recognized paper on computer science and office automation (including an introduction of workflow).

August, 1972 - July, 1978
Associate Professor (tenured)
Department of Computer Science
Boulder, Colorado

Teaching and research in computer system organization and performance.

1969-1972	1967-1969	1964-67
Research Assistant and Student	Engineer and Programmer	Draftsman and Programmer
University of Washington	Boeing Aircraft	Idaho Power Company
Seattle, Washington	Seattle, Washington	Boise, Idaho

CONSULTING RELATIONSHIPS

AT&T Information Systems (1987)

Bell Laboratories (1976-78)

Bull Worldwide Information Systems (1993)

CableLabs, Inc. (1997-98)

Convex Computer Corporation (1994)

Eastman Kodak Company (1989-91)

Ford Aerospace Corporation (1988)

Hewlett-Packard (1976)

Interactive Systems Corporation (1986-93)

Martin Marietta Corporation (1988)

Microsoft (1998-2003)

NASA Langley ICASE (1978)

National Center for Atmospheric Research (1976) NCR (1992-93)

Quark (1997)

Seybold Office Computing Group (1987-8)

Storage Technology Corporation (1978, 1996)

U.S. Department of Interior, BLM (1976)

U.S. Department of Commerce, NBS (1977

Xerox Palo Alto Research Center (1977).

LEGAL EXPERIENCE

Special Master

Case: *Religious Technology Center v. F.A.C.T.Net*, United States District Court, Case No. 95-K-2143, 1995-99. Special Master to Honorable John L. Kane, Jr.

Case: *Corporate Express Office Products, Inc. v. EON Enterprises, et al.*, District Court, Jefferson County, Colorado, Case number 2001CV1316, 2002. Special Master to Honorable Jack W. Berryhill.

Court-appointed Expert

Case: *Arnold Pohs, et al., v. Steven Gotsdinger and Gotsdinger and Associates, LLC*, District Court, Arapahoe County, Colorado, Case Number 03 CV 1672, 2004. Court-appointed expert, Hon. Gerald Rafferty, Division 408. James B. Powers, Harris, Karstaedt, Jamison & Powers (Plaintiff), 383 Inverness Parkway, Suite 400, Englewood, CO 80112-5816. Erich L. Bethke, Canges, Iwashko, Bethke & Bailey (Defendant), 950 Seventeenth Street, Suite 1700, Denver, CO 80202.

Computer Forensics

Case: *Religious Technology Center v. F.A.C.T.Net*, U.S. District Court, Case No. 95-K-2143, 1995. Expert retained by the defendant. Testified in a hearing. Tom Kelley and Natalie Hanlon-Leh, Faegre and Benson, LLP, 317 17th Street, Denver, CO. This work preceded my appointment as Special Master on this case.

Case: *Kensington Apartments v. David Canola*, preparation for potential civil action, 1997. Expert retained by the plaintiff. Christopher M. Leh, Caplan and Earnest, LLP, 2595 Canyon, Suite 400, Boulder, CO.

Case: *People v. Bernstein Brothers, City of Denver*, 1998. Expert retained by the defendant Bernstein Brothers. Gary Lozow of Isaacson, Rosenbaum, Woods, and Levy, 633 17th Street, Denver, CO.

Case: *Ralph A. Reiff v. Rocky Mountain News*, U.S. District Court, Case No. 98-Z-1658, 1999. Expert retained by the plaintiff. Richard S. Shaeffer, attorney, 2063 S. Worchester Way, Aurora, CO.

Case: *Marconi Communications Inc. v. Ellingford*, SR File No. 4416-1, 2000. Expert retained by the plaintiff. Todd Blakely and Robert R. Brunelli of Sheridan Ross, 1560 Broadway, Suite 1200, Denver, CO.

Case: *Corporate Express Office Products, Inc. v. EON Enterprises, et al.* District Court, Jefferson County, Colorado, Case number 2001CV1316, 2002. Expert retained by the plaintiff. Darrell M. Daley of Faegre & Benson, LLP, 1900 Fifteenth Street, Boulder, CO

Case: *Allan L. Pallarito v. Storage Technology Corporation and Randy Settergren*. United States District Court of Colorado, Case number 01 M 222, 2001. Expert retained by the defendant. Christopher M. Leh of Holland & Hart LLP, 1050 Walnut Street, Boulder, CO.

Case: *People v. Bruce Gillie*, 01CR338, 2002, Eagle County, Colorado. Expert retained by the defendant. David Lane of Kilmer & Lane, 1543 Champa Street, Suite 400, Denver, CO.

Case: *Xcel Energy Inc. v. Andrew Olsen (individually) and Andrew Olsen, Charles L. Butland and Kristen Tollefsen (dba Colorado Energy Services)*. .District Court, City and County of Denver, Case number 01 CV 3021, 2001. Expert retained by the plaintiff. Natalie Hanlon-Leh, Faegre & Benson LLP, 370 Seventeenth Street, Denver, CO.

Case: Internal system investigation. Adams Twelve Five Star School District, Phil Padilla, Educational Support Center, 1500 East 128th Avenue, Thornton, CO.

Case: *The Retirement Group v. Linsco/Private Ledger, et al.*, Superior Court of the State of California, San Diego County, Case number GIC786976, 2002. Expert retained by the defendant. Kevin Allen, Allen & Vellone, 1600 Stout Street, Suite 1100, Denver, CO.

Case: *Corporate Express Office Products, Inc. v. Karen Young and EON Office Products, Inc.*, United States District Court for the District of Colorado, Case Number 03-B-1131 (MJW), 2003. Expert retained by the defendant. Kevin Allen, Allen & Vellone, 1600 Stout Street, Suite 1100, Denver, CO.

Case: *Corporate Express Office Products, Inc. v. Karen Young and EON Office Products, Inc.*, United States District Court for the District of Colorado, Case Number 03-B-1131 (MJW), 2003. Expert retained by the plaintiff. Heather Perry, Connelly Sheehan & Moran, LLP, 150 S. Wacker Drive, Suite 1600, Chicago, IL 60606.

Case: *People v. French, et al. (KidZtime)*, 2004. Expert retained by the Jean W. Walters and Jack Luellen, Office of Attorney General, 1525 Sherman St., 5th Floor, Denver, CO 80203.

Case: *Palo Duro Hardwoods, Inc. v. Thomas J. Heese*, United States District Court for the District of Colorado, Case Number 04-D-0459 (PAC), 2003. Expert retained by the plaintiff. Kevin Allen, Allen & Vellone, 1600 Stout Street, Suite 1100, Denver, CO.

Case: *Western Stone & Metal Corp. v. Chris Riggs and John A. Wegman*, United States District Court for the District of Colorado, Case Number 05-cv-799-WYD-BNB, 2005. Expert retained by the defendants. F. Stephen Collins, Ducker, Montgomery, Aronstein & Bess, P.C., One Civic Center Plaza, 1560 Broadway, Suite 1400, Denver, CO 80202. James W. Hubbell, Kelly Haglund, Garnsey Kahn, LLC, 1441 18th Street, Suite 300, Denver, CO 80202.

Patents, Intellectual Property, and Trade Secrets

Case: Seismic Image Software, Ltd. and Green Mountain Geophysics, Inc., American Arbitration Association, No. 77 117 00250 96, 1997. Testifying expert retained by the defendant. Deposed and testified in arbitration. William D. Meyer, Hutchinson, Black and Cook, LLP, 921 Walnut Street, Suite 200, Boulder, CO.

Case: *Joseph H. Mohr, d/b/a Roaring Fork Software v. VICORP Restaurants, Inc.*, Denver District Court, Case No. 96 CV 6526, 1997. Testifying expert retained by the defendant. Deposed. Wesley B. Howard, Brega and Winters, 1700 Lincoln Street, Suite 2222, Denver, CO.

Case: *Media Optik, Inc. v.. Sega of America*, Validity Investigation of U.S. Patent 5,012,407, 2000. Testifying expert retained by the defendant. John M. Romary, Finnegan, Henderson, Farabow, Garrett, and Dunner, LLP, Washington, D.C.

Case: EdiSync patent, testifying expert retained by plaintiff, 2004-07, David Carroll and John Kennedy, Dorsey & Whitney, LLP, 370 Seventeenth St., Denver, CO 80202.

Case: Consulting expert, 2005. Robert Chiaviello, Jr., Fulbright & Jaworski, LLP, 2200 Ross Ave., Suite 2800, Dallas, TX, 75201-2784.

Case: *American Video Graphics v.. Microsoft Corporation*, U. S. District Court, Eastern District of Texas, Civil Action No. 6:05-CV-006, 2005. Testifying expert retained by defendant, Timothy E. DeMasi, Weil, Gotshal & Manges, LLP, 767 Fifth Avenue, New York, NY 10153.

Case: Consulting expert, 2005-06. Robert Silver, Boies, Schiller & Flexner, LLP, 333 Main Street, Armonk, NY, 10504. Stuart H. Singer, Boies, Schiller & Flexner, LLP, 401 East Las Olas Blvd., Suite 1200, Fort Lauderdale, FL 33301-2211.

Case: Consulting expert, 2005. James Pinto and Dorsey & Whitney, LLP, 370 Seventeenth St., Denver, CO 80202. Devan Padmanabhan., Dorsey & Whitney, LLP, 50 South Sixth Street, Suite 1500, Minneapolis, MN 55402-1498.

Case: *UniSys Corporation v. Accenture LLP*, District Court, County of Denver, Colorado, Case No. 04CV1509, 2005. Testifying expert retained by the plaintiff. Deposed. Gregory Kanan and Jesús Vásquez, Rothgerber, Johnson & Lyons, LLP, 1200 Seventeenth Street, Suite 3000, Denver, CO 80202-5855.

Case: Registrar Systems patent, 2006-07. Testifying expert for plaintiff, John Kennedy and Gregory S. Tamkin, Dorsey & Whitney, LLP, 370 Seventeenth St., Denver, CO 80202.

Case: Consulting expert, 2006, Mark J. Shean, Orrick, Herrington & Sutcliffe, LLP, 4 Park Plaza, Suite 1600, Irvine, CA 92614-2558.

Case: Consulting expert, 2006. Thomas C. Webster, Blakely, Sokoloff, Taylor & Zafman, LLP, 1279 Oakmead Parkway, Sunnyvale, CA 94085-4040.

Case: Consulting expert, 2006. William D. Meyer, Hutchinson, Black & Cook, LLP, 921 Spruce St., Suite 200, Boulder, CO 80302.

Case: Consulting expert, 2006. David L. McCombs, Haynes and Boone, LLP, 901 Main St., Suite 3100, Dallas, TX 75202.

Case: *Adept Computer Systems v. Zoll Data Systems*, U. S. District Court, Colorado District, Case No. 05-cv-01917-ZLW-MJW, 2006-07. Testifying expert retained by the defendant. Deposed. Natalie Hanlon-Leh, Faegre and Benson, LLP, 3200 Wells Fargo Center, 1700 Lincoln Street, Denver, CO 80203-4532. Mindy Sooter, Faegre and Benson, LLP, 1900 Fifteenth Street, Boulder, CO.

Case: *Veritas Operating Corporation v. Microsoft Corporation*, U. S. District Court, Western District of Washington, Case No. C-06-0703, 2006. 2006-07. Testifying expert retained by defendant. Bruce R. Braun and Pei Yuan Chung, Winston & Strawn, LLP, 35 W. Wacker Drive, Chicago, IL 60601-9703. Todd M. Siegel, Klarquist Sparkman, LLP, One World Trade Center, 121 S. W. Salmon Street, Suite 1600, Portland, OR 97204-2988.

Case: *Paterson v. Evans, et al.*, U. S. District Court, Western District of Washington, Case No. 2:05-CV-01719-TSZ, 2006-07. Expert retained by the defendant. Bruce E. H. Johnson and

Kaustuv M. Das, Davis Wright Tremaine, LLP, 2600 Century Square, 1501 Fourth, Avenue, Seattle, WA 98101-1688.

Case: *Sun Microsystems v. Computer Hardware Equipment*, U. S. District Court, District of Colorado, Case No. 06-cv-01699 MSK-MJW, 2006. Expert retained by the plaintiff. Nancy Gegenheimer and Lucky Vidmar, Holme Roberts and Owen, LLP, 1700 Lincoln Street, Suite 4100, Denver, CO 80203-4541.

Case: *QuickPen International, Inc. v. Daniel P. Bittinger*, U. S. District Court, District of Colorado, Case No. 06-CV-02265 DME-MEH, 2006. Expert retained by the plaintiff, 2007. Stuart Pack and Byeongsook Seo, Isaacson Rosenbaum, P.C., 633 17th Street, Suite, 2200, Denver, CO 80202-5656.

EDUCATION

Ph.D., *The Formulation and Application of Evaluation Nets*, Computer Science, University of Washington, Seattle, Washington, 1972.

M.S., *Performance Evaluation of a CDC 6600*, Computer Science, University of Washington, Seattle, Washington, 1970.

B.A., Mathematics, Boise State University, Boise, Idaho, 1967.

Ph.D. Students

1. Steinke, Robert C., “Consistency Model Transitions for Shared Memory,” May, 2001.

2. Siewert, Sam, “A Real-Time Execution Performance Agent Interface for Confidence-Based Scheduling,” December, 2000.

3. Adam Jonathon Griff, “Gryphon: A Dynamically Tailorable Mechanism for Customizing Location and Caching Policies in Distributed Object Subsystem,” May, 2000.

4. Scott A. Brandt, “Soft Real-Time Processing with Dynamic QoS Level Resource Management,” August, 1999.

5. Richard L. Blumenthal, “Supporting Unstructured Activities with a Meta-Contextual Protocol in Situation-Based Workflow,” May, 1998.

6. Zulah K. F. Eckert, “Trace Extrapolation for Parallel Programs on Shared Memory Multiprocessors,” May, 1995.

7. Jeffrey D. McWhirter, “Characterization, Specification and Generation of Visual Language Applications,” May, 1995.

8. Adam L. Beguelin, “Deterministic Parallel Programming in Phred,” May, 1990.

9. Isabelle M. Demeure, “A Model, ParaDiGM, and a Software Tool, VISA for the Representation, Design and Simulation of Parallel, Distributed Computations”, August, 1989.

M.S. Students

1. Paul Griepentrog, “Network Enhancements to the Dynamic QoS Manger,” 1999.
2. James Grosvenor Garnett, “Distributed Phase and Frequency Synchronization,” 1999.
3. Shahzad Bhatti, “A 3-D Graphics Tool for Software Engineering,” 1998.
4. Paul J. Hamill III, “Internet Structure Visualization,” (ECE), 1998.
5. William D. Anderson, “An Empirical Study of Optimizations on the CM-5,” 1990.
6. Kimbal S. Smith, “An Interface for Interprocess Communication and Control with the MAP Architecture”, 1979.
7. Gregory A. Smith, “Input/Output as Process Communication: A Method of Evaluation,” 1978.
8. Bruce W. Sanders, “The Design and Simulation of a Bit Slice Implementation for MAP,” 1978.
9. Robert T. Krivacic, “The Refinement and Implementation of a Simulation System,” 1978.
10. Heinrich Siegman, “Design and Simulation of a Main Memory-Control Units Interface for the Multi Associative Processor System,” 1976.
11. Thomas Dodge, 1976 (jointly supervised with C. A. Ellis).
12. Janis P. Osterweil, “A Deadlock Model Based on Process-Resource Actions,” 1975.

PROFESSIONAL ACTIVITIES

- Chair of ACM SIGMETRICS (1979-1981).
- General Chair
  - CIO Forum Colorado Open Systems Consortium Workshop (1991)
  - 1990 ACM SIGMETRICS Conference on Modeling and Measurement of Computer Systems.
- Program Chair
  - OpenExpo (1993)

  - IEEE Lake Arrowhead Workshop on Office Information Systems (1980)
  - ACM Conference on Simulation, Modeling and Measurement of Computer Systems (1976, 1979).
- Program Committee
  - International Conferences on Applications and Theory of Petri Nets (1993, 1994, 1997-99)
  - NSF New Challenges for Directions in System Research, Program Committee (1997)
  - International Workshop on Petri Nets and Performance Models (1997)
  - International Conference on Parallel Processing (1994)
  - Annual Software Reliability Workshop (1992)
  - Conference on Office Information Systems (1982)
- Member of Organizing Committee of MASCOTS (1994).
- Local arrangements chair of ACM Conference on Simulation, Modeling and Measurement of Computer Systems (1975).

GRANTS

1. “OS Kernel Lab,” University of Colorado, Engineering Excellence Fund, 2006-7, $51,850.
2. “OS Kernel Lab Renovation,” University of Colorado College of Engineering and Applied Science grant, $7,500.
3. “Gift for OS Kernel Lab,” Adam Beguelin, entrepreneur, 2005, $50,000.
4. “A Scalable, Flexible, Secure Educational Lab,” National Science Foundation proposal, 2003-2004.
5. “A Windows Server Facility for Supporting Specific Software Environments,” University of Colorado, Engineering Excellence Fund, $25,000, 2002. Also partially funded by Microsoft Corporation, $20,000.
6. “The Digital CommonSpace,” National Science Foundation proposal, 2000-2004.
7. “Small Computer Operating System Course,” University of Colorado, Engineering Excellence Fund, $50,000, 2000. Also partially funded by Microsoft Corporation, $35,000.
8. “Technology for Very Large Scale Collaboration,” National Science Foundation Grant No. IRI-9732085, 1998-2000.

9. “Architectures for Workflow Systems,” National Science Foundation Grant No. IRI-9307619, 1994 to 1998

10. “CU-Convex Joint Research Program,” Convex Computer Corporation, 1993 to 1994.

11. “Generating Visual Frontends,” NCR Corporation, 1992 to 1993,

12. “Colorado Open Systems Consortium Program Plan,” Colorado Advanced Technology Institute, 1992 to 1993.

13. “FlowWorks Research and Development,” Bull S. A., IOS, 1992 to 1994.

14. “Undergraduate Participation in Research Network Administration and Parallel Program Performance (Supplement to CDA-8922510), National Science Foundation, 1991 to 1992.

15. “Effective Use of Parallel and Distributed Computing,” National Science Foundation, CDA-8922510, 1990 to 1995.

16. “CU - U S West Partnership Program,” U S West Advanced Technologies Grant, 1990 to 1993.

17. “Center for Software Systems Science ME SE Planning,” U S West Advanced Technologies Grant, 1990 to 1991.

18. “Petri Net Simulation System,” Ford Aerospace and Communications Corporation, 1990 to 1991.

19. “Parallel Program Modeling and Evaluation,” National Science Foundation Grant CCR-8802283, 1988 to 1990.

20. “Systems Laboratory Proposal,” AT&T, 1988 to 1989.

21. “Software Development Environment Research,” U S West Advanced Technologies Grant, 1987 to 1989.

22. “Research on Parallel Processing Using a Hypercube,” Amoco Production Company, 1987 to 1988.

23. “Systems Research Laboratory Equipment Grant,” Sun Microsystems, Inc., 1987.

24. “Systems Research Laboratory,” University of Colorado Research Initiation Grant, 1986.

25. “Coordinated Experimental Research: A Facility for Research in Numerical Computation and Software Environments,” National Science Foundation Cooperative Agreement No. DCR-8420944, 1986 to 1991.

26. “Measuring Performance for Associative and Array Processors,” National Science Foundation Grant MCS74 08328 A01, 1974 to 1978.

PATENT ACTIONS

Patent Portfolio Administration: At Inktomi, I interviewed engineers to identify patentable technology, to prepare an initial disclosure to corporate counsel, then to work with outside counsel to prepare patent specifications.

Patent Application: Nutt, Gary, Vikas Jha, Chung-Kei Wong, Ashok Sudarsanam, Spyro Papademetriou, and Anshu Aggarwal, "Delta Encoding Using Canoncal Reference Files," provisional patent specification, April, 2001.

Provisional Patent Application: Deuster, Tammy, Peter Mattis, and Gary Nutt, "Electronic Digital Media System and Apparatus," provisional patent specification, May, 2000.

PUBLICATIONS

Books

1. Nutt, Gary, *Distributed Virtual Machines: Inside the Rotor CLI*, First Edition Addison Wesley, 2005, ISBN 0-321-15983-7.

2. Nutt, Gary, *Operating Systems: A Modern Approach*, Third Edition, Addison Wesley, 2004, ISBN 0-201-77344-9, 928 pages. (Second Edition Lab Update, 2002, ISBN 0-201-74196-2, 704 pages; Second Edition Addison Wesley, 2000, ISBN 0-201-61251-8, 650 pages; First edition, Addison Wesley, 1997, ISBN 0-8053-12951, 650 pages.)

3. Nutt, Gary, *Kernel Projects for Linux*, Addison Wesley, 2001, ISBN 0-201-61243-7, 240 pages.

4. Nutt, Gary, *Operating System Projects Using Windows NT*, Addison Wesley, 1999, ISBN 0-201-47707-6, 250 pages.

5. Nutt, Gary J., *Centralized and Distributed Operating Systems*, Prentice Hall, 1992, ISBN 0-13-122326-7, 418 pages.

6. Nutt, Gary J., *Open Systems*, Prentice Hall, 1992, ISBN 0-13-636234-6, 292 pages.

Book Chapters

1. Nutt, Gary J., "Simulation and Modeling, " to appear in *Encyclopedia of Distributed Computing*, Kluwer Academic Publishers, 2001.

2. Nutt, Gary J., "Set-top Boxes," *Encyclopedia of Electrical and Electronics Engineering*, John Wiley and Sons, 1999.

3. Eckert, Zulah K. F. and Gary J. Nutt, "Tracing Nondeterministic Programs on Shared Memory Multiprocessors," *Advanced Computer Performance Modeling and Simulation*, edited by Kallol Bagchi, Jean Walrand, and George Zobrist, Gordon and Breach Science Publishers, Amsterdam, Chapter 5, pp. 93-104, 1998.

4. Rover, Diane T., Allen D. Malony, and Gary J. Nutt, "Summary of Working Group on Integrated Environments V.. Toolkits", *Debugging and Performance Tuning for Parallel Computing Systems*, edited by Margaret L. Simmons, Ann H. Hayes, Jeffrey S. Brown, and Daniel A. Reed, IEEE Computer Society Press, pages 371-389, 1996.

Journals

1. Nutt, Gary, "Addressing Small Computers in the First OS Course," *Journal on Educational Resources in Computing*, Vol. 6, No. 2, June, 2006, pp. 1-13.

2. Steinke, Robert C. and Gary J. Nutt, "A Unified Theory of Shared Memory Consistency," *Journal of the ACM*, Vol. 51, No. 5, September, 2004, pp 800-849.

3. Brandt, Scott and Gary J. Nutt, "Flexible Soft Real-Time Processing in Middleware," *Real-Time Systems*, Vol. 22, No. 1-2, January-March, 2002, pp. 77-118.

4. Nutt, Gary, Scott Brandt, Adam Griff, Sam Siewert, Marty Humphrey, and Toby Berk, "Dynamically Negotiated Resource Management for Virtual Environment Applications," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 12, No. 1, January/February 2000, pp. 78-95.

5. Nutt, Gary J., "The Evolution Toward Flexible Workflow Systems," *Distributed Systems Engineering*, Volume 3, Number 4, December 1996, pp. 276-294.

6. Demeure, Isabelle M. and Gary J. Nutt, "Prototyping and Simulating Parallel, Distributed Computing,*" Journal of Parallel and Distributed Computing*, Vol. 23, No. 1 (October, 1994), pp. 1-9.

7. Beguelin, Adam L. and Gary J. Nutt, "Visual Parallel Programming and Determinacy: A Language Specification, an Analysis Technique, and a Programming Tool,*" Journal of Parallel and Distributed Computing*, Vol. 22, No. 2 (August, 1994), pp. 235-250.

8. Nutt, Gary J., "Book Review: Coloured Petri Nets Basic Concepts, Analysis Methods and Practical Use, Volume 1," by Kurt Jensen, Springer-Verlag, 1992 (234 pages). The review appears in *ACM SIGOPS Operating Systems Review*, Vol. 28, No. 1 (January, 1994), pp. 1-2.

9. Nutt, Gary J., "A Simulation System Architecture for Graph Models," *Advances in Petri Nets 90*, Springer-Verlag, 1990, pp. 417-435.

10. Nutt, Gary J., "Cognitive Congruity," *UNIX Review*, Vol. 4, No. 6 (June, 1986), pp. 38-46.

11. Nutt, Gary J., "An Experimental Distributed Modeling System,*" ACM Transactions on Office Information Systems*, Vol. 1, No. 2 (April, 1983), pp. 117-142.

12. Nutt, Gary J. and D. L. Bayer, "Performance of CSMA/CD Networks under Combined Voice and Data Loads," *IEEE Transactions on Communications*, Vol. COM-30, No. 1 (January, 1982), pp. 6-11.

13. Nutt, Gary J. and P. A. Ricci, “Quinault: An Office Modeling System,” *IEEE Computer*, Vol. 14, No. 5 (May, 1981), pp. 41-57.

14. Ellis, C. A. and G. J. Nutt, “Office Information Systems and Computer Science,” *ACM Computing Surveys*, Vol 12, No. 1 (March, 1980), pp. 27-60.

15. Osterweil, J. and G. J. Nutt, “Modeling Process-Resource Activity,” *International Journal of Computer Mathematics*, Vol. 7, No. 1 (1979), Section A, pp. 21-35.

16. Nutt, Gary J., “A Case Study of Simulation as a Computer System Design Tool,” *IEEE Computer*, Vol. 11, No. 10 (October, 1978), pp. 31-36.

17. Nutt, Gary J., “A Comparison of PASCAL and FORTRAN as Introductory Programming Languages,” *ACM SIGPLAN Notices*, Vol. 13, No. 2 (February, 1978), pp. 57-62.

18. Nutt, Gary J., “A Parallel Processor Operating System Comparison,” *IEEE Transactions on Software Engineering*, Vol. SE-3, No. 6 (November, 1977), pp. 467-475.

19. Nutt, Gary J., “Memory and Bus Analysis of an Array Processor,” *IEEE Transactions on Computers*, Vol. C-26, No. 6 (June, 1977), pp. 514-521.

20. Nutt, Gary J., W. A. Schulz and K. H. Williamson, “Generating Code for a Hypothetical Computer Using a Production Assembler,” *Software -- Practice and Experience*, Vol. 7, No. 1 (January-February, 1977), pp. 147-158.

21. Nutt, Gary J., “Some Resource Allocation Policies in a Multi Associative Processor,” *Acta Informatica*, Vol. 6, (1976), pp. 211-225.

22. Nutt, Gary J., “Computer System Resource Requirements of Novice Programming Students,” *Software – Practice and Experience*, Vol. 6, No. 1 (January-March, 1976), pp. 43-50.

23. Nutt, Gary J., “Computer System Monitors,” *IEEE Computer*, Vol. 8, No. 11 (November, 1975), pp. 51-61.

24. Nutt, Gary J. and J. D. Noe, “Macro E-Nets for Representation of Parallel Systems,*” IEEE Transaction on Computers*, Vol. C-22, No. 8 (August, 1973), pp. 718-727.

Conference Papers

1. Nutt, Gary , “Applying Shared Memory Consistency Models to Collaborative Editing,” *ACM SIGGROUP (Group 01) Collaborative Editing Workshop*, position paper abstract, October, 2001.

2. Nutt, Gary, “Group Digital CommonSpace,” NSF RI meeting, poster paper, July, 2001.

3. Steinke, Robert C. and Gary J. Nutt, "A Lattice Based Framework for Distributed Shared Memory Consistency Models,*" IEEE International Conference on Distributed Computer Systems*, poster paper, April, 2001.

4. Siewert, Sam and Gary Nutt, "Multi-Epoch Scheduling Within the Real-Time Execution Performance Agent Framework," *21st IEEE Real-Time Systems Symposium* (RTSS 2000), work-in-Progress, November 2000.

5. Nutt, Gary J., "Windows NT in the OS Curriculum," 3rd Usenix Windows NT Symposium, poster paper, July, 1999.

6. Siewert, Sam, Gary Nutt, and Elaine Hansen, "The Real-Time Execution Performance Agent: An Approach for Balancing Hard and Soft Real-Time Execution for Space Applications," *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, June, 1999.

7. Brandt, Scott, Gary Nutt, Toby Berk, and James Mankovich, "A Dynamic Quality of Service Middleware Agent for Mediating Application Resource Usage," *19th IEEE Real-Time Systems Symposium (RTSS '98)*, December 1998, Madrid, Spain.

8. Brandt, Scott, Gary Nutt, and Ken Klingenstein, "A Discrete and Dynamic Approach to Application/Operating System QoS Resource Management," *First Internet 2 Joint Applications/Engineering QoS Workshop (I2QoS '98)*, May, 1998, Santa Clara, CA, pp. 22-25.

9. Brandt, Scott, Gary Nutt, Toby Berk, and Marty Humphrey, "Soft Real-Time Application Execution with Dynamic Quality of Service Assurance," *Sixth IEEE International Workshop on Quality of Service (IWQoS '98)*, May 1998, Napa, CA, pp. 154-163.

10. Humphrey, Marty, Gary Nutt, Scott Brandt, and Toby Berk, "The DQM Architecture: Middleware for Application-centered QoS Resource Management,*" 1997 IEEE Workshop on Middleware for Distributed Real-Time Systems and Services*, December 1997, San Francisco, CA, pp. 97-104.

11. Nutt, Gary, Toby Berk, Scott Brandt, Marty Humphrey, and Sam Siewert, "Resource Management for a Virtual Planning Room," *1997 International Workshop on Multimedia Information Systems*, September, 1997, Como, Italy, pp. 129-134.

12. Siewert, Sam, Gary J. Nutt and Marty Humphrey, "Real-Time Parametrically Controlled In-Kernel Pipelines," Work-in-progress paper, *Third IEEE Real-time Technology and Applications Symposium*, June, 1997, Montreal, Canada.

13. Ellis, Clarence A. and Gary J. Nutt, "Multi-Dimensional Workflow," *The Second World Conference on International Design and Process Technolog*y, Society for Design and Process Science, Austin, Texas, December, 1996.

14. Siewert, Sam and Gary J. Nutt, "A Space System Testbed for Situated Agent Observability and Interaction," *The 5th Intl. Conf. and Exposition on Engineering, Construction, and*

Operations in Space and The 2nd Conf. and Exposition/Demonstration on Robotics for Challenging Environments, Aerospace Division of the American Society of Civil Engineers, Albuquerque, New Mexico, June 1996.

15. Ellis, Clarence A. and Gary J. Nutt, “Workflow: The Process Spectrum,” *NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions*, Athens, Georgia, pp. 140-145, May, 1996.

16. Blumenthal, Richard and Gary J. Nutt, “Supporting Unstructured Workflow Activities in the Bramble ICN System,” *Conference on Organizational Computing Systems*, ACM, August, 1995, Milpitas, CA, pp. 130-137.

17. Nutt, Gary J., “Software Engineering Process Model – A Case Study,*” Conference on Organizational Computing Systems*, ACM, August, 1995, Milpitas, CA, pp. 324-335.

18. Nutt, Gary J., Adam J. Griff, James E. Mankovich and Jeffrey D. McWhirter, “Extensible Parallel Program Performance Visualization,” *Proceedings of the Third International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS ‘95)*, January 18-20, 1995, Durham, NC, pp. 205-211.

19. McWhirter, Jeffrey D. and Gary J. Nutt, “Escalante: An Environment for the Rapid Construction of Visual Language Applications,” *Proceedings of the 1994 IEEE Symposium on Visual Languages*, October 4-7, 1994, St. Louis, MO, pp. 15-22.

20. Eckert, Zulah K. F. and Gary J. Nutt, “Parallel Program Trace Extrapolation,” *1994 International Conference on Parallel Programmin*g, St. Charles, Illinois, August 15-19, 1994, Volume II, pp. 103-107.

21. Nutt, Gary J., “A Parallel Program Tuning Environment,*” 1993 International Conference on Parallel Programming*, St. Charles, Illinois, August 16-20, 1993.

22. Ellis, Clarence A. Ellis and Gary J. Nutt, “Modeling and Enactment of Workflow Systems,” invited paper, *14th International Conference on Application and Theory of Petri Nets*, Chicago, Illinois, June 21-25, 1993, pp. 1-16.

23. Eckert, Zulah K. F. and Gary J. Nutt, “On the Complexity of Trace Migration for Parallel Programs on Shared Memory Machines,” abstract, *3rd ACM/ONR Workshop on Parallel and Distributed Debugging*, May, 1993, pp. 187-189.

24. McWhirter, Jeffrey D. and Gary J. Nutt, “Generation of Visual Language Environments,” short paper, *Interchi ‘*93, Amsterdam, The Netherlands, April 25-29, 1993.

25. Grunwald, D., G. Nutt, A. Sloane, D. Wagner, and B. Zorn, “A Testbed for Studying Parallel Programs and Parallel Execution Architectures,” *Proceedings of MASCOTS ’93,* San Diego, CA, January, 1993, pp 95-106.

26. Ellis, Clarence A. and Gary J. Nutt, “The Modeling and Analysis of Coordination Systems,” *Workshop position paper, ACM 1992 Conference on Computer-Supported Cooperative Work*, November, 1992, Toronto, Canada.

27. McWhirter, Jeffrey D. and Gary J. Nutt, “A Characterization Framework for Visual Languages,” *Proceedings of the 1992 IEEE Workshop on Visual Languages,* September, 1992, Seattle, pp. 246-248.

28. Grunwald, Dirk, Gary Nutt, Anthony Sloane, David Wagner, William Waite, and Benjamin Zorn, “Execution Architecture Independent Program Tracing,” poster session paper, *Supercomputing ‘92*, Los Alamos, NM, November, 1991.

29. Beguelin, Adam L. and Gary J. Nutt, “Examples in Phred,*” Proceedings of the Fifth SIAM Conference on Parallel Processing for Scientific Computing,* Houston, Texas, March, 1991, pp. 602-608.

30. Demeure, I. M. and Gary J. Nutt, “The VISA Distributed Computation Modeling System,” *Proceedings of the Fifth International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, Turin, Italy, February 13-15, 1991.

31. Nutt, Gary J., “Distributed Simulation Design Alternatives,” *Proceedings of the SCS Conference on Distributed Simulation,* San Diego, CA, January 17-19, 1990, pp. 51-55.

32. Demeure, I. M. Demeure, S. L. Smith, and Gary J. Nutt, “Modeling Parallel, Distributed Computations using ParaDiGM -- A Case Study: The Adaptive Global Optimization Algorithm,” *Proceedings of the Fourth SIAM Conference on Parallel Processing for Scientific Computing*, ed. by J. Dongarra, P. Messina, D. C. Sorensen, and R. G. Voight, December, 1989, pp. 154-161.

33. Nutt, Gary J, A. Beguelin, I. Demeure, S. Elliott, J. McWhirter, and B. Sanders, “Olympus: An Interactive Simulation System,” 1989 Winter Simulation Conference Proceedings, Washington, D.C., December 4-6, 1989, pp. 601-611.

34. Nutt, Gary J., “A Flexible, Distributed Simulation System,” *Tenth International Conference on Application and Theory of Petri Nets*, Bonn, West Germany, June, 1989, pp. 210-226.

35. Nutt, Gary J., “Visual Programming Methodology for Parallel Computations,” *MCC-University Research Symposium Proceedings,* Austin, Texas, July, 1987.

36. Nutt, Gary J., “Business User Interfaces for UNIX,” *Proceedings of Comunix ‘86*, (June, 1986), London, pp. 107-114.

37. Nutt, Gary J., “Information Flow Modeling for Office Automation,*” International Symposium on Office Automation,* Stanford University, March, 1980.

38. Nutt, Gary J. and Clarence A. Ellis, “Personal Computers in the Office: An Example,” *The Best of the Computer Faires*, Vol. V (March, 1980), pp. 101-107.

39. Ellis, Clarence A. and Gary J. Nutt, “Backtalk: An Office Environment Simulator,” *ICC ‘79 Conference Record*, (June, 1979), pp. 22.3.1-22.3.5.

40. Nutt, Gary J., “Microprocessor Implementation of a Parallel Processor,” *Proceedings of the Fourth Annual Computer Architecture Symposium*, (1977), pp. 147-152.

41. Nutt, Gary J., “A Parallel Processor for Evaluation Studies,” *AFIPS Proceedings of the NCC*, Vol. 45 (1976), pp. 769-775.

42. Nutt, Gary J., “Some Uses of Simulation in System Design,” *Proceedings of the Third Symposium on the Simulation of Computer Systems*, (1975), pp. 221-228.

43. Nutt, Gary J., “An Overview of a Multi Associative Processor System,” *Proceedings of the 1974 National ACM Conference*, pp. 101-104.

44. Nutt, Gary J., “The Simulation of Computer Systems in a University Environment,” *Proceedings of the Second Symposium on the Simulation of Computer Systems*, (1974), pp. 25-29.

45. Nutt, Gary J., “The Computer System Representation Problem,” *Proceedings of the First Symposium on the Simulation of Computer Systems*, (1973), pp. 145-149.

46. Nutt, Gary J., “Evaluation Nets for Computer Systems Performance Analysis,” *AFIPS Proceedings of the FJCC*, Vol. 41 (1972), pp. 279-285.

47. Nutt, Gary J. and Jerre D. Noe, “Validation of a Trace-Driven CDC 6400 Simulation,” *AFIPS Proceedings of the SJCC*, Vol. 40 (1972), pp. 749-757.

Technical Reports

1. Lin, Wang-ting and Gary Nutt, “Building on the BACKSLASH Algorithm,” University of Colorado Department of Computer Science Technical Report No. CU-CS-1022-06, December, 2006, 20 pages.

2. Nutt, Gary, “Revitalizing the Computer Science Curriculum,” University of Colorado Department of Computer Science Technical Report No. CU-CS-1020-06, December, 2006, 18 pages.

3. Nutt, Gary, “A Note on Scaling Wireless Sensor Networks,” University of Colorado Department of Computer Science Technical Report No. CU-CS-1001-05, December, 2006, 17 pages.

4. Deuster, Tammy, Peter Mattis, and Gary Nutt, “Electronic Digital Media System and Apparatus,” provisional patent specification, May, 2000.

5. Nutt, Gary J., “Windows NT in the OS Curriculum,” University of Colorado Department of Computer Science Technical Report No. CU-CS-880-99, February, 1999.

6. Brandt, Scott and Gary Nutt, “Performance Metrics for Soft Real-Time Systems,” University of Colorado Department of Computer Science Technical Report No. CU-CS-876-98, December, 1998.

7. Nutt, Gary J., “Software for Small, Communicating Computers,” University of Colorado Department of Computer Science Technical Report No. CU-CS-835-97, March, 1997.

8. Nutt, Gary J., Joe Antell, Scott Brandt, Chris Gantz, Adam Griff, and Jim Mankovich, “System Support for a Virtual Planning Room,” University of Colorado Department of Computer Science Technical Report No. CU-CS-800-95, December, 1995.

9. Nutt, Gary J., “Model-Based Virtual Environments for Collaboration,” University of Colorado Department of Computer Science Technical Report No. CU-CS-799-95, December, 1995.

10. Nutt, Gary J., “Software Engineering Process Model Case Study.” University of Colorado Department of Computer Science Technical Report No. CU-CS-760-94, July, 1994.

11. Eckert Zulah, K. F. and Gary J. Nutt, “Trace Change Points in Parallel Programs,” University of Colorado Department of Computer Science Technical Report No. CU-CS-730-94, July, 1994.

12. Eckert Zulah, K. F. and Gary J. Nutt, “Sequence Comparison for Parallel Executions,” University of Colorado Department of Computer Science Technical Report No. CU-CS-726-94, March, 1994.

13. McWhirter, Jeffrey D. and Gary J. Nutt “Experimenting with Visual Parallel Programming Laguages,” University of Colorado Department of Computer Science, March, 1994.

14. Eckert Zulah, K. F. and Gary J. Nutt, “A Framework for Execution Order Distortion in Shared Memory Multiprocessor Event Traces,” University of Colorado Department of Computer Science Technical Report No. CU-CS-708-94, January, 1994.

15. McWhirter, Jeffrey D. and Gary J. Nutt “Escalante: An Environment for the Rapid Construction of Visual Language Applications,” University of Colorado Department of Computer Science Technical Report No. CU-CS-692-93, December, 1993.

16. McWhirter, Jeffrey D., Zulah K. F. Eckert, and Gary J. Nutt “Building Visual Language Applications with Escalante,” University of Colorado Department of Computer Science Technical Report No. CU-CS-655-93, September, 1993.

17. Nutt, Gary J., “A Parallel Program Tuning Environment,” University of Colorado Department of Computer Science Technical Report No. CU-CS-664-92, August, 1993.

18. Nutt, Gary J. “Using Workflow in Contemporary IS Applications,” University of Colorado Department of Computer Science Technical Report No. CU-CS-663-92, August, 1993.

19. Nutt, Gary J., “FlowPATH Principles of Operation,” Bull Worldwide Information Systems, Imaging and Office Solutions, 7 rue Ampère, 91343 Massy Cedex, France, July, 1993.

20. Nutt, Gary J., Jacques Bicard-Mandel, Jean Faget, Beatrice Michel, and Marc Powell-Evans, “FlowPATH Planner Functional Requirements,” Bull Worldwide Information Systems, Imaging and Office Solutions, 7 rue Ampère, 91343 Massy Cedex, France, June, 1993.

21. Nutt, Gary J., Jacques Bicard-Mandel, Jean Faget, Beatrice Michel, and Marc Powell-Evans, “FlowPATH Planner Requirements,” Bull Worldwide Information Systems, Imaging and Office Solutions, 7 rue Ampère, 91343 Massy Cedex, France, June, 1993.

22. Nutt, Gary J., “Recommendation for Future Directions of FlowPATH,” Bull Worldwide Information Systems, Imaging and Office Solutions, 7 rue Ampère, 91343 Massy Cedex, France, May, 1993.

23. Nutt, Gary J. and Clarence A. Ellis, “Adding Actors and Roles to the Basic ICN Model,”

24. Bull Worldwide Information Systems, Imaging and Office Solutions, 7 rue Ampère, 91343 Massy Cedex, France, May, 1993.

25. Nutt, Gary J., “Enhancing Mobile,” Bull Worldwide Information Systems, Imaging and Office Solutions, 7 rue Ampère, 91343 Massy Cedex, France, March, 1993.

26. McWhirter, Jeffrey D. and Gary J. Nutt, “A Characterization Framework for Visual Languages,” University of Colorado Department of Computer Science Technical Report No. CU-CS-586-92, March, 1992.

27. Nutt, Gary J., “Trace Driven Petri Net Simulation,” University of Colorado Department of Computer Science Technical Report No. CU-CS-569-91, December, 1991.

28. Grunwald, Dirk, Gary Nutt, David Wagner, and Benjamin Zorn, “A Parallel Execution Evaluation Testbed,” University of Colorado Department of Computer Science Technical Report No. CU-CS-560-91, November, 1991.

29. Grunwald, Dirk, Gary Nutt, Anthony Sloane, David Wagner, William Waite, and Benjamin Zorn, “Execution Architecture Independent Program Tracing,” University of Colorado Department of Computer Science Technical Report No. CU-CS-525-91, April, 1991.

30. Nutt, Gary J., “Olympus Application Note: Capacity Planning with PN-Olympus,” University of Colorado Department of Computer Science Technical Report No. CU-CS-522-91, March, 1991.

31. Nutt, Gary J., “Collected Papers on Olympus,” University of Colorado Department of Computer Science Technical Report No. CU-CS-518-91, February, 1991.

32. Nutt, Gary J., “The Case for Open Systems,” University of Colorado Department of Computer Science Technical Report No. CU-CS-515-91, February, 1991.

33. Grunwald, Dirk, Gary Nutt, David Wagner, William Waite, and Benjamin Zorn,

34. "A Testbed for Improving the Performance of Parallel Programs and Systems," University of Colorado Department of Computer Science Technical Report No. CU-CS-512-91, January, 1991.

35. Beguelin, Adam L. and Gary J. Nutt, "Collected Papers on Phred," University of Colorado Department of Computer Science Technical Report No. CU-CS-511-91, January, 1991.

36. Demeure, Isabelle M. and Gary J. Nutt, "Collected Papers on VISA and ParaDiGM," University of Colorado Department of Computer Science Technical Report No. CU-CS-488-90, August, 1990.

37. Nutt, Gary J., "Understanding the ImageLink Platform," Interactive Systems Corporation Technical Report No. ISM-K-06-89, December, 1989 (company confidential).

38. Demeure, Isabelle M. and Gary J. Nutt, "Modeling and Prototyping Parallel Distributed Computations," University of Colorado Department of Computer Science Technical Report No. CU-CS-450-89, October, 1989.

39. Nutt, Gary J., "Open Systems," Upside Technologies Technical Report No. 102-89, May, 1989.

40. Elliott, R. Stephen and Gary J. Nutt, "Remarks on the Cost of Using A Remote Procedure Call Facility," University of Colorado Department of Computer Science Technical Report No. CU-CS-426-89, February, 1989.

41. Demeure, Isabelle M., Sharon L. Smith, and Gary J. Nutt, "Modeling Parallel, Distributed Computations Using ParaDiGM -- A Case Study: The Adaptive Global Optimization Algorithm," University of Colorado Department of Computer Science Technical Report No. CU-CS-419-87, December, 1988.

42. Nutt, Gary J., "Olympus: An Extensible Modeling and Programming System," University of Colorado Department of Computer Science Technical Report No. CU-CS-412-87, October, 1988.

43. Nutt, Gary J., "A Formal Model for Interactive Simulation Systems," University of Colorado Department of Computer Science Technical Report No. CU-CS-410-87, September, 1988.

44. Nutt, Gary J., "Kodak Computational Platforms," Interactive Systems Corp. Technical Report No. ISM-K-04-88, July, 1988 (company confidential).

45. Nutt, Gary J., "Image Storage Technology," Interactive Systems Corp. Technical Report No. ISM-K-03-88, June, 1988 (company confidential).

46. Nutt, Gary J., "Technology Issues for Kodak Products in the 1990s," Interactive Systems Corp. Technical Report No. ISM-K-02-88, June, 1988 (company confidential).

47. Nutt, Gary J., "Compatible Products," Interactive Systems Corp. Technical Report No. ISM-K-01-88, June, 1988 (company confidential).

48. Nutt, Gary J., "Designing Distributed Systems," Upside Technologies Technical Report No. 101-88, May, 1988.

49. Nutt, Gary J., "BM/C^3 Architecture Model" Ford Aerospace & Communications report, January, 1988.

50. Nutt, Gary J., A. Beguelin, I. Demeure, S. Elliott, J. McWhirter, and B. Sanders, "Olympus User's Manual, Version 0.5," University of Colorado Department of Computer Science Technical Report No. CU-CS-382-87, December, 1987 (revised June, 1989).

51. Nutt, Gary J., "A Rapid Simulation Modeling System," September, 1987.

52. Demeure, Isabelle M. and Gary J. Nutt, "Modeling Parallel Program Architectures," University of Colorado, Department of Computer Science, September, 1987.

53. Demeure, Isabelle M. and Gary J. Nutt, "Modeling Distributed Computations," University of Colorado Department of Computer Science Technical Report No. CU-CS-371-87, July, 1987.

54. Nutt, Gary J., "Bilogic Precedence Graph Models," University of Colorado Department of Computer Science Technical Report No. CU-CS-363-87, May, 1987.

55. Nutt, Gary J, "Compound Documents and Windows," Interactive Systems Corporation, Applications Product Group, January, 1986.

56. Nutt, Gary J., "Notes on Local Area Networks," NBI Research Department Memo No. SE-11, January, 1982.

57. Nutt, Gary J., "A Network Architecture Recommendation," NBI Research Department Memo No. SE-5, July, 1981, (company confidential).

58. Nutt, Gary J. and Robert F. Rosin, "Business Communication Services for the 1980s", Bell Laboratories, Memo TM Nos. 81-43242-1 and 81-11353, February, 1981, (company confidential).

59. Nutt, Gary J. and Bruce W. Sanders, "An Experimental Internetwork Architecture," Bell Laboratories, Memo No. 43242-810220.01MF, February, 1981, (company confidential).

60. Nutt, Gary J., "Simulating Voice Loads on an Ethernet-like Network," Bell Laboratories, Memo No. 3243-800912.01MF, September, 1980, (company confidential).

61. Nutt, Gary J. and Clarence A. Ellis, "On the Equivalence of Office Models," Xerox PARC, SSL Publication No. SSL-79-8, 1979.

62. Nutt, Gary J., “A Survey of Remote Monitoring,” U. S. National Bureau of Standards, Special Publication No. 500-42, January, 1978.

63. Nutt, Gary J., B. W. Sanders and K. S. Smith, “Evaluation During Machine Design: A Case Study,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-125-77, October, 1977.

64. Nutt, Gary J., “Integrating Architecture and Operating Systems,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-104-77, March, 1977.

65. Nutt, Gary J., “Notes on a MAP Microprocessor Implementation,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-102-77, January, 1977.

66. Nutt, Gary J., “Measuring User Programs for a SIMD Processor,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-089-76, April, 1976.

67. Nutt, Gary J., “The Architecture of a Multi Associative Processor,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-070-75, June, 1974 (revised August, 1978).

68. Nutt, Gary J., “Sample Programs for a Hypothetical Computer,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-058-74, October, 1974.

69. Nutt, Gary J., “Evaluation Net Simulation System Reference Manual,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-042-74, April, 1974.

70. Nutt, Gary J. and Clarence A. Ellis, “Preliminary Thoughts on Degrees of Protection in Multiprocessor Systems,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-036-74, January, 1974.

71. Nutt, Gary J., “Some Applications of Finite State Automata Theory to the Deadlock Problem,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-017-73, April, 1973.

72. Nutt, Gary J., “Some Applications of Finite State Automata Theory to the Deadlock Problem,” University of Colorado, Department of Computer Science, Technical Report No. CU-CS-017-73, April, 1973.

73. Noe, J. D., and Gary J. Nutt, “Multiprogramming and Multiprocessing System Descriptions,” University of Washington, Computer Science Group, Technical Report No. 72-01-08, August, 1972.

74. Noe, J. D., and Gary J. Nutt, “Some Evaluation Net Macro Structures,” University of Washington, Computer Science Group, Technical Report No. 72-01-07, August, 1972.

75. Nutt, Gary J., “SIM6000 Program Description and Use,” University of Washington, Computer Science Group, Technical Report No. 71-07-05, July, 1971.

76. Nutt, Gary J., “The Boeing Dayfile Analysis Modified for the University of Washington Computer Center,” University of Washington, Computer Center Technical Report No. 4, October, 1970.

Newsletter Columns

1. Nutt, Gary, “Introduction to the Shell,” *Patricia Seybold’s UNIX in the Office,* Vol. 3, No. 6 (June 1988), pp. 21-23.

2. Nutt, Gary, “Network Protocols: Part 2,” *Patricia Seybold’s UNIX in the Office*, Vol. 3, No. 2 (February 1988), pp. 25-26.

3. Nutt, Gary, “Network Protocols: Part 1,” *Patricia Seybold’s UNIX in the Office*, Vol. 2, No. 12 (December 1987), pp. 20-21.

4. Nutt, Gary, “System V.3 Streams,” *Patricia Seybold’s UNIX in the Office*, Vol. 2, No. 10 (October 1987), pp. 28-29.

5. Nutt, Gary, “Unix Device Support,” *Patricia Seybold’s UNIX in the Office*, Vol. 2, No. 8 (August 1987), pp. 16-17.

6. Nutt, Gary, “Uniting Unix,” *Patricia Seybold’s UNIX in the Office*, Vol. 2, No. 7 (July 1987), pp. 12-13.

7. Nutt, Gary, “Systems Administration Expert Programs,” *Patricia Seybold’s UNIX in the Office*, Vol. 2, No. 4 (April 1987), pp. 21-22.

8. Nutt, Gary, “Administering Unix Systems,” *Patricia Seybold’s UNIX in the Office,* Vol. 2, No. 3 (March 1987), pp. 18-19.