

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DROPBOX, INC. AND MICROSOFT CORPORATION,
Petitioners,

v.

DATANET, LLC,
Patent Owner.

U.S. Patent No.: 9,218,348
Issued: December 22, 2015
Application No.: 13/925,768
Filed: June 24, 2013

Title: Automatic Real-Time File Management Method And Apparatus

DECLARATION OF DAVID MAIER

TABLE OF CONTENTS

	Page(s)
I. INTRODUCTION AND ENGAGEMENT	6
II. BACKGROUND AND QUALIFICATIONS	6
III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '348 PATENT.....	10
IV. UNDERSTANDING OF PATENT LAW	12
V. PERSON OF ORDINARY SKILL IN THE ART	15
VI. CLAIM CONSTRUCTION	17
VII. STATE OF THE ART	17
A. File-Based Backup Systems Were Well-Known	17
B. From The Perspective Of The Backup–Restore Application, There Was No Distinction Between Local and Remote File Storage	18
VIII. GROUND 1: CLAIMS 1-6, 8-20, 23-31 ARE OBVIOUS OVER KENLEY (EX1014) AND LONG (EX1004)	19
A. Kenley (EX1014)	19
B. Long (EX1004).....	20
C. Combining The Teachings Of Kenley And Long.....	25
1. Claim 1	32
2. Claim 2	54
3. Claim 3	55
4. Claim 4.....	57
5. Claim 5	58
6. Claim 6.....	61

7.	Claim 8.....	62
8.	Claims 9-11	64
9.	Claims 12-15	64
	a) Element [12.6]	67
	b) Element [13.6]	68
	c) Element [14.6]	69
	d) Element [15.6]	70
10.	Claim 16.....	71
11.	Claim 17.....	71
12.	Claim 18.....	72
13.	Claim 19.....	72
14.	Claim 20.....	73
15.	Claim 23.....	73
16.	Claim 24.....	74
17.	Claim 25.....	74
18.	Claim 26.....	76
19.	Claim 27.....	76
20.	Claim 28.....	77
21.	Claim 29.....	77
22.	Claim 30.....	77
23.	Claim 31	78

IX. GROUND 2: CLAIMS 1-6, 8-20, AND
23-31 ARE OBVIOUS OVER CABRERA (EX1005),
SCHNEIDER (EX1020), AND WHITING (EX1006) 78

A. Cabrera (EX1005) 78

B. Schneider (EX1020).....82

C. Whiting (EX1006).....83

D. Combining Cabrera, Schneider, And Whiting 85

1. Claim 1 89

2. Claim 2 98

3. Claims 3-5 99

4. Claim 6 100

5. Claim 8 101

6. Claims 9-11 102

7. Claims 12-15 102

a) Element [12.6] 103

b) Element [13.6] 103

c) Element [14.6] 103

d) Element [15.6] 104

8. Claim 16 104

9. Claim 17 104

10. Claim 18 105

11. Claim 19 105

12. Claim 20 105

13. Claim 23 106

14.	Claim 24	106
15.	Claim 25	107
16.	Claim 26	108
17.	Claim 27	108
18.	Claim 28	109
19.	Claim 29	109
20.	Claim 30	110
21.	Claim 31	110

I, David Maier, do hereby declare as follows:

I. INTRODUCTION AND ENGAGEMENT

1. I have been retained as an independent expert on behalf of Dropbox, Inc. and Microsoft Corporation in connection with the above-captioned Petition for *Inter Partes* Review (“IPR”) to provide my analyses and opinions on certain technical issues related to U.S. Patent No. 9,218,348 (hereinafter “the ’348 patent”).

2. I am being compensated at my usual and customary hourly rate for the time spent in connection with this IPR. My compensation is not affected by the outcome of this IPR.

3. Specifically, I have been asked to provide my opinions regarding whether claims 1-6, 8-20, and 23-31 (each a “Challenged Claim” and collectively the “Challenged Claims”) of the ’348 patent would have been obvious to a person having ordinary skill in the art (“POSITA”) as of September 21, 2000. It is my opinion that each Challenged Claim would have been obvious to a POSITA after reviewing the prior art discussed herein.

II. BACKGROUND AND QUALIFICATIONS

4. I am an expert in the fields of database systems and information technology. More specifically, I have expertise in database management systems, database applications, and distributed data processing. In formulating my opinions, I have relied upon my training, knowledge, and experience in the relevant art. A

copy of my curriculum vitae is appended to this declaration as Appendix A and provides a description of my professional experience, including my academic and employment history, publications, conference participation, awards and honors, and more. The following is a brief summary of my relevant qualifications and professional experience.

5. I graduated from the University of Oregon Honors College with a B.A. in Mathematics and Computer Science (double major) in 1974. I earned my Ph.D. in Electrical Engineering and Computer Science from Princeton University in 1978.

6. I am currently Maseeh Professor Emeritus of Emerging Technologies in the Department of Computer Science at Portland State University, where I have been since 2004.

7. I have more than forty years' technical experience in the computer science field, including extensive experience with database systems dating back to 1977. I worked on the development of the GemStone and O2 object-oriented database systems in the 1983-90 timeframe, and the StreamInsight data-stream engine (Microsoft), and have served as a technical consultant on database systems for a range of companies and federal laboratories from 1979 until 2018, including Mentor Graphics, IBM, Microsoft, and Informix. I am also an inventor on five patents related to database technologies, which issued between 2013 and 2018. While I am serving as an independent expert for attorneys representing Microsoft in

relation to U.S. Patent Nos. 10,585,850; 9,218,348; and 8,473,478 (one of which is the subject of this declaration), I am not engaged in any technical consulting work for Microsoft at present.

8. My research is in the field of computer science, primarily those fields related to database systems.

9. I have taught a range of courses related to database systems at Portland State and elsewhere, including Introduction to Database Management Systems, Cloud and Cluster Data Management, Principles of Database Systems, Object-Oriented Database Systems, Data Stream Management and Information Retrieval on the Internet. I have also supervised 26 Ph.D. students and 14 Master's projects and theses. My research has been supported by a variety of federal, state and private sources.

10. During the 40-plus years since receiving my Ph.D., I have held multiple positions as a professor and researcher in the field of computer science. I have held faculty positions in Computer Science at the State University of New York at Stony Brook, the Oregon Graduate Institute, Oregon Health and Science University, and, currently, Portland State University. I have had sabbatical or visiting appointments at INRIA (French National Informatics Laboratory), University of Wisconsin, Microsoft Research, and the National University of Singapore. I have also been a participant in the Intel Science and Technology Center for Big Data.

11. My research work has been recognized by multiple organizations. I received one of the first NSF Presidential Young Investigator Awards in 1984 (precursor of the NSF CAREER awards), the SIGMOD Innovations Award in 1997, an ACM Fellow in 1997 and the Microsoft Outstanding Contributor Award in 2016. At the university level, I have received multiple awards for teaching and research, including the Branford Price Millar award at Portland State, which recognizes excellence in research and teaching.

12. I am a Fellow in the Association for Computing Machinery (ACM), and a member of the ACM Special Interest Group on the Management of Data (SIGMOD). I was General Co-Chair for the SIGMOD 2020 conference. I am a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), including the IEEE Computer Society and the IEEE Technical Committee on Data Engineering. I am also a member of the Society for Industrial and Applied Mathematics (SIAM). I have served on the Board for Mathematical Sciences and Analytics (BMSA) of the National Academies of Science, Engineering and Medicine, and have been a member of two expert study panels convened by BMSA.

13. In total, I have been an author on more than 245 scholarly publications, dating back to 1973. These publications include articles in peer-reviewed journals, papers in selective conferences, books, book chapters and reports resulting from agenda-setting panels. As of last count, these publications involve 400+ distinct co-

authors, including faculty from Stanford, MIT, Wisconsin, UC Berkeley, Washington, Brown, Edinburgh, UPenn, Univ. of Virginia, Michigan, CMU, Stony Brook, Illinois and National University of Singapore as well as researchers from IBM, AT&T Labs and Microsoft.

14. I am still active in multiple research projects at Portland State University, most recently around the topics of Urban Informatics, Data Productivity, and Computational Photography. I also have current collaborations with faculty and students from Oregon State University on Data Integration and with Meta on Managing Data Infrastructure.

III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '348 PATENT

15. In preparing this declaration, I have reviewed the following materials bearing Exhibit Nos. that I understand are being referenced in the IPR which my declaration accompanies:

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 9,218,348 (“ 348 patent ”)
1002	File History of U.S. Patent No. 9,218,348 (“ 348 FH ”)
1004	U.S. Patent No. 5,325,519 (“ Long ”)
1005	U.S. Patent No. 5,953,729 (“ Cabrera ”)
1006	U.S. Patent No. 5,778,395 (“ Whiting ”)

No.	Description
1007	Marc Farley, <i>Building Storage Networks</i> (2000) (excerpts) (“ Farley ”)
1008	Helen Custer, <i>Inside the Windows NT File System</i> (1994) (excerpts) (“ Custer ”)
1009	Newton’s Telecom Dictionary (2000) (excerpts)
1010	U.S. Patent No. 5,860,124 (“ Matthews ”)
1011	Ann L. Chervenak, et al., <i>Protecting File Systems: A Survey of Backup Techniques</i> (1998)
1012	U.S. Patent No. 5,485,606 (“ Midgley ”)
1013	U.S. Patent No. 5,745,313 (“ Sliger ”)
1014	U.S. Patent No. 5,276,867 (“ Kenley ”)
1015	Abraham Silberschatz, et al., <i>Operating System Concepts</i> (3d. ed. 1991) (“ Silberschatz ”)
1016	U.S. Patent No. 6,016,553 (“ Schneider-US ”)
1017	Joost S.M. Verhofstad, <i>Recovery Techniques for Database Systems</i> , Computing Surveys (June 1978)
1018	U.S. Patent No. 5,771,354 (“ Crawford ”)
1019	Microsoft Press Computer Dictionary (1994) (excerpts)
1020	PCT Publication No. WO 99/12101 (“ Schneider ”)
1021	U.S. Patent No. 5,684,991 (“ Malcolm ”)
1022	Russell Sandberg, et al., <i>Design and Implementation of the Sun Network Filesystem</i> , in <i>Innovations in Internetworking</i> (Craig Partridge ed., Artech House 1988) (excerpts) (“ Sandberg ”)

IV. UNDERSTANDING OF PATENT LAW

16. I am not an attorney. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to my opinions. My understanding of the law was provided to me by the Petitioner's attorneys.

17. I understand that when considering the scope of the claims of a patent that the patent claim terms should generally be given the ordinary meaning that the terms would have to a person of ordinary skill in the art in question after reading the patent as of the earliest claimed priority date.

18. I understand that the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the term appears, but in the context of the entire patent, including the specification. I further understand that the principal considerations regarding the scope and meaning of the claims are the plain language of the claim (including the surrounding claim language and context), the patent specification, and the prosecution history. I understand that while a claim is to be read in light of the specification, one must generally avoid importing limitations into the claim from the specification. I am also informed that the prosecution history can often inform the meaning of the claim by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it

would otherwise be. I applied these understandings when considering the scope and meaning of the claims of the '348 patent.

19. I understand that a prior art reference anticipates an asserted claim, and thus renders the claim unpatentable, if all elements of the claim are disclosed in that prior art reference, either explicitly or inherently (i.e., necessarily present or implied).

20. I further understand that a claim is unpatentable if it would have been obvious. Obviousness of a claim requires that the claim would have been obvious from the perspective of a POSITA at the time the alleged invention was made. I understand that a claim could have been obvious from a single prior art reference or from a combination of two or more prior art references.

21. I understand that an obviousness analysis requires an understanding of the scope and content of the prior art, any differences between the alleged invention and the prior art, and the level of ordinary skill in evaluating the pertinent art.

22. I further understand that a claim would have been obvious if it unites old elements with no change to their respective functions, or merely substitutes one element for another known in the field, and that combination yields predictable results. While it may be helpful to identify a reason for this combination, I understand that there is no strict requirement of finding an express teaching, suggestion, or motivation to combine within the references. When a product is

available, design incentives and other market forces can prompt variations of it, either in the same field or different one. If a POSITA can implement a predictable variation, obviousness likely bars its patentability. For the same reason, if a technique has been used to improve one device and a POSITA would recognize that it would improve similar devices in the same way, using the technique would have been obvious. I understand that a claim would have been obvious if common sense directs one to combine multiple prior art references or add missing features to reproduce the alleged invention recited in the claims.

23. I further understand that certain factors may support or rebut the obviousness of a claim. I understand that such secondary considerations include, among other things, commercial success of the patented invention, skepticism of those having ordinary skill in the art at the time of invention, unexpected results of the invention, any long-felt but unsolved need in the art that was satisfied by the alleged invention, the failure of others to make the alleged invention, praise of the alleged invention by those having ordinary skill in the art, and copying of the alleged invention by others in the field. I understand that there must be a nexus—a connection—between any such secondary considerations and the alleged invention. I also understand that contemporaneous and independent invention by others is a secondary consideration tending to show obviousness.

24. I am not aware of any allegations by the named inventors of the '348 patent or any assignee of the '348 patent that any secondary considerations tend to rebut the obviousness of any Challenged Claim of the '348 patent.

25. I understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the descriptions found in the patent being considered.

26. I understand that other challenges to the patentability of a patent, including patent ineligibility, enablement, written description, and definiteness or clarity of claim language, cannot be raised in IPR proceedings before the Board.

27. I understand that Petitioner has the burden of proving unpatentability by a preponderance of evidence, which means that the claims are more likely than not unpatentable.

28. The analysis in this declaration is in accordance with the above-stated legal principles.

V. PERSON OF ORDINARY SKILL IN THE ART

29. For purposes of this declaration, I understand the time of the invention to be September 2000, which is the filing date of the provisional application to which the '348 patent purportedly claims priority (provisional application no. 60/234,221). I am not offering an opinion on the propriety of this priority date at this time.

30. I understand that several factors are relevant to determining the level of ordinary skill in the art at the time of the claimed invention. I understand that those factors include: (1) the educational level of the inventor(s); (2) the type of problems encountered in the art; (3) prior art solutions to those problems; (4) the rapidity with which innovations are made; (5) sophistication of the technology; and (6) the educational level of active workers in the field.

31. Having reviewed and considered the '348 patent, its prosecution history, the state of the art at the time of the invention, including the level of skill of the persons with whom I worked, and the above-listed factors, it is my opinion that the person of ordinary skill in the art for purposes of the '348 patent in September 2000 ("POSITA") would have held a bachelor's degree in computer science, electrical engineering, computer engineering, or a closely related field, and one or more years of experience in the design and development of file management, restoration, and archival systems. More education, e.g., a Master's or Ph.D. could compensate for a lack of work experience and vice versa. There would be no meaningful distinction if the critical date were 1999 or 2001, as far as my analysis is concerned. The technology overview provided below (*infra* Section VII) reflects some of the information within a POSITA's knowledge.

32. I have used this level of ordinary skill in the art in forming my opinions as set forth in this declaration. My opinions are from the perspective of this level of

ordinary skill in the art. I met or exceeded this level of skill in the art in September 2000.

VI. CLAIM CONSTRUCTION

33. I understand that, in IPR, claim terms should be given the ordinary meaning that the terms would have to a person of ordinary skill in the art on the earliest effective filing date, in view of the specification and file history.

34. For this declaration, I apply the plain and ordinary meaning of all claim terms and do not rely on any specific constructions. I reserve the right to offer my opinions on any constructions or construction issues presented by Patent Owner Datanet LLC.

VII. STATE OF THE ART

A. File-Based Backup Systems Were Well-Known

35. In 1978, Verhofstad wrote that “[r]ecover techniques can be used to restore data in a system to a usable state. Such techniques are widely used in filing systems and database systems in order to cope with failures.” EX1017, i; *see also* EX1011, 19 (“Backup software can operate either on files or on physical disk blocks.”).

36. Cabrera, with an application filed in 1997, explained that backup/archival systems “typically copy data from a local storage medium to a backup or archival storage medium, sometimes located at a remote location.”

EX1005, 1:35-42. This “is not an instantaneous process,” so a conventional solution that “allow[s] access to files while archive or backup copies are created is to duplicate the information that will be backed up or archived and ‘stage’ the information in a temporary storage area. The information may then be copied from the staging area and sent to backup or archive storage.” *Id.*, 1:43-45, 2:5-10.

37. Midgley explained that “[i]nformation processed by a computer is organized into distinct units called files, each of which contains related data.” EX1012, 1:26-28. When an application or an operating system alters data on a storage device (e.g., hard disk), “[t]he computer’s operating system is responsible for communicating with the mass storage device in order to create, delete, modify, or append files stored within the mass storage device. Data is stored on the disk by writing to a file within the mass storage device.” EX1010, 1:13-19; *see also* EX1015, 370 (“The operating system provides system calls to create, write, read, reset, and delete files.”); EX1016, 1:35-41 (“The applications that run on computers typically operate under an operating system (OS) that has the responsibility, among other things, to save and recall information from a hard disk. The information is typically organized in files.”); EX1004, 4:23-25, 1:19-25; EX1013, 5:18-47.

**B. From The Perspective Of The Backup–Restore Application,
There Was No Distinction Between Local and Remote File Storage**

38. By the 1990s, Network File Systems (such as NFS, first released by Sun Microsystems in 1984) erased any difference between how users accessed local

and remote files. EX1022, 370-390 (showing first release); EX1018, 3:64-66 (“Drives on a remote system defined using NFS ... are available for mounting by other remote systems on the Internet.”). This technology meant that applications on the user computer did not need to know the actual storage location of the file, a property known as “location transparency.” Therefore, with respect to backup applications, there was no significant technical barrier between storing a file locally and storing it remotely.

39. By 1997, then-state-of-the-art backup and archival “systems typically cop[ied] data from a local storage medium to a backup or archival storage medium, sometimes located at a remote location. The process of transferring data from a local storage medium to a backup or remote storage medium is much the same in either case.” EX1005, 1:38-42; *id.*, 3:3-7. “[C]ommonplace” remote networking systems utilized a wide area network (WAN) and “the computer 20 typically includes a modem 54 or other means for establishing communications over the wide area network 52, such as the Internet.” *Id.*, 6:8-28.

**VIII. GROUND 1: CLAIMS 1-6, 8-20, 23-31 ARE
OBVIOUS OVER KENLEY (EX1014) AND LONG (EX1004)**

A. Kenley (EX1014)

40. U.S. Patent No. 5,276,867 (EX1014, “Kenley”) issued January 4, 1994, from an application filed December 19, 1989, and I understand that is prior art to the ’348 patent. Kenley is analogous art to the ’348 patent because it is from the

same field of endeavor. The '348 patent describes its "Field Of The Invention" as "file preservation, file capture, file management and file integrity techniques for computing devices." EX1001, 1:10-14. Similarly, Kenley relates to "digital data processing" and "improved data migration." EX1014, 1:13-15. Thus, the '348 patent and Kenley are both within the same field of endeavor of data management and preservation.

41. Kenley describes a three-tier "hierarchical storage configuration" that stores data organized as files in three storage elements of varying capacity and access speed: a faster, lower-capacity primary storage; a slower, higher-capacity secondary storage; and an even slower and higher-capacity backing store. EX1014, 2:44-55; *id.*, 4:54-66. For example, "[a] file storage element in the secondary storage maintains data in the form of files." *Id.*, 3:1-2, 3:29-30 ("the backing store stores data organized as files"). Kenley teaches migrating data from secondary storage to the backing store, either at selected times or upon receiving a secondary storage full message. *Id.*, 2:56-68. Kenley further teaches a relational database with a "system catalog" for maintaining and providing transparent access to file attribute and location data. *Id.*, 3:17-28; *id.*, 6:36-40.

B. Long (EX1004)

42. U.S. Patent No. 5,325,519 (EX1004, "Long") issued June 28, 1994, from an application filed October 18, 1991, and I understand that qualifies as prior

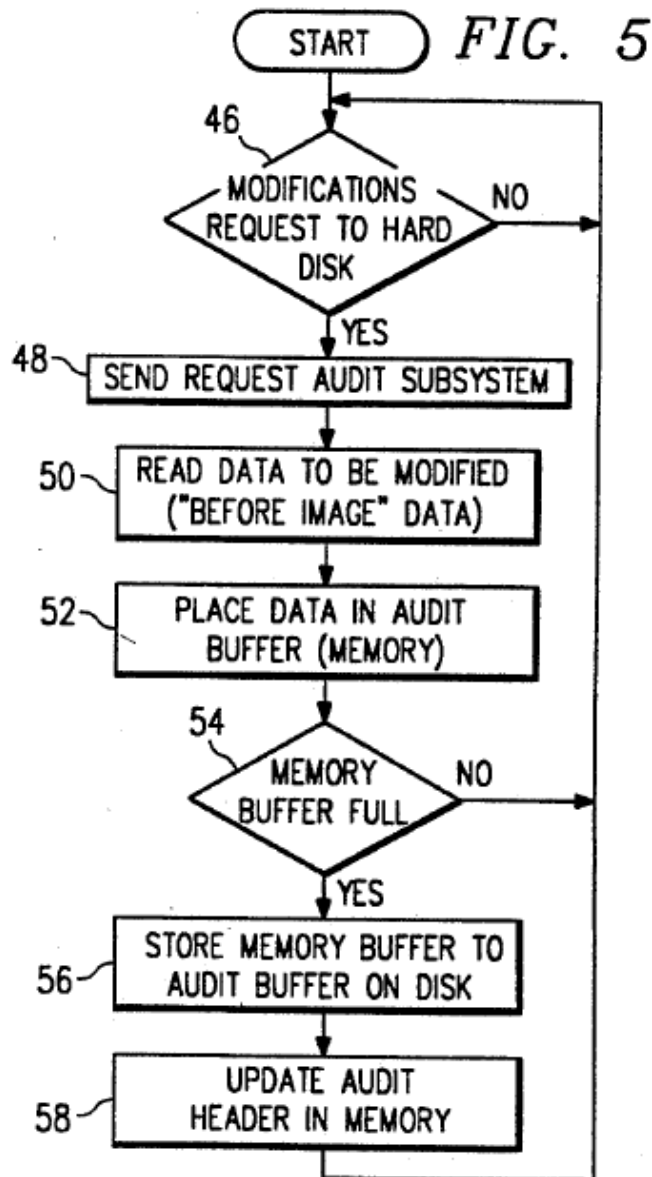
art to the '348 patent. Long is analogous art to the '348 patent because it is from the same field of endeavor. Similar to how the '348 patent describes its “Field Of The Invention” as “file preservation, file capture, file management and file integrity techniques for computing devices” (EX1001, 1:10-14), Long describes its “Technical Field” as relating to a “fault tolerant computer,” which Long explains has “an effective and cost[-]efficient method of recovering from an error affecting the hard disk drive.” EX1004, 1:5-7, 1:56-59. Thus, the '348 patent and Long are both within the same field of endeavor of data management and preservation.

43. Long discloses a general-purpose backup technique that involves creating a backup copy of data prior to modification of that data, as well as archiving that data to separate storage. Long teaches that its technique is broadly “[a]pplicable to any mass storage media wherein old data may be overwritten with new data.” EX1004, 9:54-56.

44. Long’s background notes the increasing importance of data stored on personal computers and the issues caused by faulty data, including the impact on “critical PC applications.” *Id.*, 1:10-17. Long explains how a computer’s application programs or operating system frequently change “the data on the hard disk” and that system events can lead to “valuable information being lost or corrupted.” *Id.*, 4:23-27. Long thus emphasizes the importance of the “ability to return to a previous known state after a failure.” *Id.*, 4:21-23.

45. To preserve historic versions of data, Long teaches creating copies prior to any changes or modifications to data. Creation of these backup copies is triggered, for example, by an application issuing a “modification request” ... to modify the contents of the hard disk.” *Id.*, 7:28-29; *see also id.*, 7:33-34 (backup occurs when the “computer systems [] needs to write to a hard disk”). Then, “before the data on the disk is modified” by the system, “the request is sent to the audit subsystem,” which is Long’s name for the process that creates the copies of the data. *Id.*, 7:33-37.

46. After a “modification request” generated by “an application to modify the contents of the hard disk,” a backup copy is first stored in a “memory buffer” in faster semiconductor memory. *Id.*, 7:28-29, 7:43-53. From the memory buffer, the data is then copied to an “audit buffer” on the local disk. *E.g., id.*, 7:38-43 (describing how data is “placed in the audit buffer as ‘before image’ data”). Long also maintains an audit header that “contains information that is used to manage the audit buffer 30, archive media ... and the general environment of the audit subsystem.” *Id.*, 4:42-47. Thus, “when the data on the hard disk(s) is changed,” Long creates a backup that “contains the data that existed on one sector of the disk prior to the write occurring.” *Id.*, 6:46-48. Long’s process of first storing to the higher-speed semiconductor “memory buffer” and then to the computer’s hard disk “audit buffer” is reflected in Long Fig. 5:



Id., FIG. 5.

47. Long further teaches an archive process that copies the data from the local-disk audit buffer to a separate archive media on “external storage.” *E.g.*, *id.*, 8:36-9:9. The “[a]rchiving of the audit buffer 30 occurs at predetermined intervals,” wherein the data is transferred from the audit buffer to an archive media. *Id.*, 8:36-42. Long describes an archive method that “allows the archiving to take place in the

background so that the user may continue processing during an archive operation.” *Id.*, 8:42-45. Long explains that the archive media is “external storage” on a “removable media device,” such as a floppy disk. *Id.*, 7:12-15, 8:32-33; *id.*, 2:48-49. This archive process, which works together with the audit buffer backup process, is reflected in Long Fig. 6:

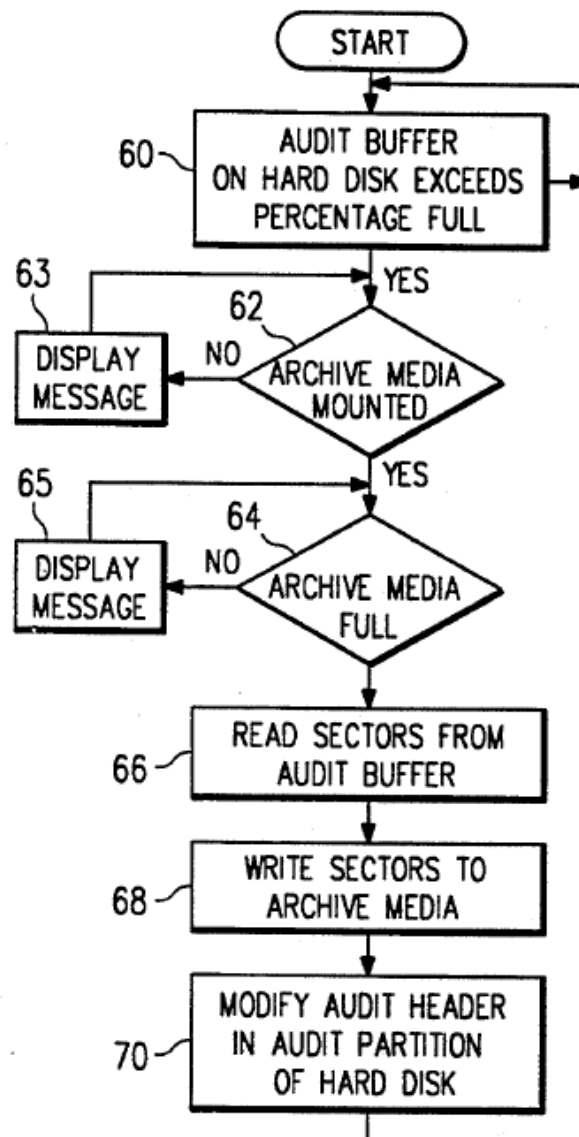


FIG. 6

Id., FIG. 6.

48. Long discloses modifying the audit header to reflect transfer of the data to the archive media. *Id.*, 9:7-9. In this way, both “the audit partition 26 and any archive media allow rollback of the state of the hard disk to a previous known state” upon failure. *Id.*, 9:10-13.

C. Combining The Teachings Of Kenley And Long

49. As discussed below and herein, there was a motivation to combine Kenley and Long, as well as a reasonable expectation of success in doing so. Specifically, a POSITA would have been motivated to use the modification-driven archival approach taught by Long for a file-based backup system as taught by Kenley because backing up files when they are modified would improve the reliability of the file-based backup system.

50. For example, a POSITA looking to implement the “incremental backup” technique of Kenley would have recognized that the incremental backup is only “[a] copy of ... specified files of data that has been modified since a specified date, usually the date of the last full or incremental backup.” EX1014, 7:19-30 (discussing use of the incremental backup), 5:3-6 (defining incremental backup). Kenley explains that the incremental backup process “greatly reduc[es] the amount of data that must be copied” which “vastly reduce[s] the amount of time required to back up” files. EX1014, 12:66-13:17. Long, in turn, notes the data risks posed by

power outages and operator error (EX1004, 1:46-55), and provides “significant improvements in hard disk error recovery” (*id.*, 1:62-65). These improvements include increased reliability by creating backup data after each modification, thus avoiding the potential loss of data created in between the timed backups of Kenley and other incremental backup systems. *Id.*, 4:23-27. The POSITA would have been motivated to incorporate Long’s technique of backing up data upon modification into a file-based backup system such as Kenley because the Long technique further increases the reliability of file backup, as there is no delay between modification and backup. This combination serves the goal of “reduc[ing] the time associated with conventional data migration schemes” taught by Kenley, which is consistent with the goal of providing an “efficient, reliable and cost[-]effective architecture for disk fault tolerance” taught by Long. EX1014, 2:26-30; EX1004, 2:13-14.

51. For these and the following additional reasons, a POSITA would have been motivated with a reasonable expectation of success to implement a file-based system such as that described by Kenley with Long’s backup and archival techniques.

52. First, Long teaches a “Fault Tolerant Computer With Archival Rollback Capabilities” that “includes significant improvements in hard disk error recovery” and “provides a[n] efficient, reliable and cost[-]effective architecture for disk fault tolerance.” EX1004, 1:2-3, 1:62-64, 2:13-14. A POSITA would have thus been

motivated to improve Kenley's file-based "hierarchical digital data storage system" (EX1014, 2:33-36, 3:1-47) by adding Long's event-driven backup process that efficiently saved backups prior to any modifications of a file stored on hard disk.

53. Second, a POSITA would have known that Long is applicable to files. While Long broadly discusses "data on a hard disk" and the copying of such data prior to modifying "the contents of the hard disk," Long does not specifically consider data stored in files. EX1004, 1:46-48, 4:23-25, 7:28-29. However, a POSITA reading Long, with its discussion of applications modifying data, would immediately contemplate use of the Long approach within file-based systems, for preserving data stored in files within that system. The POSITA knew that data on a hard disk was most commonly organized as files to facilitate data access using a file system. *E.g.*, EX1013, 5:18-51 (describing the File Allocation Table (FAT) file system created for the MS-DOS operating system); EX1012, 1:27-60 (explaining different file systems for different operating systems such as MS-DOS, Macintosh, and UNIX); *supra* Section VII. Indeed, the background section of one background reference flatly states what was well-known: "[i]nformation processed by a computer is organized into distinct units called files, each of which contains related data." EX1012, 1:26-27. Moreover, it was well-known that applications in file-based systems would perform write and other operations that modified the file. *E.g.*, EX1010, 1:40-43. As Kenley explains, "stor[ing] data organized as files" and using

file attributes to track the backup location of files enables the data to be more easily and transparently recovered from archive storage media for use by an application. EX1014, 3:29-32, 7:19-62 (explaining how the file-based approach enables the system to “properly backup and restore files”).

54. Third, the two references are complementary, as each teaches at least three tiers of storage—primary storage (e.g., RAM or semiconductor memory buffer), secondary storage (e.g., hard disk), and archive or backing storage (e.g., removable or external storage devices)—for use in working with and backing up data. EX1014, 2:44-55 (describing three tiers of storage); EX1004, 7:43-53 (describing first copying data to a semiconductor memory buffer and then writing the data to audit buffer on disk), 8:29-9:9 (describing moving the data from the audit buffer on disk to archive media). Kenley also teaches “‘transparent’ access to digital data” stored in archive storage media, with “access that entails no significant delay in ongoing processes” and “a storage system interface that communicates with applications programs in a manner functionally identical to that of the existing system interface.” EX1014, 2:33-43. This “transparent” archiving aligns with Long’s teachings on a system that “allows the archiving to take place in the background so that the user may continue processing during an archive operation.” EX1004, 8:42-49.

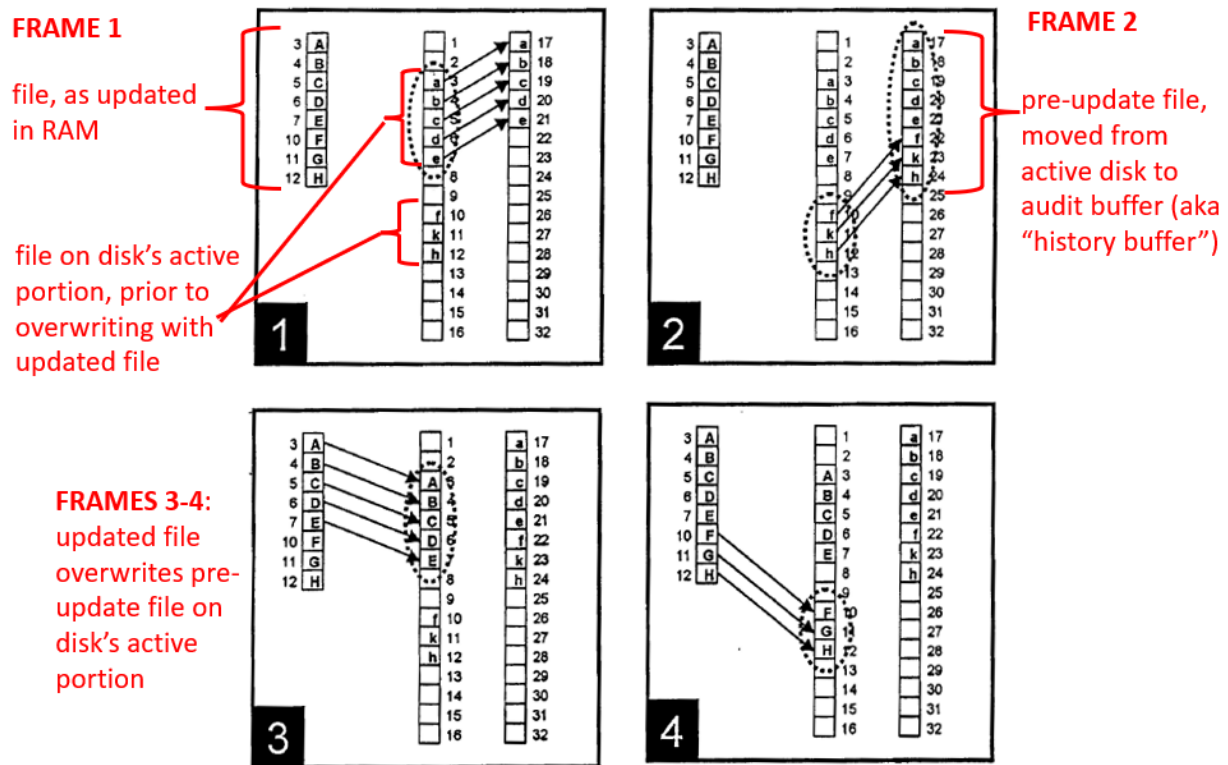
55. Fourth, Kenley's teaching of a database-held system catalog that "stor[es] and provid[es] transparent access to catalogued file attribute data" (EX1014, 3:17-24) dovetails with Long's teachings on an audit header for similarly maintaining data on the current location of backup copies of data (e.g., EX1004, 7:47-53). Kenley's system catalog "enhances transparent access to file attribute data, independent of whether the file is stored in the secondary storage element or has been previously migrated from the secondary storage element" to the backing store. EX1014, 3:25-28. Thus, a POSITA would naturally augment the catalog database of Kenley to store relevant information from the audit header taught by Long, thus "enhanc[ing] transparent access." *Id.*; see EX1004, 7:47-53, 9:7-9 (describing updating the audit header to reflect changes after migrating data); EX1014, 6:35-40 (describing the system catalog as a part of a relational database), 8:26-29 (describing updating the database with a catalog of all files backed up), 10:43-52 (describing using the backup catalogs in the database to locate backup files).

56. Fifth, Long teaches a straightforward modification-driven archival approach that applied to known multi-tier backup–archival systems. EX1004, 11:54-56 ("The present invention is applicable to any mass storage media wherein old data may be overwritten with new data."). A POSITA would have easily applied that approach in a file-based system such as Kenley's. As explained in mapping the claims below, this method uses the same event-driven approach as the Challenged

Claims. *Infra* Sections VIII.C.1-VIII.C.23. Thus, a POSITA implementing a file-based backup system such as that taught by Kenley, adapted with Long's archival method, had a reasonable expectation of success in implementing the methods of the Challenged Claims.

57. The reasonable expectation of success, and overall obviousness, of implementing Long's technique for a file-based system such as Kenley's is confirmed by another prior-art patent, U.S. 6,016,553 (EX1016; "Schneider-US") that expressly describes use of Long's audit buffer technique for saving a file's data before the file is altered. EX1016, 46:36-47:8 (using file overwrite operation to explain "what normally happens when data is written" and then describing how the Long technique (referred to as the "Move" method) saves a file just prior to it being overwritten); *see also id.*, 46:25-26 (discussing "fundamental methods for saving the prior states of overwritten data"); *id.*, 6:33-36 (presenting "fundamental methods of recording the original state of information prior to its being altered"). In reference to FIG. 38 and related figures that illustrate the "Move" method, Schneider-US explains that "[t]he column on the left represents a file" (*id.*, 46:39-40), the column in the middle "represents the OS-visible area" of the hard disk, and the column on the right "is a history buffer visible only to the engine" (*id.*, 46:63-47:1). Schneider-US then explains how, before the file overwrite is carried out on the hard disk (middle column), the system copies the file's contents into the history buffer. *Id.*,

47:1-9. Schneider-US assumes a POSITA would understand Long's method can be readily applied to file data, because Schneider-US does not indicate any changes to Long in order to be able to do so.



Id., FIG. 38 (annotated based on description at 46:36-47:9).

58. Thus, it would have been obvious to implement a file-based system as taught by Kenley using Long's data backup and archiving method. So implemented, the resulting Kenley-Long system would, based on computer-executable code within the system:

- Generate a modification request, as taught by Long, any time the operating system or a user application "need[ed] to modify the contents of" a file stored

on the hard disk. For example, a text-editing program issuing a write operation for a file would be received by the computer system and then “sent to the audit subsystem.” EX1004, 7:27-53.

- Carry out the audit and archive technique taught by Long, on the file being modified, including the copying of files first to the memory buffer, second to the audit buffer portion of the hard drive, and then to external storage, as taught by Long, thus creating “archive copies” of files as taught by Kenley. EX1014, Abstract.
- Maintain relevant audit header information, as taught by Long, in a backup catalog in a relational database, as taught by Kenley. *Supra* ¶55.

1. Claim 1

[1.1]¹ A method for archiving files, comprising steps of (a) to (i) following:²

59. To the extent the preamble is limiting, Kenley-Long satisfies it. Long teaches a backup and archiving process “with archival rollback capabilities,” as reflected in its flow charts illustrating the archiving of data. EX1004, FIGS. 4-6; *id.*,

¹ Reference numbers are added throughout in the format [claim#.limitation#].

² To the extent antecedent basis requires an order of certain steps, that is addressed in mapping the claim elements.

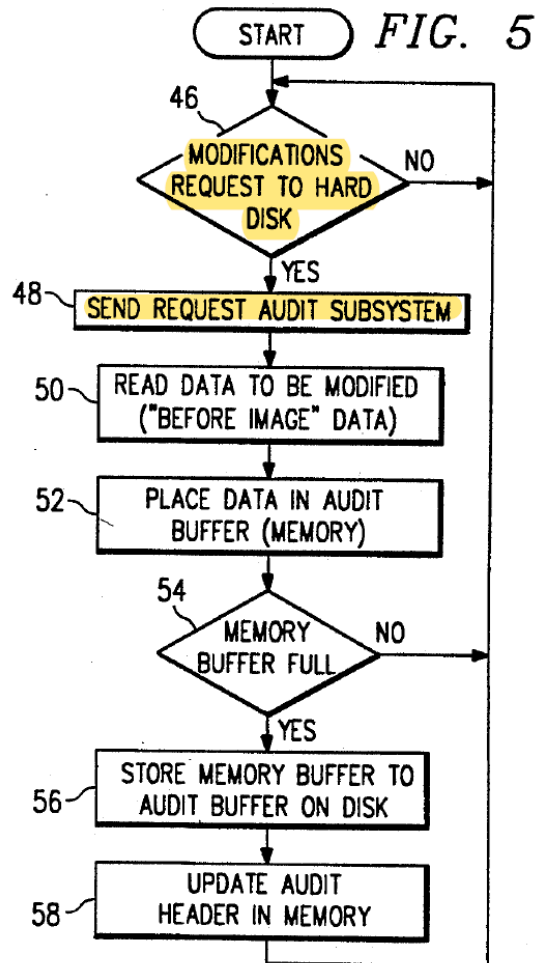
2:28-33; *id.*, 12:19-37 (Claim 31). When used to implement a file-based system such as Kenley's, the Long process archives files. *Supra* Section VIII.C.

[1.2] (a) the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;

60. Long teaches “detecting access requests to alter data on respective sets of one or more locations of the storage device.” EX1004, Abstract. Long specifies that “[p]rocessing circuitry detects accesses to the [storage] device which would alter data thereon.” EX1004, 1:67-68. Long further discloses that “[a] ‘modification request’ is a request from an application to modify the contents of the hard disk.” EX1004, 7:28-29; *id.*, FIG. 5. A modification request is generated “[w]hen the computer system 10 needs to write to a hard disk.” EX1004, 7:32-37. Thus, Long discloses detecting an instruction by a resident program in a computing device for performing an operation on data.

61. When used to implement a file-based system such as Kenley's, the Long audit subsystem process would detect instructions from user applications and the operating system for “performing an operation on an operating file.” *Supra* Section VIII.C. It was well known for applications and operating systems to request modifications to a file, e.g., write, creation, or deletion operations. EX1004, 1:48-53 (noting “typical failure” of a “power outage during a write operation in which the new data has been only partially written to the hard disk and the previous data has

been partially overwritten to the write operation”); *id.*, 4:23-25 (“the data on the hard disk is frequently changed during operation of an application program or by the operating system.”); *see also* EX1006, 1:11-13 (noting that “data and program files” are “often together referred to as ‘data’”); EX1008, 62 (illustrating data of a “normal ... file”); EX1005, 6:65-7:2 (data is traditionally stored in a file format); EX1010, 1:13-17 & 1:40-43; EX1012, 1:26-35. In the Kenley-Long system, any of these file modification requests would be detected by the audit subsystem. The operating system was a resident program by definition of the ’348 patent, and user applications were resident programs at least because they have control over file operations such as read, write, open, and save. *See* EX1001, 4:49-52 (defining “Resident Program”). Long’s audit subsystem is also a resident program as it is “part of the system BIOS” or a “memory-resident program.” EX1004, 6:60-63.



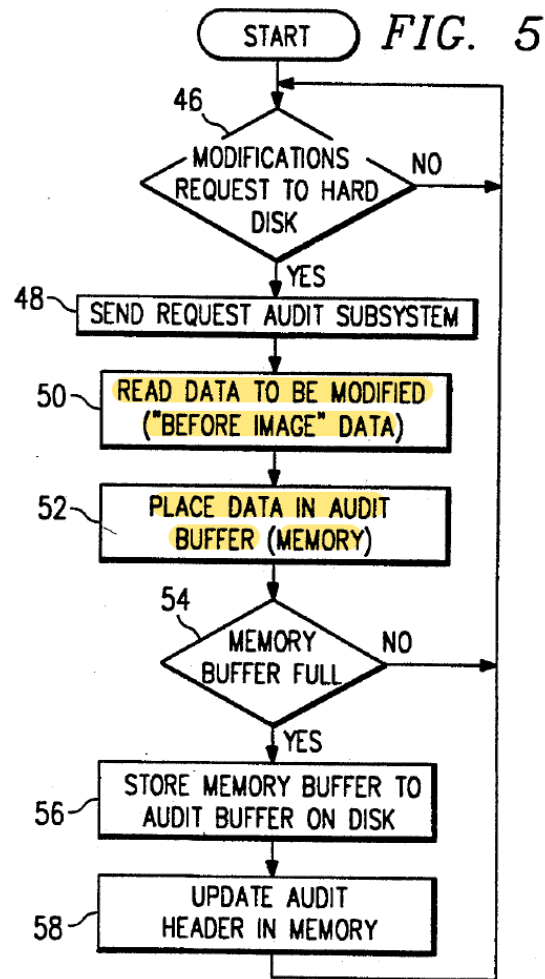
EX1004, FIG. 5 (highlighting added).

62. Kenley-Long thus includes “the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file.”

[1.3] (b) the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction;

63. Kenley discloses storing “archive copies” of “data files.” EX1014, Abstract. Long teaches creating a copy of the data to be modified and storing it in a

temporary storage location, responsive to detecting the modification request. For example, the actions “in response to a modification request” include (1) reading the data to be modified, i.e., the “before image” data, from the disk and (2) storing that data in “the memory buffer.” EX1004, 7:29-40, 7:43-46, 7:54-55, 7:59-64. These actions occur “temporally proximate” to the modification of the operating file at least because they (1) are responsive to the modification request and (2) the “requested write” or other operation (i.e., the modification) is made to the hard disk just after the actions are completed. *See* EX1004, 8:14-17, 7:33-40; EX1001, 5:12-17 (defining “temporally proximate” as “just before and/or just after”). The “before image” data is therefore created and placed in the memory buffer just before (i.e., “temporally proximate to”) the application modifying the contents of the hard disk. Long shows exemplary operations in FIG. 5:



EX1004, FIG. 5 (highlighting added).

64. When used to implement a file-based system such as Kenley's, these actions taught by Long result in a copy of the operating file being created and stored in the memory buffer (temporary storage) temporally proximate to the operating file being modified. Kenley teaches storing data in the form of files and thus a POSITA using Long's method to archive files would have initially stored the copy of the operating file (i.e., the archive file) in the memory buffer as taught by Long. EX1014, 3:1-2 ("the secondary storage maintains data in the form of files"), 3:29-30 ("the

backing store stores data organized as files”); *supra* Section VIII.C. This copy of the file is an archive file at least because it is a file that captures all of the data of an operating file. EX1001, 4:17-21 (defining “Archive File” as “a file containing all of the data of an operating file in a native or altered format and/or a file containing at least some of the data of an operating file...”); *see also* EX1014, Abstract (describing storing “archive copies” of “data files”).

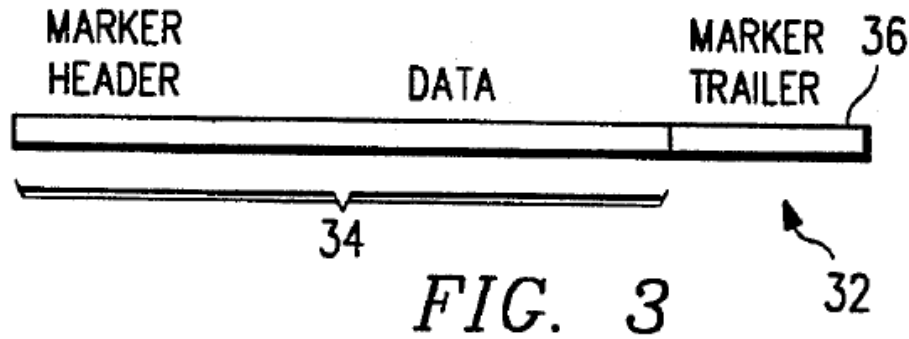
65. Thus, Kenley-Long includes “creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction.”

[1.4] (c) the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event;

66. Long teaches identifying the presence of the archived data in the memory buffer, responsive to a “first event,” in at least two ways. It teaches events that include (i) the memory buffer filling up and (ii) the “processing unit 14 requir[ing] memory used in the memory buffer for another purpose.” EX1004, 7:47-53. A POSITA knew that most obviously this information would arise in the form of messages from the operating system or some other resident program such as the BIOS. *E.g.*, EX1001, 5:46-49 (giving non-limiting examples of “event[s]” as messages from resident program(s)). Either event triggers the audit subsystem

identifying the data's presence in the memory buffer as reflected by the fact that, responsive to either of these two events, this data is moved to the audit buffer. EX1004, 8:20-23 (“When another part of the computer system 10 needs to write data to the hard disk, it will flush the memory buffer that contains the audit markers to the physical disk.”). To flush or move the memory buffers data, the system must identify the presence of the data in the memory buffer; absent such identification the system would neither know what it is moving nor when the move has completed. A POSITA would have understood that any one of these events—the memory buffer being full, the processing unit requiring memory in the memory buffer, the computer system needing to write data to the hard disk—constitutes a “first event,” as claimed, that triggers identifying presence of the data in the memory buffer.

67. Long further teaches creating other information that is used to identify the presence of the archived data in the memory buffer responsive to the occurrence of the events discussed above. For example, it creates a marker in the memory buffer during “[s]torage of the data in the memory buffer.” EX1004, 7:59-8:13. The “audit marker 32 comprises a marker header 34 and a marker trailer 36” where captured data “is appended to the marker header 34.” *Id.*, 5:30-34; *id.*, FIG. 3.



Id., FIG. 3.

68. Long teaches using the marker trailer and marker header to determine the location of the marker. For example, “[t]he marker trailer 36 is used to access the audit buffer 30 in reverse order,” and “contains ... information required to determine the location of the marker header.” *Id.*, 5:54-59. Furthermore, the audit header stores information regarding the markers, and the audit header is specifically updated with information regarding the last marker (“LastMarker”) which indicates that new archive data has been placed in the audit buffer. *Id.*, 5:22-24 (Table 1, “LastMarkerLoc”).

69. Additionally, “all data is written out to the physical disk before any of the actual data is,” which “is done by having the memory buffer semaphored.” *Id.*, 8:18-20. A semaphore on the memory buffer indicates the presence of the data within the memory buffer to other processes. *E.g.*, EX1019, 354 (defining “semaphore” as “a signal that is used to govern access to shared system resources. A semaphore is a flag variable—an indicator—that helps maintain order among processes that are

competing for use of such critical resources as microprocessor time and input/output ports.”). A POSITA therefore would have understood that the presence of the data in the memory buffer is identified in response to a process checking that semaphore (before writing to physical disk), which would constitute a “first event” as discussed above.

70. When used to implement a file-based system such as Kenley’s, these operations taught by Long result in the audit subsystem identifying the presence of the copied (archive) file in the memory buffer (temporary storage location) responsive to a first event. This identification would lead to the archive file being moved, or “flush[ed]” to the audit buffer on the secondary disk storage, where the archive file is stored. *E.g.*, EX1004, 8:20-23, 7:47-53; *see also* EX1014, 3:1-4 (“A file storage element in the secondary storage maintains data in the form of files ...”).

71. Thus, Kenley-Long includes “the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event.”

[1.5] (d) the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location;

72. Long teaches transmitting the copied data to a second storage location responsive to a second event. For example, it teaches “archiving data to removable media” or “archive media.” EX1004, 7:1-3; *id.*, 7:10-14; *id.*, 8:29-9:9; *id.*, FIG. 6.

This archiving to the removable storage media is responsive to a second event as it occurs “at predetermined intervals (for example, every five seconds),” based on a “five second ticker.” EX1004, 8:36-42. Each completion of the ticker is an event, and a POSITA would have naturally implemented the ticker to alert the system after the passage of 5 seconds. Long discloses other events that trigger the writing of data to the archive media, e.g., detecting that the archive media is mounted with available storage space. *Id.*, FIG. 6, 8:52-59, 7:18-26. For purposes of claim 1, the removable media storage is a permanent storage because it is a persistent “archive media for external storage.” *Id.*, 8:33. Long FIG. 6 provides an exemplary data archive process, with 66 showing the data read from the audit buffer and 68 showing the data written to the archive (removable) media. *See also* EX1004, 8:63-9:7 (describing reading data from the audit buffer and writing it to the archive media). In reading the data from the audit buffer and writing it to the archive storage, Long teaches transmitting the data to a second storage location.

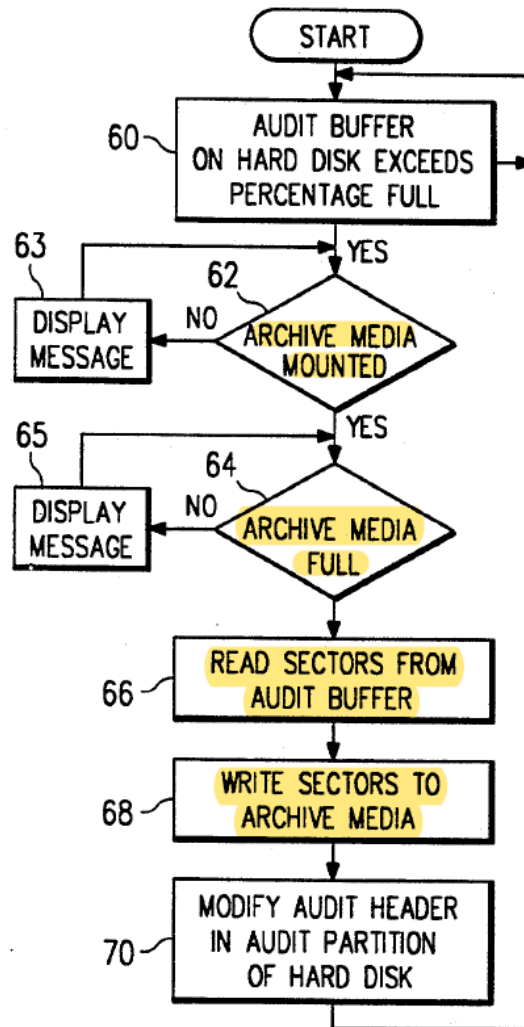


FIG. 6

EX1004, FIG. 6 (highlighting added).

73. I note that the “NO” in step 64 of FIG. 6 is mislabeled. Long explains that “[o]nce the archive media is mounted, it is determined whether the archive media is full in decision block 64. If full, a message is displayed (block 65) until a new media is mounted which is not full.” EX1004, 8:55-59. A POSITA therefore would have understood that the “NO” between blocks 64 and 65 should read “YES” or that the “NO” should be placed between blocks 64 and 66.

74. When used to implement a file-based system such as Kenley's, these operations taught by Long result in the transmitting of the copied (archive) file to the "backing store" taught by Kenley. *E.g.*, EX1014, 3:29-30 ("the backing store stores data organized as files"). The archive media of Long parallels the backing store of Kenley, as both are additional, slower, and higher-volume storage separate from the local computer disk.

75. Kenley-Long thus includes "the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location."

[1.6] (e) after storing the archive file in the temporary storage location, updating a database to indicate that the archive file is located in the temporary storage location;

76. Kenley-Long renders this step obvious. Kenley teaches a relational database that holds a "system catalog" for storing and maintaining the locations of backup copies of a file after the backup copy is moved (migrated). *E.g.*, EX1014, 3:17-28 (describing using a system catalog element and "maintaining" it after migrating files to the backing store), 6:35-40 (disclosing relational database storing "backup catalogs (i.e., lists of all files copied to specific backup savesets)"), 8:26-29 (describing recording to the "database 14 a catalog of all files backed up..."); *see also id.*, 2:6-14 (identifying not tracking the location of backup copies as a problem to be solved).

77. Long, in turn, teaches an audit header that is updated to reflect changes in the data's storage location, including when the copied data is stored within the memory buffer (temporary storage location). EX1004, 7:49-53. The audit header "contains information that is used to manage the audit buffer 30, archive media ... and the general environment of the audit subsystem." *Id.*, 4:44-47; *id.*, 4:57-60, 2:5-8. The audit header includes a number of fields that specify the location of data managed by the audit subsystem, whether that data is stored in the memory buffer, audit buffer, or archive media. *Id.*, 4:61-5:24 (Table 1). For example, the audit header includes a "CacheLoc" field that provides the "current location of the audit buffer in the disk cache," a "BufLoc" field that provides the "physical location of the beginning of the audit buffer on disk," and a "CLoc" field that provides the "physical location of the last data in the buffer." *Id.* Long discloses "construct[ing] the audit header" for a marker after "[s]torage of the data in the memory buffer" in the marker. *Id.*, 7:59-68. Then, when the data is moved from the memory buffer (temporary storage) to the audit buffer, the audit header is updated "to reflect the changes." *Id.*, 7:47-53. Constant updating of this audit header to indicate the location of the data in the memory buffer and other locations, after storing the data in the respective locations, facilitates the audit subsystem's accurate retrieval and movement of data for both backup and recovery.

TABLE 1

Audit Header Fields	
Data Label	Data Description
Ver	The version of the audit system that created the audit partition.
Name	A name to assist in determining that the audit partition is valid.
HdrSeqNum	Sequence number that is used every time the header is written to disk.
Time	The time the audit partition was created or reset (cleared out).
CloseErr	The last audit error. This is used to determining if a failure occurred and the system came down before the audit

TABLE 1-continued

Audit Header Fields	
Data Label	Data Description
	header was updated on the hard disk(s).
SeqNum	The next Archive disk sequence number to use.
QSeqNum	The last archive disk that contains a user quiescence marker.
Flush	A note to the system that the audit header needs to be written to disk at the next available opportunity (used in memory only).
LabelLoc	The physical location of the label on disk (audit header).
BufLoc	The physical location of the beginning of the audit buffer on disk.
CacheLoc	The current location of the audit buffer in the disk cache.
CLoc	The physical location of the last data in the buffer. The next update of the buffer will begin at this address. This is on a sector by sector basis.
CLocBytes	Current offset in the current sector that we are at (see CLoc).
MaxSectors	The number of sectors that make up the audit buffer. This does not include the first sector that contains the audit header.
MaxUsable	When the audit buffer gets to this point the archive system will start archiving.
MinUsable	When the audit buffer gets to this point the archive system will stop archiving.
UsedSectors	The number of sectors that have been used in the audit buffer.
CurSector	The logical sector number currently in use by the audit buffer.
FirstMarker	The contents of the first marker header.
FirstMarkerLoc	The location of the first marker in the audit buffer.
FirstMarkerOffset	The offset in the sector that contains the first marker.
FirstMarkerSector	The logical sector that contains the first marker.
LastMarker	The contents of the last marker header.
LastMarkerLoc	The last marker location in the audit buffer.
LastMarkerOffset	The offset in the sector that contains the last marker.

EX1004, 4:61-5:24 (Table 1) (highlighting added).

78. Thus, in view of these complementary teachings, a POSITA would have been motivated with a reasonable expectation of success to implement Kenley-Long such that information on current locations of different backup files, including the relevant audit-header information taught by Long, was stored in the relational database catalog taught by Kenley. The audit header of Long, with all of its fields, already is nothing more than a database of information and thus a POSITA would have easily implemented the Kenley-Long system so as to store Long's audit header

information in Kenley's relational database. The '348 patent itself only describes a "database" functionally as a place to record information "indicat[ing] the location of the archive files." *E.g.*, EX1001, 6:29-38.

79. And as with other elements, when used to implement a file-based system such as Kenley's, these database updates taught by Long would indicate that the copied file (archive file) is located in the memory buffer (temporary storage location).

80. Kenley-Long thus includes "after storing the archive file in the temporary storage location, updating a database to indicate that the archive file is located in the temporary storage location."

[1.7] (f) determining a final destination for the archive file;

81. The '348 patent provides no explanation for the term "final destination," nor is the recited "final destination" recited elsewhere in the claims.

82. Kenley teaches to predetermine the location of the data in its backing store archive storage, as it specifies moving "data contained in the selected data files to a predetermined staging volume of the backing store element." EX1014, 3:9-12.

83. Similarly, as explained for Element [1.4], Long teaches "archiving data to removable media." EX1004, 7:1-14, 8:29-9:9, FIG. 6. Before storing the data to the removable media, Long first "determine[s] whether archive media is mounted" and "whether the archive media is full." *Id.*, 8:50-57. Archiving of the audit buffer

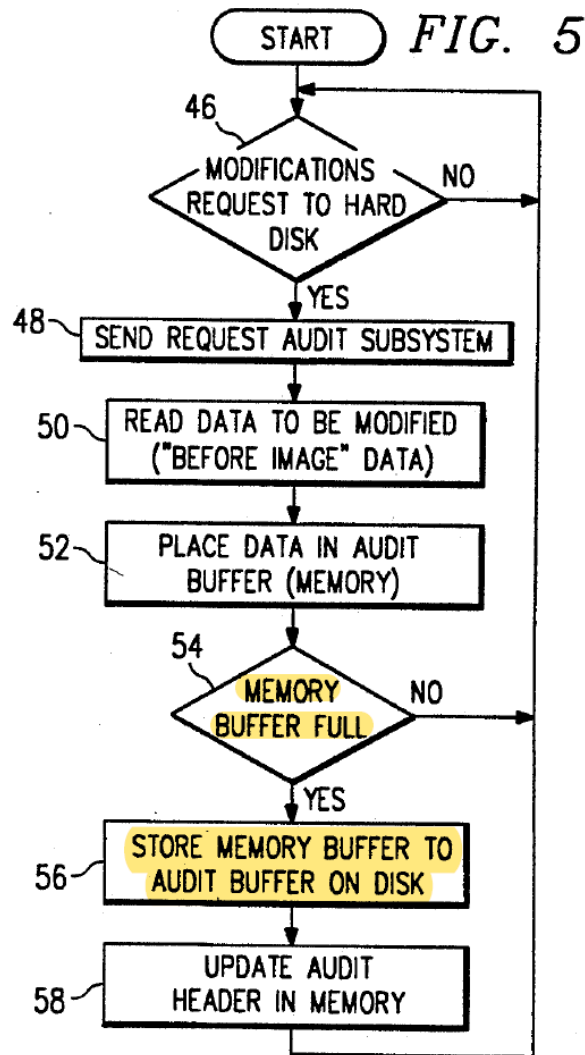
begins once “a new media is mounted which is not full.” *Id.*, 8:57-59. Long further discloses preparing an “archive track” to be written to the archive media, including an “archive end” marker at the end of the data to indicate the end of the archive track. *Id.*, 9:3-6; *id.*, 8:38-42 (“the size of the track is dependent upon the structure of the archive media”). Fields in the audit header specify an “Archive disk sequence number to use.” EX1004, 4:61-5:24 (Table 1) (describing a “SeqNum” as “[t]he next Archive disk sequence number to use” and “QSeqNum” as “[t]he last archive disk that contains a user quiescence marker”). Long thus teaches determining a final destination within the archive media to store the archive data, in order to prepare the “archive track.”

84. Thus, a POSITA would have implemented Kenley-Long such that the final location of each archived file was predetermined, e.g., determined before moving the file, at least so that the system would know where in secondary storage to move the archive file during the archive process.

[1.8] (g) moving the archive file from the temporary storage location to an intermediate storage location;
--

85. Long teaches moving the copied data from the memory buffer (temporary storage) to the audit buffer (intermediate storage location) prior to ultimately archiving the data to the removable media. EX1004, 7:47-53. For example, Long’s audit subsystem “writes the data stored in the memory buffer to the

end of the audit buffer 30 on disk” when the memory buffer is full or when the memory buffer is needed for another purpose. EX1004, 7:47-53; *id.*, 8:20-23 (“[F]lush the memory buffer that contains the audit markers to the physical disk.”); *id.*, FIG. 5.



EX1004, FIG. 5 (highlighting added).

86. The audit buffer is an intermediate storage location when archiving is enabled as taught by Long, at least because the data is stored in the audit buffer in

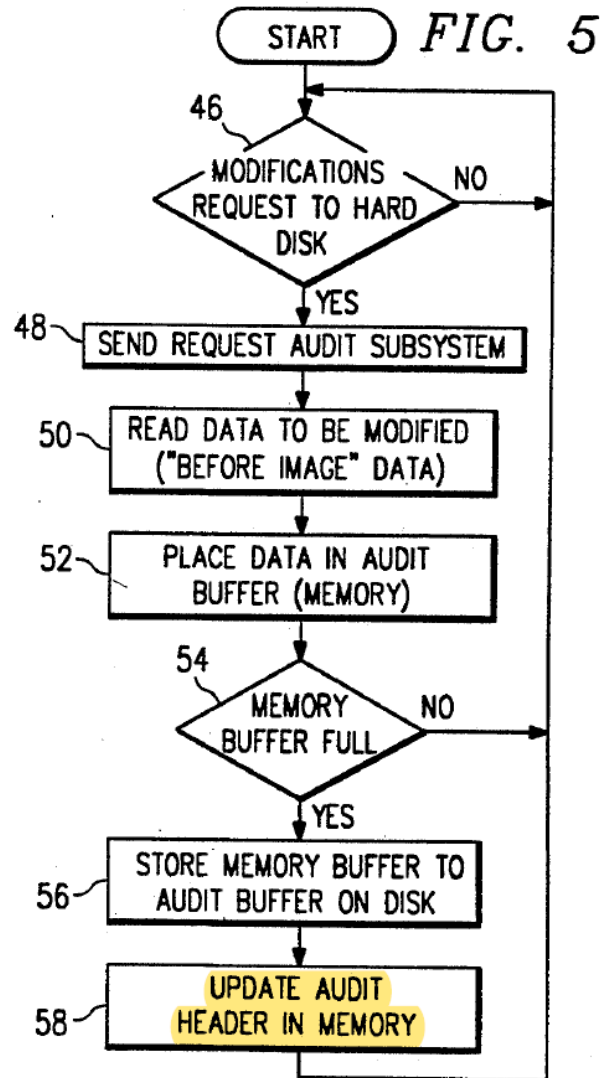
between being stored in the memory buffer (temporary storage) and the archive media (second or permanent storage location). EX1004, 8:29-9:9 (explaining the archive process); *id.*, FIG. 6.

87. When used to implement a file-based system such as Kenley's, this action would involve movement of the copied file (archive file) to an audit buffer as taught by Long, in a hard disk secondary storage such as that taught by Kenley. EX1014, 3:1-4 ("A file storage element in the secondary storage maintains data in the form of files"), 4:56-60 (secondary storage as hard disk).

88. Kenley-Long thus includes "moving the archive file from the temporary storage location to an intermediate storage location."

[1.9] (h) updating the database to indicate the archive file is located in the intermediate storage location; and

89. Long teaches updating its audit header to indicate the copied data (archive data) is located in the audit buffer (intermediate storage): after "writ[ing] the data stored in the memory buffer to the end of the audit buffer 30 on disk" Long specifies "updat[ing] the audit header (in the memory buffer) to reflect the changes." EX1004, 7:47-53; *id.*, FIG. 5.



EX1004, FIG. 5 (highlighting added). The audit header is updated to indicate “[t]he physical location of the beginning of the audit buffer on disk.” EX1004, 4:61-5:24 (Table 1) (defining “BufLoc”).

90. As discussed above for Element [1.6], Kenley teaches a complementary relational database with a backup catalog containing, e.g., “lists of all files copied to specific backup savesets.” EX1014, 6:35-40. For the same reasons explained above, a POSITA would have naturally implemented Kenley-Long such that a backup

catalog database, as taught by Kenley, was updated with the location of the copied file (archive file) in the secondary storage audit buffer, upon movement of the copied file to that secondary storage. *Supra* Section VIII.C.1 (Element [1.6]).

91. Kenley-Long thus includes “updating the database to indicate the archive file is located in the intermediate storage location.”

[1.10] (i) after moving the archive file to the second storage location, updating the database to indicate the archive file is located in the second storage location.

92. Long teaches updating its audit header to indicate that the copied (archived) data is located in the removable archive media (second storage location), after the data has been moved to that location. Long teaches, “[a]fter the write [to the archive media] is completed, the audit header in memory is modified to show the removal of the audit markers from the audit buffer 30.” EX1004, 9:7-9; *id.*, FIG. 6.

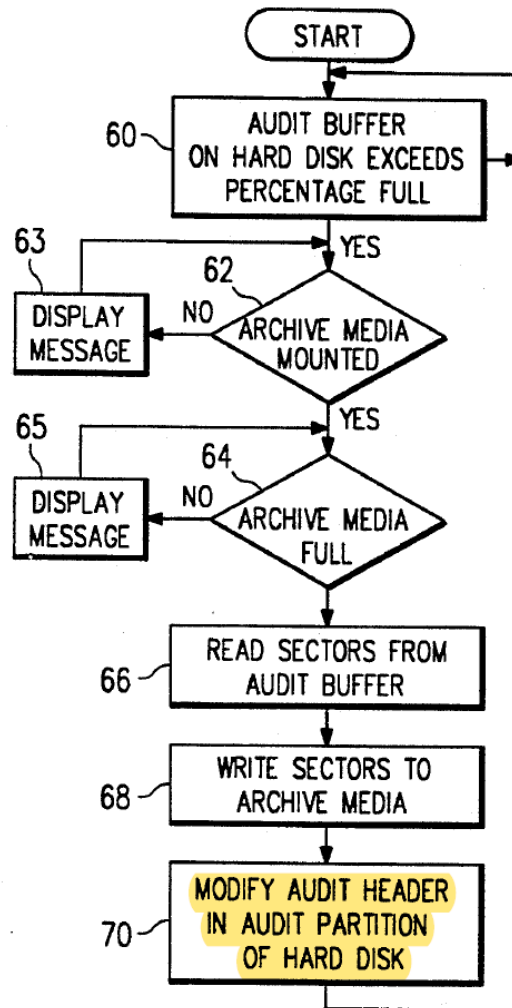


FIG. 6

EX1004, FIG. 6 (highlighting added).

93. The audit header includes fields specifying the location of data in the archive media that are “used to manage the ... archive media,” such as the “Archive disk sequence number to use” and the “last archive disk that contains a user quiescence marker.” *Id.*, 4:44-47; *id.*, 4:61-5:24 (Table 1) (describing a “SeqNum” as “[t]he next Archive disk sequence number to use” and “QSeqNum” as “[t]he last archive disk that contains a user quiescence marker”). A POSITA would have

understood the audit header as a database indicating data locations. *Supra* Section VIII.C.1 (Claim [1.6]).

94. As discussed above for Elements [1.6] and [1.9], Kenley teaches a database with a backup catalog containing, e.g., “lists of all files copied to specific backup savesets.” EX1014, 6:35-40. For the same reasons explained above, a POSITA would have naturally implemented Kenley-Long such that a backup catalog database, as taught by Kenley, was updated with the location of the copied file (archive file) in the removable archive media or backing store (second, permanent storage location), upon movement of the copied file to removable media. *Supra* Section VIII.C.1 (Element [1.6]).

95. Kenley-Long therefore includes, “after moving the archive file to the second storage location, updating the database to indicate the archive file is located in the second storage location.”

2. Claim 2

[2] The method of claim 1, wherein the second storage location includes a personal attached storage device.

96. The '348 patent defines “Personal Attached Storage Device” as “any internal or external storage device connected to a computing device.” EX1001, 4:32-33. In the Kenley-Long system, it would have been an obvious design choice to have the backing store (second storage location) use one of the many storage devices

taught by Kenley and Long, each of which is either an internal or external storage connected to the computing device, e.g.,:

- backing store devices taught by Kenley, such as “erasable optical, write-once-read-many (WORM) disks, or tape volumes” (EX1014, 4:61-66), or
- the removable “archive media for external storage” taught by Long (EX1004, 8:32-33; *id.*, 7:12-14 (“removable media, such as a floppy disk”), 2:46-50 (disclosing a computer including “floppy drives 16 (or other removable media device)”), FIG. 1).

These devices are connected to the computing device, as reflected by the fact that the computing device can read and write data from and to these devices.

97. Kenley-Long therefore discloses “[t]he method of claim 1, wherein the second storage location includes a personal attached storage device.”

3. Claim 3

[3] The method of claim 1, wherein the second storage location includes a network attached storage device.
--

98. The '348 patent defines “Network Attached Storage Device” as “any storage device connected directly to a network to which a first computing device is also temporarily or permanently connected, or any storage device connected to a second computing device that is also temporarily or permanently connected to the

network to which the first computing device is temporarily or permanently connected”). EX1001, 4:34-40.

99. In implementing the Kenley-Long system, it would have been at least obvious to include a network attached storage device as the archive media or backing store. Kenley, as background, notes “particular interest” in “the Network File System (NFS) protocol, developed by Sun Microsystems, allowing users to share files across a local area network such as Ethernet.” EX1014, 1:30-33; *see also id.*, 1:42-45 (noting “data storage requirements of professional work groups employing networked workstations”), 2:11-14 (discussing “migrating data ... so that the files can be read and utilized by computers or processors in the network”). Kenley then teaches use of the “conventional NFS network” to provide “rapid and transparent data migration” by “sending conventional NFS requests and responses to the network.” EX1014, 13:42-50. It thus would have been at least obvious to implement Kenley-Long such that it used the “conventional NFS network” to connect the personal computer or other computing device to the devices providing the temporary and intermediate storage locations. *E.g.*, EX1014, 2:11-14 (presenting one problem to solve as “transferring computer data files ... so that the files can be read and utilized by computers or processors in the network”); *id.*, 6:29-66, FIGS. 1A-B & 2.

100. Thus, it would have been obvious to implement Kenley-Long such that the backing store (second storage location) included a network attached storage

device, e.g., connected via an Ethernet with files accessible using the conventional network file sharing (“NFS”) protocol, as taught by Kenley.

101. Kenley-Long therefore discloses or renders obvious “[t]he method of claim 1, wherein the second storage location includes a network attached storage device.”

4. Claim 4

[4] The method of claim 1, wherein the second storage location includes a peer-to-peer storage device.

102. The ’348 patent defines “Peer-to-Peer Storage Device” as “any storage area (device, collection of devices, etc.) that can be accessed by the computing device when it is sharing resources with other network or Internet accessible computers.” EX1001, 4:45-48.

103. In implementing the Kenley-Long system, it would have been at least obvious to include a peer-to-peer storage device as the archive (backing) store, for the same reasons provided above regarding Claim 3. Kenley explains that “[t]he data storage requirements of professional work groups employing networked workstations can measure billions of bytes—i.e., Gigabytes—in size,” and that “[a]dditional file servers can be incorporated into an existing computer network” to increase data storage capacity. EX1014, 1:42-51. Thus, it would have been obvious to implement Kenley-Long such that the second storage location includes a peer-to-

peer storage device, because Kenley describes incorporating a file server (i.e., storage area) into an existing computer network that can be accessed by a computing device when it is sharing resources (i.e., data storage resources) with networked workstations (i.e., network-accessible computers). *See infra* Section VIII.C.5 (explaining obviousness of shared internet-based external storage device for the archive).

104. Kenley-Long therefore discloses or renders obvious “[t]he method of claim 1, wherein the second storage location includes a peer-to-peer storage device.”

5. Claim 5

[5] The method of claim 1, wherein the second storage location includes an internet storage area network.

105. The ’348 patent defines “Internet storage area network” as “any storage area (device, collection of devices, etc.) that can be accessed by the computing device when the computing device is temporarily or permanently connected to the Internet.” EX1001, 4:41-44.

106. As early as 1993, it was well known that the Internet was “commonly used to transfer ... files between computers.” EX1018, 3:13-19 (background provided in patent issued from application filed November 4, 1993). Even then, the Internet was a known alternative to the local area network (“LAN”), e.g., Ethernet, taught by Kenley (EX1014, 1:30-33). “The Internet is similar to a LAN/WAN in that

it was designed to allow many computers of differing types to interconnect and exchange information and programs.” EX1018, 3:37-39; *see also* EX1005, 6:9-36 (1999 patent describing remote backup storage where “logical connections to one or more remote computers,” such as a “memory storage device,” included “commonplace” LAN and WAN connections, “such as the Internet”).

107. It was also well known to access external drives using the Internet, as “[d]rives on a remote system defined using NFS . . . [we]re available for mounting by other remote systems on the Internet.” EX1018, 3:64-66; EX1005, 6:9-36; *see also supra* Section VIII.C.3. By 2000, people had been “sharing data on the Internet for decades already” (EX1007, 544) and it was known that “[o]ne of the earliest Internet-based storage applications to arise since the emergence of the World Wide Web is system backup and recovery” (*id.*, 556). Thus, the POSITA knew how to implement Internet-connected drives, and that they were an established design choice for backup and recovery storage drives.

108. Given this knowledge, and Kenley’s teachings to use the NFS protocol for transmission of files to networked storage devices (*e.g.*, EX1014, 1:30-33, 13:44-48), a POSITA would have naturally implemented the backing store of Kenley-Long using a removable media (*e.g.*, optical disk) accessed via a wide-area network such as the Internet. At the very least, this was an obvious, well-established design choice. Moreover, the POSITA would have been motivated to do so for at least two benefits.

First, it reduced the chances of the backup information being destroyed. EX1007, 556 (“[T]he chances are pretty good that the web site will not be destroyed by the same disaster that strikes a system in the office or at home.”). Second, it would beneficially conserve local storage space and power and allow computers in different locations to access and share external storage disk resources. EX1018, 13:28-31; *see also id.*, 3:64-66 (“Drives on a remote system defined using NFS and RFS are available for mounting by other remote systems on the Internet.”). A POSITA would easily implement this known alternative using the NFS protocol taught by Kenley. *E.g.*, EX1018, 3:61-63; *see also id.*, 34:46-65 (noting it was “well known” (in 1993) that “personal computers ... communicate with various devices” using an “interrupt” to access “a shared resource such as the data bus or hard disk”).

109. The obviousness of this design choice is confirmed by the '348 patent itself, which states that the storage device “may be a personal attached storage device, a network attached storage device, an Internet storage area network, a peer-to-peer storage device, or other storage device,” without any explanation of how to implement any of these alternatives or why one would be preferred over the other. EX1001, 5:32-35. It treats them as interchangeable and easily implemented because they were. Accordingly, claim 5 was obvious over Kenley, Long, and the knowledge of a POSITA.

110. Kenley-Long therefore discloses or renders obvious “[t]he method of claim 1, wherein the second storage location includes an internet storage area network.”

6. Claim 6

[6] An article of manufacture comprising a computer usable medium having computer readable program code for performing the method of claim 1.

111. Kenley-Long renders this claim obvious. The method of claim 1 is obvious in view of Kenley-Long. *Supra* Section VIII.C.1. Long discloses that “the audit subsystem is part of the system BIOS (Basic Input/Output System), but could also be a memory-resident program.” EX1004, 6:60-63; *see also* EX1014, 1:6-9 (disclosing the Kenley method implemented as “software”). Long further discloses “[p]rocessing circuitry” that performs the method of claim 1. *See, e.g.*, EX1004, 1:67-2:2, *id.*, 2:45-57 (describing the processing unit 14 of the computer system 10); *id.*, FIGS. 1, 2a; *id.*, 10:10-11:13 (Claims 1-15) (claiming a computer system comprising processing circuitry to perform the audit-archive method of Long).

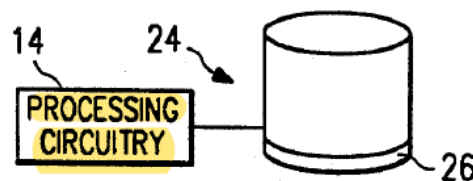


FIG. 2a

EX1004, FIG. 2a (highlighting added).

112. Kenley-Long thus discloses an “article of manufacture comprising a computer usable medium having computer readable program code for performing the method of claim 1.”

7. **Claim 8**

113. Elements [8.1]-[8.6] are identical to elements [1.1]-[1.6]. For example, the table below depicts elements [1.1]-[1.6] next to [8.1]-[8.6], with green highlighting in the Claim 8 column showing exact identity with Claim 1.

Claim 1	Claim 8
[1.1] A method for archiving files, comprising steps of (a) to (i) following:	[8.1] A method for archiving files, comprising the steps of:
[1.2] (a) the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;	[8.2] (a) the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;
[1.3] (b) the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction;	[8.3] (b) the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction;
[1.4] (c) the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event;	[8.4] (c) the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event;
[1.5] (d) the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an	[8.5] (d) the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an

Claim 1	Claim 8
intermediate or a permanent storage location;	intermediate or a permanent storage location;
[1.6] (e) after storing the archive file in the temporary storage location, updating a database to indicate that the archive file is located in the temporary storage location;	[8.6] (e) after storing the archive file in the temporary storage location, updating a database to indicate that the archive file is located in the temporary storage location;

114. Elements [8.1]-[8.6] are therefore satisfied by Kenley-Long for the same reasons as explained above for elements [1.1]-[1.5]. *See supra* Section VIII.C.1.

[8.7] (f) the step of moving the archive file from the temporary storage location to an output buffer;
[8.8] (g) the step of updating the database to indicate that the archive file is located in the output buffer;

115. Kenley-Long satisfies these elements, as it renders obvious moving the copied file (archive file) from the memory buffer (temporary storage) to the audit buffer (output buffer), and then updating the database catalog and audit header to indicate the location of the copied file in the audit buffer. *See supra* Section VIII.C.1 (Elements [1.8-1.9]). The '348 patent only describes an “output buffer” functionally as a place where the data is moved prior to being moved to archive storage. *See* EX1001, 5:57-62. An audit buffer, as taught by Long and implemented in the combined Kenley-Long system, thus satisfies the claimed “output buffer.”

[8.9] (h) the step of transmitting the archive file from the output buffer to the intermediate or the permanent storage location; and

116. Kenley-Long satisfies this element, as it renders obvious transmitting the copied (archive) file from the audit buffer (output buffer) to the backing store or archive media (permanent storage location). *Supra* Section VIII.C.1 (Elements [1.5]); *see also* EX1004, 7:1-3; *id.*, 7:10-14; *id.*, 8:29-9:9; *id.*, FIG. 6. As the archive media (e.g., backing store) is separate from the audit buffer, the copied file must be transmitted to the archive media.

[8.10] (i) the step of updating the database to indicate that the archive file is located in the intermediate or the permanent storage location.

117. Kenley-Long satisfies this element. *Supra* Section VIII.C.1 (Claim [1.10]).

8. Claims 9-11

118. Claims 9-11 are identical to claims 2-3 and 5, respectively (other than the independent claim they cite), and are obvious over Kenley-Long for the same reasons as explained above for claims 2-3 and 5. *Supra* Sections VIII.C.2, VIII.C.3, VIII.C.5.

9. Claims 12-15

119. Elements [12.1]-[12.5], [13.1]-[13.5], [14.1]-[14.5], and [15.1]-[15.5] are identical to one another and to Elements [1.1]-[1.5]. The table below compares

these elements of claims 12-15 to the corresponding elements of claim 1, with green highlighting to indicate exact correspondence to the language of claim 1.

Claim 1	Claim 12	Claim 13	Claim 14	Claim 15
[1.1] A method for archiving files, comprising steps of (a) to (i) following:	[12.1] A method for archiving files, comprising steps of (a) to (d) following:	[13.1] method for archiving files, comprising steps of (a) to (d) following:	[14.1] A method for archiving files, comprising steps of (a) to (d) following:	[15.1] A method for archiving files, comprising steps of (a) to (d) following:
[1.2] the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;	[12.2] the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;	[13.2] the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;	[14.2] the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;	[15.2] the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;
[1.3] the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being	[12.3] the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being	[13.3] the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being	[14.3] the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being	[15.3] the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being

Claim 1	Claim 12	Claim 13	Claim 14	Claim 15
performed on the operating file and responsive to detecting the instruction;	performed on the operating file and responsive to detecting the instruction;	performed on the operating file and responsive to detecting the instruction;	performed on the operating file and responsive to detecting the instruction;	performed on the operating file and responsive to detecting the instruction;
[1.4] the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event;	[12.4] the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event; and	[13.4] the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event; and	[14.4] the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event; and	[15.4] the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event; and
[1.5] the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location;	[12.5] the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location,	[13.5] the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location,	[14.5] the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location,	[15.5] the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location,

120. Given that these Elements are identical to Elements [1.1]-[1.5], Elements [12.1]-[12.5], [13.1]-[13.5], [14.1]-[14.5], and [15.1]-[15.5] are satisfied by Kenley-Long for the same reasons as explained above for Elements [1.1]-[1.5]. *Supra* Section VIII.C.1.

a) **Element [12.6]**

[12.6] wherein the second event includes a message indicating when the second storage location is available.
--

121. Long teaches “determin[ing] whether archive media is mounted” and then “whether the archive media is full” prior to transmitting the audit data from the audit buffer to the archive media. EX1004, 8:50-59; FIG. 6; *see also* EX1014, 5:36-39, 5:57-63, 6:40-43 (discussing mount queue manager (MQM) to manage access to backing store volumes), 16:38-59 (explaining that “[n]o I/O processes can occur until the volume is physically mounted in a drive” so data migration is blocked while MQM directs mounting of the volume). Making these determinations would have required some kind of communication between the audit subsystem and the removable archive media device, and thus it would have been at least obvious to implement Kenley-Long such that messages indicating the mounting and availability of the removable device would be sent to the audit subsystem.

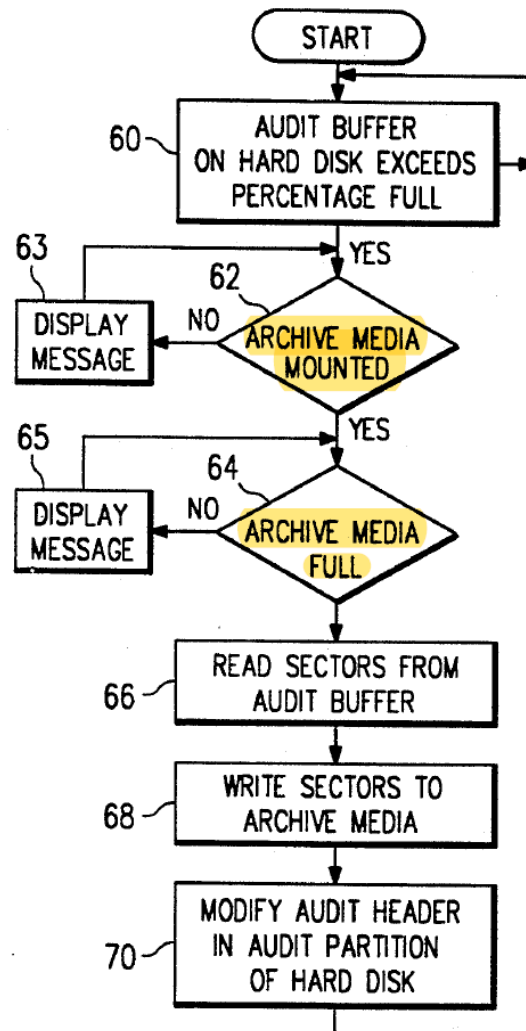


FIG. 6

EX1004, FIG. 6 (highlighting added). As discussed above in paragraph 73, the “NO” label at 64 in FIG. 6 should read “YES” to be consistent with Long’s disclosure.

b) Element [13.6]

[13.6] wherein the second event includes a message from a timer.

122. Long teaches that the second event includes a message from a timer. *Supra* Section VIII.C.1 (Element [1.5]). It would have been obvious to implement

Kenley-Long such that the “five second ticker” taught by Long (EX1004, 8:36-37) generated a message to signal completion of a five second (or other) time increment and trigger transmission of the file to the archive media (backing store). EX1004, 8:38-42.

c) Element [14.6]

[14.6] wherein the first event includes a message from a timer.

123. Kenley-Long renders this element obvious. *Supra* Section VIII.C.1 (Element [1.5]). Long teaches a timer for archiving data from the audit buffer. EX1004, 8:36-42 (“five second ticker”). Long also teaches flushing the memory buffer to the audit buffer on disk “[w]hen the memory buffer is full ... or the processing unit 14 requires memory used in the memory buffer for another purpose.” EX1004, 7:47-53. A POSITA would have also found it obvious to implement a “ticker” (i.e., timer) for triggering transfer of the audit data in the memory buffer to the audit buffer, in order to further “increase the throughput of the audit subsystem” as doing so would allow the data transfer “to take place in the background” and thus be “asynchronous to the rest of the system.” EX1004, 7:43-46; *id.*, 8:42-45. The express motivation for using the timer to trigger the archival transfer also applies to the memory buffer. The POSITA would have easily implemented such a timer, with a reasonable expectation of success, as reflected in Long’s teachings to use such a timer for similar data transfers. The ease of implementation is further confirmed by

the '348 patent's failure to provide any meaningful description of how to implement such a timer. Other background prior art further confirms the obviousness of the claimed timer. *See, e.g.*, EX1005, 10:53-55 (“the triggering event ... would be the expiration of an elapsed time rather than the accumulation of an amount of data”); *id.*, 11:15-16 (“the triggering event may comprise the expiration of a fixed time”).

d) Element [15.6]

[15.6] wherein the first event is different from the second event.
--

124. Long teaches that the events causing movement of data from the memory buffer (temporary storage) to the audit buffer (intermediate storage) are distinct from the events causing movement of data from the audit buffer to the archive media (permanent storage). *Compare* EX1004, FIG. 5 *with id.*, FIG. 6; *see also supra* Section VIII.C.1 (Elements [1.4]-[1.5]). Long teaches several types of “first event” that lead to moving data from the memory buffer (e.g., the memory buffer being full, the processing unit requiring memory). *Supra* Section VIII.C.1 (Claim [1.4]). It teaches distinct types of “second event” that lead to movement of data from the audit buffer (e.g., the audit buffer reaches a certain capacity or a timer). EX1004, 8:38-42; *supra* Section VIII.C.1 (Claim [1.5]). Thus, it was obvious to implement Kenley-Long such that the first event is different from the second event.

10. Claim 16

[16] The method of claim 15, wherein the resident program is separate from an operating system.

125. In Kenley-Long, modification requests (instructions for performing an operation on an operating file) originate from an application or the operating system, and are received by the audit subsystem. *Supra* Section VIII.C. When the request originates from an application and is received by the audit subsystem, the resident program is separate from the operating system, regardless of whether the claim requires the resident program receive or issue the instruction.

11. Claim 17

[17] The method of claim 16, where the resident program comprises a program having control over file operations on one or more operating files.

126. In Kenley-Long, an application controls file operations on at least one operating file as it is able to change, e.g., “write” data. EX1004, 4:23-25, 7:28-29, 7:40-42; *supra* Section VIII.C.1. The ability of the “application to modify the contents of the hard disk” as taught by Long indicates that the application comprises “a program having control over file operations on one or more operating files.” EX1004, 7:28-29. Additionally, the audit subsystem also has control over file operations, as it is able to read and write data from and to memory, the local hard disk, and archive media. *E.g.*, EX1004, 7:28-64, FIGS. 5-6. Thus, Kenley-Long renders claim 17 obvious.

12. Claim 18

[18] The method of claim 15, wherein the operation comprises one of read, write, save, rename, delete, copy, move, open, close, or create operations.

127. Kenley-Long renders claim 15 obvious. *Supra* Section VIII.C.9. Kenley discloses “read” and “write” access to a file, “open/close” operations, and “file delete and rename operations.” EX1014, 11:29-32, 17:23-25 (discussing monitoring “all open/close and read/write operations”). Furthermore, a POSITA would have understood that the ability of the “application to modify the contents of the hard disk” as taught by Long indicates that the operation comprises at least one of read, write, save, rename, delete, copy, move, open, close, or create operations, insofar as these operations modify the contents of the hard disk. EX1004, 7:28-29. Certain operations such as write, save, rename, delete, move, copy, and create modify file contents, while other operations such as read, open, and close do not typically modify file contents, though these operations may modify file metadata such as “last access time.” *E.g.*, EX1015, 370-71; EX1014, 18:39-42 (noting that time of “last access” is stored for each file).

13. Claim 19

[19] The method of claim 15, wherein the operating file comprises one of a system file or a user file.

128. Kenley-Long renders this claim obvious. *Supra* Section VIII.C.9 (Kenley-Long render Claim 15 obvious). Kenley teaches backing up and archiving

both system and user files. EX1014, 2:6-10 (discussing “users” and “their own information files”), 3:29-32 (discussing “enabling a user or an applications program to request selected files stored in backing store”). Long teaches archiving both system and user files. EX1004, 4:23-25 (“[T]he data on the hard disk is frequently changed during operation of an application program or by the operating system.”); *id.*, 7:28-29; *see also* EX1014, 1:30-33, 1:45-47. Thus, it was obvious to implement Kenley-Long to archive both system files and user files.

14. Claim 20

[20] The method of claim 15, wherein the archive file comprises at least some data of the operating file.

129. Long teaches the “before image” data (i.e., the archive file) is “[t]he data to be modified” and therefore comprises at least some data of the operating file, as claimed. EX1004, 7:37-40; *supra* Section VIII.C.

15. Claim 23

[23] The method of claim 15, wherein the temporary storage location is located in a storage device internal to the computing device.

130. Long teaches a “semiconductor memory buffer” internal to the processing unit 14 of the computer system 10, thus at least rendering obvious temporary storage in a storage device internal to the computing device. EX1004, 7:43-46; *id.*, 2:45-50; *see also* EX1014, 6:3-4 (“system memory”), 6:64 (“core memory”).

16. Claim 24

[24] The method of claim 23, wherein the temporary storage location comprises a buffer located in the storage device.

131. Long teaches a memory buffer as its temporary storage location. EX1004, 7:43-46; *supra* Section VIII.C.1 (Elements [1.3]-[1.4]).

17. Claim 25

[25] The method of claim 15, wherein the temporary storage location is located in a storage device external to the computing device.

132. Kenley-Long renders obvious an external storage device for the temporary storage location. Kenley discloses a file server with “a hierarchical storage structure” that enables “rapid and transparent data migration.” EX1014, 13:42-44, FIG. 2. The file server includes “RAM 30, magnetic disks 18, and optical disks 22.” *Id.*, 13:48-50. The RAM is used as a “RAM buffer cache” for quickly accessing files. *Id.*, 14:59-68 (describing how files are “also written asynchronously [from the RAM] to magnetic disk, to allow for the possibility that they will be referenced again in the intermediate future”).

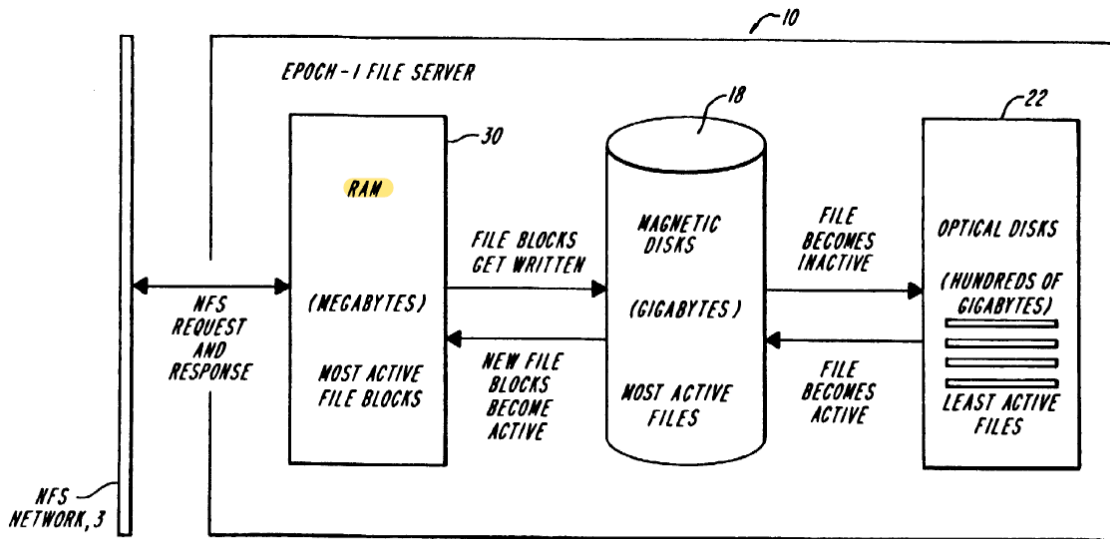


FIG. 2

EX1014, FIG. 2 (highlighting added).

133. It would have been obvious to implement Kenley-Long with a temporary storage location in a storage device external to the computing device, such as within the RAM of the file server disclosed by Kenley, because doing so would reduce contention for storage internal to the computing device by offloading the backup–archive process to a file server accessible over a network. Further, the external storage devices disclosed by Kenley in FIG. 2, as well as others, were well-known and easily implemented by a POSITA. *Supra* Sections VIII.C.2, VIII.C.5 (Claims 2, 5); *see also* EX1005, 5:48-57 (explaining various types of computer-readable media known to and usable by a POSITA), 6:43-57, 5:10-20.

18. Claim 26

[26] The method of claim 15, wherein the first event comprises one of the following: a message from a process that created the archive file from the operating file, a message from the resident program, or a message from a timer.

134. Long teaches various events that trigger identifying the presence of the data in the memory buffer (first events), including “[w]hen the memory buffer is full ... or the processing unit 14 requires memory used in the memory buffer for another purpose.” EX1004, 7:47-53; *supra* Section VIII.C.1 (Claim [1.5]). It was also obvious to use a timer message as the first event. *Supra* Section VIII.C.9.c (Element 14.6). Thus, Kenley-Long renders obvious Claim 26.

19. Claim 27

[27] The method of claim 15, wherein the second event comprises one of the following: a message indicating when the intermediate or the permanent storage location is available, a message indicating when the intermediate or the permanent storage location is inserted, a message indicating high-bandwidth availability, a message from a server associated with the intermediate or the permanent storage location, or a message from a timer.

135. Kenley-Long renders claim 27 obvious, at least because Long teaches a message from a timer, a message indicating when the permanent storage location is available, and a message indicating when the permanent storage location is inserted as second events triggering archiving. EX1004, 8:36-59; *supra* Sections VIII.C.1 (Claim [1.5]), VIII.C.9 (Claim [12.5]).

20. Claim 28

[28] The method of claim 15, wherein the intermediate or the permanent storage location is located in one of a personal attached storage device, a network attached storage device, a Internet storage area network, or a peer-to-peer storage device.

136. Kenley-Long renders obvious permanent storage locations that include those recited in claim 28. *Supra* Sections VIII.C.2-5 (claims 2-5).

21. Claim 29

[29] A non-transitory computer readable medium comprising program instruction for performing the method of claim 15.

137. Long and Kenley each describes implementing its technology as a program (software) run by a computing system, using a processor. *E.g.*, EX1004, 1:67-2:2, *id.*, 45-57; *id.*, 6:60-63 (“memory-resident program”); *id.*, FIGS. 1, 2a; *id.*, 10:10-11:13 (claims 1-15); EX1014, 5:48-6:18, FIGS 1A-3B. Thus, Kenley-Long at least renders obvious implementing the archive system using program instructions on non-transitory computer readable media.

22. Claim 30

[30] The method of claim 15, wherein the intermediate or the permanent storage location is accessible by the computing device through the Internet.

138. Kenley-Long renders this limitation obvious. *Supra* Section VIII.C.5 (Claim 5).

23. Claim 31

[31] The method of claim 15, wherein the second storage location is the permanent storage location.

139. Kenley-Long satisfies this element. *Supra* Section VIII.C.9 (Claim [15.5]).

**IX. GROUND 2: CLAIMS 1-6, 8-20, AND
23-31 ARE OBVIOUS OVER CABRERA (EX1005),
SCHNEIDER (EX1020), AND WHITING (EX1006)**

140. As explained below, claims 1-6, 8-20, and 23-31 are obvious over Cabrera in view of Schneider and Whiting.

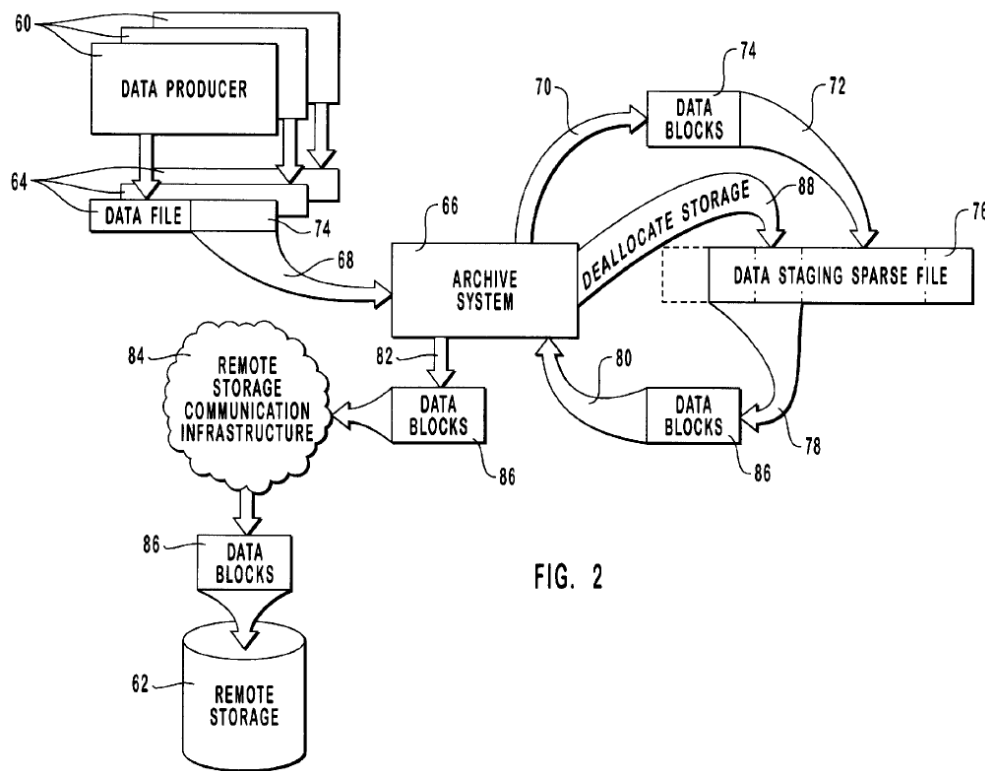
A. Cabrera (EX1005)

141. U.S. Patent No. 5,953,729 (EX1005, “Cabrera”) was filed on December 23, 1997, and granted on September 14, 1999, and I understand that is prior art to the ’348 patent. EX1005, Cover. Cabrera is analogous art to the ’348 patent because it is from the same field of endeavor. Cabrera discloses “systems and methods for transferring or archiving data from a local storage area to a remote storage area.” EX1005, 1:7-12. Thus, the ’348 patent and Cabrera are both within the same field of endeavor of preserving and managing data files.

142. Cabrera discloses “systems and methods for transferring or archiving data from a local storage area to a remote storage area,” and specifically “for temporarily storing or staging data prior to its transfer to remote storage.” EX1005, 1:7-12. Cabrera explains that these systems and methods “minimize the amount of

storage space required for staging data while, simultaneously, minimizing the administrative burden on archive or backup systems.” *Id.*, 2:55-61. These backup–archive systems and methods are file-based because they are applicable to “a particular file,” and duplicate the entire file into a temporary storage area, thereby “allowing access to files while archive or backup copies are created.” *Id.*, 1:51-54; *id.*, 1:64-2:10.

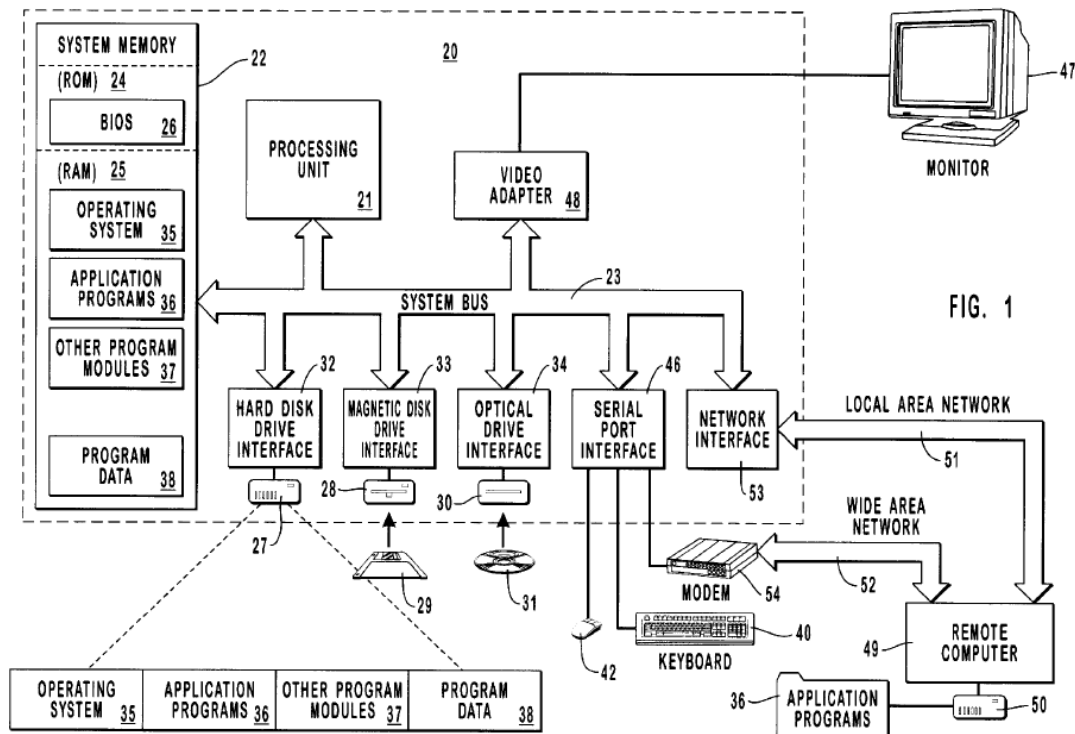
143. Cabrera discloses “data producers” that “create[] data that is to be backed up or archived to a backup or archive storage device.” *Id.*, 6:59-62, FIG. 2. Data produced by data producers is stored as a “data file” in a local storage area. *Id.*, 6:65-7:2. The data file is copied from the local storage area to the temporary, staging storage area, in response to an event. *Id.*, 7:3-20 (disclosing “reading the data from the appropriate location and then storing a copy of the data in the staging storage area”). In particular, the data of the data file is stored in a “data staging sparse file” in the staging storage area. *Id.*, 7:21-46, FIG. 2. Cabrera notes that the systems and methods “will work with any type of file” and are not limited to a particular type of file. *Id.*, 9:38-43. All data of a data file is copied as a sparse file, where any sectors or clusters corresponding to a portion of the data file but not containing data (i.e., “zero data” or “zero clusters”) are not copied. *Id.*, 9:43-10:21, FIG. 4. Instead, file attributes of the sparse file indicate the location of zero clusters with respect to the data file. *Id.*, 10:11-36, FIG. 4.



Id., FIG. 2.

144. Cabrera discloses archive or backup storage as remote storage, which “may comprise any storage medium suitable for such a purpose,” and that “[t]he location of such a storage medium may be local to the backup or archive system or may be remote from the backup or archive system.” *Id.*, 3:3-7. The remote storage could be provided via a remote computer such as “another personal computer, a server, a router, a network PC, a peer device[,] or other common network node.” *Id.*, 6:9-13. Cabrera explains that the local and remote systems may be connected via a local area network (LAN) or wide area network (WAN) as non-limiting examples,

and that “[s]uch networking environments are commonplace in offices[,] enterprise-wide computer networks, intranets and the Internet.” *Id.*, 6:16-21; *id.*, FIG. 1.



EX1005, FIG. 1.

145. Cabrera describes a method to backup or archive data that “begins when sufficient data exists on local storage that should be transferred to the staging area,” or alternatively “on a periodic basis without regard to the amount of data in local storage.” *Id.*, 3:36-38; *id.*, 10:49-55. Data is then “moved from the staging area to remote storage.” *Id.*, 3:53-56. Cabrera discloses “means for transferring data from the staging storage area to a remote storage medium” that “may be implemented by any mechanism capable of retrieving data ... and either directly delivering the data

to remote storage 62 or to an intermediate system or subsystem which will, in turn, deliver the appropriate data to remote storage.” *Id.*, 7:49-60.

146. Cabrera teaches a variety of events that trigger steps in the backup or archive method. A first triggering event for copying data to the staging area includes “the accumulation of a given amount of data, the expiration of a given time delay, the receipt of a command to stage data, or any other triggering event.” *Id.*, 10:53-60. A second triggering event for initiating connection to remote storage “may be one of several things,” such as “a command received from an outside source,” “the expiration of a fixed time,” or “the presence of a certain amount of data in the staging file.” *Id.*, 11:10-22. Cabrera also discloses that “any triggering event” may be used to determine whether to copy additional data to the staging area. *Id.*, 11:22-27. Cabrera broadly discloses “means for monitoring when a pre-determined event occurs” and provides an example “event monitor” which may be “implemented in a wide variety of ways” in order “to watch for desired triggering events and to initiate appropriate action when the events occur.” *Id.*, 12:62-13:13.

B. Schneider (EX1020)

147. Schneider (EX1020) published March 11, 1999, and thus qualifies as prior art under at least Sections 102(a-b).

148. Schneider is analogous art to the ’348 patent because it is from the same field of endeavor. The ’348 patent describes its “Field of Invention” as “file

preservation, file capture, file management and file integrity techniques in computing devices.” EX1001, 1:12-16. Schneider similarly relates to “the backup and recovery of data stored by a digital computer.” EX1020, 1:15-17. Thus, the ’348 patent and Schneider are both within the same field of endeavor of preserving and managing data files.

149. Schneider teaches maintaining the current version of a file on the user’s local disk, i.e., the “internal disk,” while maintaining historic data on a separate “external disk” remotely accessible via a network. EX1020, 112:26-114:7. To maintain such historic versions of a file, Schneider teaches “fundamental methods of recording the original state of information prior to its being altered,” which includes “automatically saving the prior states of altered data on their hard disk” by “first reading data about to be overwritten and saving it in a disk-based history buffer.” *Id.*, 34:12-13; *id.*, 34:29-35:4.

C. Whiting (EX1006)

150. U.S. Patent No. 5,778,395 (EX1006, “Whiting”) was filed on October 23, 1995, and granted on July 7, 1998, and I understand that is prior art to the ’348 patent. EX1006, Cover.

151. Whiting is analogous art to the ’348 patent because it is from the same field of endeavor. Whiting discloses a “system for backing up files from disk volumes on multiple nodes of a computer network to a common random-access

backup storage means.” *Id.*, Abstract; *id.*, 1:5-8. Thus, the ’348 patent and Whiting are both within the same field of endeavor of preserving and managing files.

152. Whiting explains that “[a]s part of the backup process, duplicate files ... may be identified across nodes, so that only a single copy of the contents of the duplicate files ... is stored in the backup storage means.” *Id.*, 5:8-12. To that end, Whiting teaches a “backup directory” that “indicat[es] where the data for each [backup data] file is to be found.” *Id.*, 8:46-50; *id.*, 6:23-27 (explaining that “[e]ach client workstation is responsible for copying the backup data to a preassigned location or directory on the file server, as well as for searching the backup ‘database’ to identify duplicate files across users ...”); *id.*, 8:41-13:9 (describing backup directory files). This backup directory data is also reflected in a “global directory database file” that enables searching against all backup files for all users. *Id.*, 20:6-38; *id.*, 20:6-26:36 (describing the global directory database). The global directory database is updated to account for “new and updated files.” *Id.*, 25:46-53.

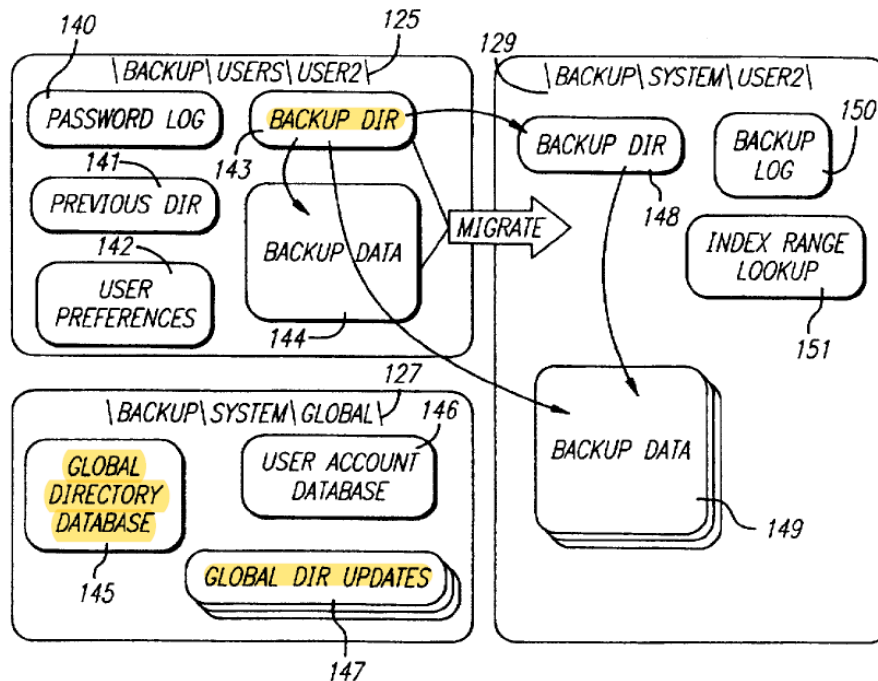


FIG. 3

Id., FIG. 3 (highlighting added).

153. Whiting further discloses that the “backup storage means incorporates hierarchical storage management (HSM), in which files that have not been accessed for a long time are migrated from disk to a secondary storage means, such as tape or optical disk.” *Id.*, 6:52-56.

D. Combining Cabrera, Schneider, And Whiting

154. For at least the reasons discussed herein and below, a POSITA implementing the teachings of Cabrera would have been motivated, with a reasonable expectation of success, to incorporate Schneider’s teachings of performing automatic backup in response to a file operation such as a write operation. Cabrera discloses that “any ... triggering event” may be used to trigger

the backup or archival process, and specifically suggests “the accumulation of a given amount of data, the expiration of a given time delay, [or] the receipt of a command to stage data” as example triggering events. EX1005, 10:56-61. Meanwhile, Schneider teaches performing backups in coordination with file manipulation events, as an improvement over purely time-based backups. EX1020, 3:10-17 (“The key shortcoming of tape backup is that you may not have performed a recent backup and therefore may lose the information or work that was subsequently generated. The present invention addresses this problem by employing a new method of saving changing disk information states providing for a continuously running disk backup system. This method could be implemented on a tape drive, as a tape drive does share the basic random read and write abilities of a disk drive.”); *id.*, 5:24-25 (“The technology of the present invention seeks to eliminate the need to pause and make backups or decide which files should be backed up in the context of short term information recovery.”); *id.*, 12:1-2 (discussing “recording the original state of sectors on a disk prior to being changed”); *id.*, 24:21-23 (“Such an implementation could save an entire prior file state just before a file is opened and the intention to modify is clear....”); *id.*, 23:11-12 (“The present invention's file implementation involves trapping high level (e.g., open and close) requests.”); *see also id.*, 12:5-8, 36:12-17. Looking at Schneider, the POSITA would appreciate the benefits to be gained from initiating each file backup based on

a file manipulation event (such as a save, a write, or a close event) instead of, e.g., a time-based backup. A POSITA therefore would have been motivated to use a file-manipulation event as taught by Schneider as a triggering event to initiate the backup–archival process taught by Cabrera, such that the Cabrera backup–archive process takes place in response to a file modification, because doing so would increase the ease and reliability of backup and archiving by “eliminat[ing] the need to pause and make backups or decide which files should be backed up in the context of short term information recovery.” EX1020, 5:24-25.

155. The POSITA would have had a reasonable expectation of success implementing the teachings of Cabrera in view of these teachings of Schneider because POSITAs knew how to monitor and intercept instructions to perform an operation on an operating file. For example, Schneider discloses “How to Intercept Disk Activity” so that backup copies can be created prior to writing to a disk. EX1020, 30:4-11. As another example, U.S. 5,684,991 (EX1021, “Malcolm”) discloses intercepting “calls to the operating system to write to files,” which may occur “whenever a program wishes to access a file,” in order to “record modifications to files.” EX1021, 4:2-7; *id.*, 20-25 (“The added software has the effect of an instruction to write data being replaced or supplemented by an alternative set of instructions. A similar technique may be used to intercept an instruction to read data from the disk and to replace or supplement this also with an alternative set

of instructions.”). Cabrera itself also discloses monitoring and capturing “changes as they occur,” specifically “changes made to a disk volume in order to allow recovery of the volume should errors occur.” EX1005, 12:21-25. The POSITA thus would have known how to implement Cabrera with the detection of an instruction for performing an operation on an operating file as a triggering event to begin the backup–archival process.

156. A POSITA implementing the teachings of Cabrera also would have been motivated to incorporate the backup directory of Whiting, with a reasonable expectation of success. Specifically, the POSITA would have been motivated to incorporate the backup directory of Whiting into the backup–archival method of Cabrera in order to track “where the data for each file is to be found.” EX1006, 8:45-50. For example, while Cabrera discloses “hierarchical storage of data” (EX1005, 4:45-46) and a “master file table” (MFT) storing the “all resident attributes of a file” (*id.*, 9:9-10), Cabrera does not expressly discuss using a database to track the location of a file as it is migrated through different storage locations. However, as disclosed in Whiting, a POSITA knew that “[t]he main purpose of HSM [hierarchical storage management] is to save on storage costs for very large storage systems by providing the management tools that allow the migration to be transparent to the system, except for the delay in accessing some files.” EX1006, 6:52-60. The POSITA reading Cabrera would have known that the location of a

backup–archive copy needed to be tracked in order to facilitate access to the backup–archive copy, and naturally would have looked to other backup–archive prior art, such as Whiting, for relevant teachings. The global directory index database of Whiting enables such transparent access by storing the location of backup data files. EX1006, 10:52-55, 21:47-49 (describing the record in the global database “which can be used to locate the actual file data contents”). The POSITA implementing Cabrera thus would have naturally used these teachings from Whiting to provide implementation details for storing and updating the location of various archived versions of a file, including as the archive files are migrated from local storage to remote storage, so that the archive files can actually be located and accessed when needed by the system or user.

1. Claim 1

[1.1] A method for archiving files, comprising steps of (a) to (i) following:

157. To the extent the preamble is limiting, Cabrera-Schneider-Whiting discloses it. Cabrera discloses “methods for transferring or archiving data from a local storage area to a remote storage area,” and discusses the archiving of such data in terms of files. EX1005, 1:7-9; *id.*, 9:43-10:21. Whiting discloses that “[b]acking up data and program files (often together referred to as ‘data’ here) from computer disks has been a well[-]known practice for many years.” EX1006, 1:12-14.

[1.2] (a) the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file;

158. Cabrera-Schneider-Whiting renders this limitation obvious. Cabrera discloses that “[a]rchive systems ... typically attempt to maintain a complete history of the changes made to a particular entity, such as a particular file or storage device.” EX1005, 1:32-34. Cabrera further teaches that “[o]ne approach to allowing access to files while archive or backup copies are created is to duplicate the information that will be backed up or archived and ‘stage’ the information in a temporary storage area.” *Id.*, 2:5-9. Cabrera discloses that “any ... triggering event” may be used to trigger the copying of data from local storage to the staging area. *Id.*, 10:56-61. Cabrera also discloses a “data producer 60 [that] creates data that is to be backed up or archived to a backup or archive storage device, such as remote storage 62.” *Id.*, 6:58-62.

159. Schneider teaches, when “a request to write [to the disk] occurs, ... intercepting the write” to trigger copying the file on the disk before completing the write to the disk. EX1020, 12:5-8; *see also id.*, 30:4-6 (explaining “How to Intercept Disk Activity”). Schneider further teaches that “applications” and “an operating system (OS) ... has the responsibility ... to save and recall information from a hard disk,” and that “applications” operating under the OS generate writes “during a write cycle (e.g., the saving of a document from a word processor or the saving of a

photograph from an appropriate editor).” *Id.*, 1:20-22, 32:19-20. The operating system of Schneider was a resident program by definition of the ’348 patent, and the applications are also a resident program because they generate writes. *See* EX1001, 4:49-52 (defining “Resident Program”).

160. A POSITA would have been motivated to use detecting or intercepting “a request to write” to disk as taught by Schneider as a triggering event, because doing so would enable the archiving system to “maintain a complete history of the changes made to ... a particular file.” EX1005, 1:32-34. Specifically, it would have been obvious to use a file manipulation event as taught by Schneider as a triggering event to initiate the backup–archival process taught by Cabrera, such that the Cabrera backup–archive process takes places in response to a file modification, because doing so would increase the reliability of backup and archiving by “providing for a continuously running disk backup system” and “eliminat[ing] the need to pause and make backups or decide which files should be backed up in the context of short term information recovery.” EX1020, 3:12-4, 5:24-25; *supra* Section IX.D.

161. Cabrera-Schneider-Whiting thus renders obvious “the step of detecting an instruction by a resident program in a computing device for performing an operation on an operating file.”

[1.3] (b) the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction;

162. Cabrera-Schneider-Whiting render this limitation obvious. Whiting explains that the most common “reason for backup is that users may inadvertently delete or overwrite important data files.” EX1006, 1:23-29. Cabrera discloses copying a file “from local storage to the staging file” in response to “any ... triggering event.” EX1005, 10:56-61. The staging file or staging area “is adapted for temporarily storing the data.” *Id.*, 7:3-7; *id.*, 2:5-10 (“‘stage’ the information in a temporary storage area”). Cabrera discloses that “[a]ny mechanism that performs this function [of moving data] may be utilized such as, for example, reading the data from the appropriate location and then storing a copy of the data in the staging storage area[, ...] direct memory transfer and so forth.” *Id.*, 7:11-16. As discussed *supra* paragraphs - and -, it would have been obvious to use interception of a request by an application or operating system to write to disk as taught by Schneider as the triggering event to initiate the backup–archive method of Cabrera.

163. Furthermore, a POSITA would have found it obvious to create the copy of the file temporally proximate to the operation being performed “so that a logically consistent backup copy is created” because creating such a copy before overwriting a file was a well-known, fundamental technique. EX1005, 1:51-55 (explaining that

“if a particular file or volume is to be backed up, it is usually important not to allow the contents of the file or volume to change during the backup procedure so that a logically consistent backup copy is created”); *see also* EX1004, 7:28-40 (describing creating a backup copy of data to be modified just before the data is modified); EX1016, 6:33-42, 8:7-18 (describing the “fundamental” “Move Method” of creating a backup copy of a file before overwriting the file). A POSITA would have found this limitation particularly obvious in view of the well-known “copy-on-write” technique in the art, wherein “old data blocks on disk [are copied] to a temporary disk location when updates are made to a file ... that is being backed up.” EX1007, 254 (“Many database companies have copy-on-write functionality built into their products to enable 24x7 operations of their databases.”).

164. Cabrera-Schneider-Whiting thus includes “the step of creating an archive file from the operating file and storing the archive file in a temporary storage location temporally proximate to the operation being performed on the operating file and responsive to detecting the instruction.”

[1.4] (c) the step of identifying presence of the archive file in the temporary storage location responsive to the occurrence of a first event;

165. Cabrera discloses this limitation. Cabrera teaches monitoring the “amount of data in the staging area ... in order to identify when a backup or archive session should be initiated.” EX1005, 3:45-47. Cabrera further teaches an “event

monitor 152” that “may be implemented in a wide variety of ways” in order “to watch for desired triggering events and to initiate appropriate action when the events occur.” EX1005, 13:1-10. Cabrera also teaches determining “whether more data should be appended to the staging file,” where “[t]his determination may be made based on any triggering event.” *Id.*, 11:22-27. A POSITA would have understood that evaluating whether to append more data to the staging file necessarily requires identifying the presence of the data in the staging file, and that this identifying step would occur responsive to the triggering event.

[1.5] (d) the step of transmitting the archive file to a second storage location responsive to a second event, the second storage location being an intermediate or a permanent storage location;

166. Cabrera teaches transmitting the archive file to a second storage location responsive to a second event. For example, Cabrera teaches, “[w]hen the second triggering event occurs and it is time to establish a connection to remote storage ... the data should be transferred from the staging file to remote storage.” EX1005, 11:46-48; FIG. 5. For purposes of claim 1, the remote storage of Cabrera is a permanent storage, and thus a second storage location, because it is a persistent “backup or archive storage device.” EX1005, 6:61-62. The second event could include the “second triggering event” which “initiate[s] connection to remote storage” so that the file can be transferred (EX1005, 11:28-31), or an indication that the remote storage is connected. *Id.*, 11:32-45 (explaining that to “establish[] a

connection to the remote storage ... [t]he mechanism used will depend on the type of remote storage used”).

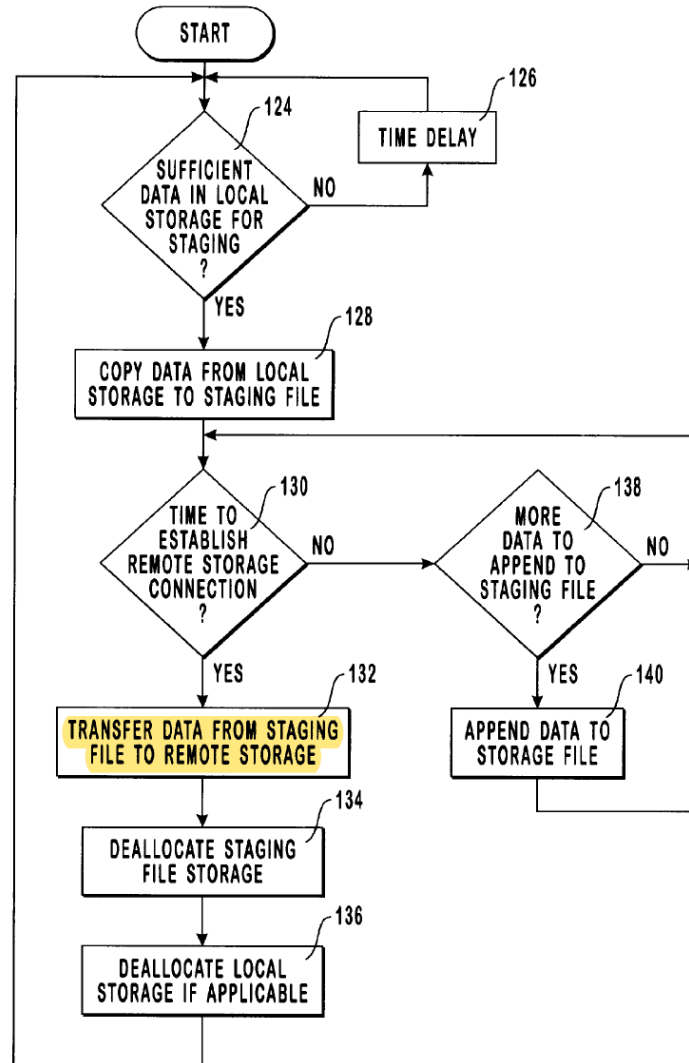


FIG. 5

EX1005, FIG. 5 (highlighting added).

[1.6] (e) after storing the archive file in the temporary storage location, updating a database to indicate that the archive file is located in the temporary storage location;

167. Cabrera-Schneider-Whiting renders this limitation obvious. Cabrera discloses deallocating storage space in the local storage area after migrating the data to the staging area. EX1005, 14:9-15; FIG. 6. Whiting discloses a “backup directory file” that “indicate[s] the directory structure of the source disk volume, as well as pointers into the backup data files ... indicating where the data for each file is to be found.” EX1006, 8:45-50.

168. It would have been obvious, after storing the backup data file in the staging area, to update the backup directory file (database) taught by Whiting to indicate that the backup data file is located in the staging area, and thus track the location of the data during the backup–archive process with a backup directory as taught by Whiting so that the data could be retrieved, for example, for transferring the data or using the data for recovery. *Supra* Section IX.D.

[1.7] (f) determining a final destination for the archive file;

169. Cabrera-Schneider-Whiting discloses this limitation. Cabrera discloses transferring backup data “from the staging file to remote storage.” EX1005, 11:46-49. Cabrera further explains that “a file stores data in a sequence of allocation units called clusters” which are enumerated through “virtual cluster numbers (VCNs)” in the NT file system. *Id.*, 9:44-54. Each VCN of a file “maps to a corresponding logical cluster number (LCN), which identifies the disk location of the cluster.” *Id.*, 9:55-62; *id.*, FIG. 4 (depicting VCNs and LCNs). A POSITA thus would have found it

obvious to determine a final destination in the remote storage for the archive file so that the archive file could be written to a specific physical location within the remote storage. EX1005, 11:49-55 (explaining that transferring the file to remote storage includes “writing data to a ... disk”).

[1.8] (g) moving the archive file from the temporary storage location to an intermediate storage location;

170. Cabrera-Schneider-Whiting discloses this limitation. Cabrera discloses that transferring data from the staging file to remote storage may be implemented by “delivering the data ... to an intermediate system or subsystem which will, in turn, deliver the appropriate data to remote storage.” EX1005, 7:52-60. This “intermediate system,” which is part of a “remote storage communication infrastructure” taught by Cabrera, is “an intermediate storage location” between the staging area and the remote storage. EX1005, 7:52-60; *id.*, 8:5-11; FIG. 2.

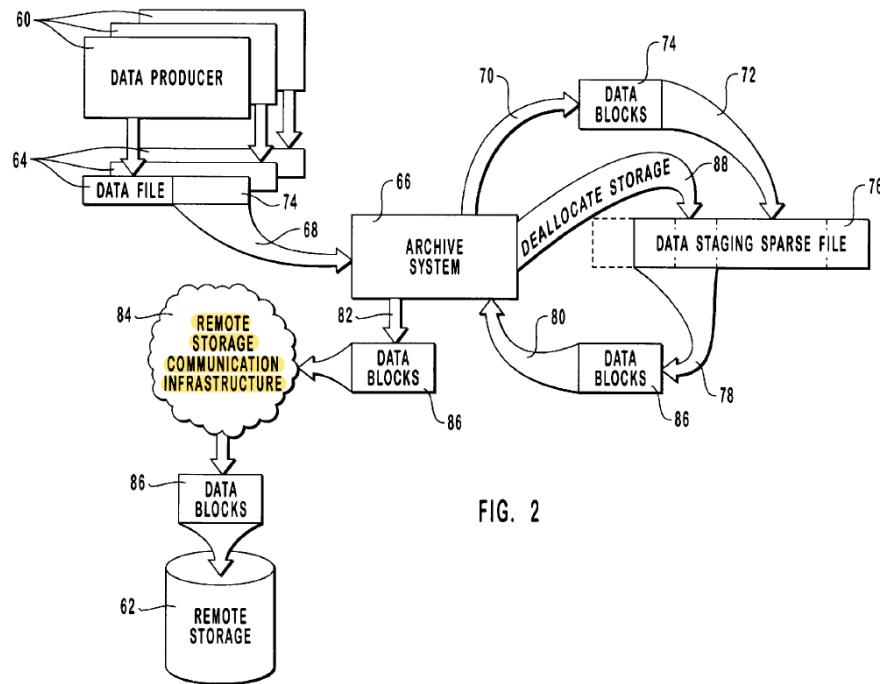


FIG. 2

EX1005, FIG. 2 (highlighting added).

[1.9] (h) updating the database to indicate the archive file is located in the intermediate storage location; and

[1.10] (i) after moving the archive file to the second storage location, updating the database to indicate the archive file is located in the second storage location.

171. As discussed above with regard to Element [1.6], Cabrera-Schneider-Whiting renders these limitations obvious. *Supra* Section IX.D.1.

2. Claim 2

[2] The method of claim 1, wherein the second storage location includes a personal attached storage device.

172. Cabrera discloses this limitation. Cabrera discloses that “[i]n some embodiments ... remote storage 62 may be directly attached to the computer system where archive system 66 resides.” EX1005, 7:63-66; *id.*, 6:49-52 (“For example,

backup or archive storage which is directly attached to a particular computer system will also be referred to as remote storage even though the storage is not located at a remote location.”).

3. Claims 3-5

[3] The method of claim 1, wherein the second storage location includes a network attached storage device.

[4] The method of claim 1, wherein the second storage location includes a peer-to-peer storage device.

[5] The method of claim 1, wherein the second storage location includes an internet storage area network.

173. Cabrera-Schneider-Whiting render these claims obvious. Cabrera teaches “remote storage” for “backup or archive storage” that “may indeed be located at a remote location” such that “establishing a connection to the remote storage ... may involve establishing connections over networks, dial-up connections, connections through other computer systems, and so forth.” EX1005, 11:32-45; *id.*, 49-55 (“The exact mechanism used to transfer the data ... may require transferring data over various networks or via various computer systems or other intermediate devices to remote storage.”). Cabrera discloses a “networked environment” including a “memory storage device” of a remote computer device logically connected to a computer via a network. *Id.*, 6:9-22; *id.*, FIG. 1. Cabrera also discloses “a peer device” including a “memory storage device” as the remote storage. *Id.*, 6:9-16. Cabrera further discloses a “[r]emote computer 49 [that] may be another personal

computer, a server, a router, a network PC, a peer device or other common network node” including “a memory storage device 50,” and connected to a computer in a networked environment such as “the Internet,” as the remote storage. *Id.*, 6:9-22.

174. Cabrera thus discloses remote storage as “a network attached storage device,” a “peer-to-peer storage device,” and “an internet storage area network” as defined by the ’348 patent. *Compare* EX1005, 6:9-22, *with* EX1001, 4:34-40 (defining “Network Attached Storage Device” as “any storage device connected directly to a network to which a first computing device is also temporarily or permanently connected, or any storage device connected to a second computing device that is also temporarily or permanently connected to the network to which the first computing device is temporarily or permanently connected”), 4:45-48 (defining “Peer-to-Peer Storage Device” as “any storage area (device, collection of devices, etc.) that can be accessed by the computing device when it is sharing resources with other network or Internet accessible computers”), 4:41-44 (defining “Internet storage area network” as “any storage area (device, collection of devices, etc.) that can be accessed by the computing device when the computing device is temporarily or permanently connected to the Internet”).

4. Claim 6

[6] An article of manufacture comprising a computer usable medium having computer readable program code for performing the method of claim 1.

175. Cabrera discloses this limitation. Cabrera discloses implementing the backup and archival methods in “the general context of computer-executable instructions, such as program modules, being executed by a personal computer.” EX1005, 5:4-6; *id.*, 15:11-42 (claim 2) (reciting a “computer program product ... comprising a computer readable medium for carrying executable instructions that are used to implement ... a method for transferring the data from the local storage area to the remote storage area ...”).

5. Claim 8

176. Elements [8.1]-[8.6] are identical to Elements [1.1]-[1.6], and are satisfied by Cabrera-Schneider-Whiting for the same reasons explained above for Elements [1.1]-[1.5]. *Supra* Section IX.D.1.

[8.7] (f) the step of moving the archive file from the temporary storage location to an output buffer;

[8.8] (g) the step of updating the database to indicate that the archive file is located in the output buffer;

177. Cabrera-Schneider-Whiting satisfies these limitations, as it renders obvious moving the copied file (archive file) from the staging area (temporary storage location) to the remote storage communication infrastructure (intermediate storage location), and updating the global database to indicate the location of the copied file in the remote storage communication infrastructure. *Supra* Section IX.D.1. A POSITA would have understood that the remote storage communication

infrastructure could include an output buffer (in an “intermediate system or subsystem”) to facilitate transferring data from the staging area to the remote storage.

[8.9] (h) the step of transmitting the archive file from the output buffer to the intermediate or the permanent storage location; and

[8.10] (i) the step of updating the database to indicate that the archive file is located in the intermediate or the permanent storage location.

178. Cabrera-Schneider-Whiting satisfies this element, as it renders obvious transmitting the copied file (archive file) from the remote storage communication infrastructure (output buffer) to the remote storage (permanent storage location), and then updating the global directory database (database) to indicate that the copied file is in the remote storage. *Supra* Section IX.D.1 (Elements [1.5], [1.10]).

6. Claims 9-11

179. Claims 9-11 are identical to claims 2-3 and 5, respectively (other than the independent claim cited), and are obvious over Cabrera and Whiting for the same reasons explained above for claims 2-3 and 5. *Supra* Sections IX.D.2, IX.D.3.

7. Claims 12-15

180. Elements [12.1]-[12.5], [13.1]-[13.5], [14.1]-[14.5], and [15.1]-[15.5] are identical to one another and to elements [1.1]-[1.5], and are satisfied by Cabrera and Whiting for the same reasons as explained above for Elements [1.1]-[1.5]. *Supra* Section IX.D.1.

a) **Element [12.6]**

[12.6] wherein the second event includes a message indicating when the second storage location is available.

181. Cabrera discloses “establish[ing] a remote storage connection” in order to transfer the data to the remote storage. EX1005, 11:46-49. A POSITA would have understood that the transmission of the data to the remote storage taught by Cabrera is in response to a message indicating that the connection to the remote storage has been established.

b) **Element [13.6]**

[13.6] wherein the second event includes a message from a timer.

182. Cabrera discloses “the expiration of a fixed time” as a triggering event for establishing a connection to remote storage. EX1005, 11:10-20.

c) **Element [14.6]**

[14.6] wherein the first event includes a message from a timer.

183. Cabrera discloses determining whether to append more data to the staging file “based on any triggering event,” such as the expiration of a timer. EX1005, 11:14-27. As discussed with regard to Claim 1.5, determining whether to append more data to the staging file involves identifying the presence of the data in the staging file. *Supra* Section IX.D.1.

d) **Element [15.6]**

[15.6] wherein the first event is different from the second event.

184. As discussed with regard to Claim 1, the first and second events taught by Cabrera are separate events. *Supra* Section IX.D.1.

8. **Claim 16**

[16] The method of claim 15, wherein the resident program is separate from an operating system.

185. Cabrera-Schneider-Whiting renders this limitation obvious. Cabrera-Schneider-Whiting renders obvious claim 15. *Supra* Section IX.D.7.d. Cabrera discloses “an operating system 35, one or more application programs 36, [and] other program modules 37,” and a POSITA would have understood that the “data producer” of Cabrera would include any of these elements. EX1005, 5:58-62, 6:58-66.

9. **Claim 17**

[17] The method of claim 16, where the resident program comprises a program having control over file operations on one or more operating files.

186. Cabrera-Schneider-Whiting renders this claim obvious. Cabrera discloses or renders obvious claim 16. *Supra* Section IX.D.8. A POSITA would have understood that the data producer of Cabrera comprises a program having control over file operations on one or more operating files. EX1005, 5:58-62; *id.*, 6:58-66; *supra* Section IX.D.1 (Claim [1.2-3]).

10. Claim 18

[18] The method of claim 15, wherein the operation comprises one of read, write, save, rename, delete, copy, move, open, close, or create operations.

187. Cabrera-Schneider-Whiting renders this claim obvious because the file-manipulation operation taught by Schneider is a “write” operation. EX1020, 12:5-8; *supra* Section IX.D.7.d (Claim 15 obvious).

11. Claim 19

[19] The method of claim 15, wherein the operating file comprises one of a system file or a user file.

188. Cabrera-Schneider-Whiting renders this claim obvious. *Supra* Section IX.D.7.d; EX1005, 6:66-67 (“Data file 64 represents a local storage area used by data producer 60 to store data it produces.”). Cabrera discloses archiving a log file, which is a system file. EX1005, 12:37-40 (discussing archiving a log file); EX1008, 38 (explaining that “[t]he log file is a system file”). Cabrera further discloses files with both system attributes and user attributes. EX1005, 8:50-9:3.

12. Claim 20

[20] The method of claim 15, wherein the archive file comprises at least some data of the operating file.

189. Cabrera-Schneider-Whiting renders this claim obvious. *Supra* Section IX.C.9.d. A POSITA would have found it obvious that the archive file comprises at

least some of the operating file in order to backup or archive the file for later recovery. *Supra* Section IX.D.1 (Elements [1.2-3]).

13. Claim 23

[23] The method of claim 15, wherein the temporary storage location is located in a storage device internal to the computing device.

190. Cabrera-Schneider-Whiting renders obvious this limitation. *Supra* Section IX.D.7.d. Cabrera discloses that “[a]ny type of technology may be utilized to implement” the staging area, where the “staging storage area ... is adapted for temporarily storing the data.” EX1005, 7:7-11; *id.*, 7:40-41. Cabrera discloses a variety of storage devices internal to the computing device. *Id.*, 5:21-57; *id.*, FIG. 1. A POSITA would have found it obvious to implement the staging area (temporary storage location) in a storage device internal to the computing device because Cabrera discloses staging data “prior to transfer to remote storage.” *Id.*, 13:32-34.

14. Claim 24

[24] The method of claim 23, wherein the temporary storage location comprises a buffer located in the storage device.

191. Cabrera-Schneider-Whiting renders obvious this limitation. *Supra* Section IX.D.13. A POSITA would have understood that a buffer is appropriate for staging the data prior to transfer because “a buffer is a temporary storage location for information being sent or received.” EX1009, 132; *see also* EX1019, 56 (defining “buffer” as “[a]n intermediate repository of data—a reserved portion of

memory in which data is temporarily held pending an opportunity to complete its transfer to or from a storage device or another location in memory”).

15. Claim 25

[25] The method of claim 15, wherein the temporary storage location is located in a storage device external to the computing device.

192. Cabrera-Schneider-Whiting renders this limitation obvious. *Supra* Section IX.D.7.d. Cabrera discloses a variety of storage devices “which can store data that is accessible by a computer.” EX1005, 5:50-57. Cabrera further discloses that “[a]ny type of technology may be utilized to implement” the staging area, where the “staging storage area ... is adapted for temporarily storing the data.” *Id.*, 7:7-11; *id.*, 7:40-41. Cabrera also discloses that “the data may be copied or moved from local storage to a staging area,” indicating that the staging area may not be a part of the computer’s local memory. *Id.*, 3:41-42 (emphasis added). A POSITA therefore would have been motivated to implement the staging area (temporary storage area) in a storage device external to the computing device in order to free up storage space in local storage. *Id.*, 3:61-67 (describing deallocating “storage space from the local storage area ... once the data has been copied to the staging area”). The POSITA would have had a reasonable expectation of success implementing the staging area in a storage area external to the computing device because Cabrera expressly discloses storage areas for the backup–archive process that are “separate and apart

from the local storage volumes typically utilized by a computer system” (e.g., external to the computing device). *Id.*, 2:66-3:1.

16. Claim 26

[26] The method of claim 15, wherein the first event comprises one of the following: a message from a process that created the archive file from the operating file, a message from the resident program, or a message from a timer.

193. Cabrera-Schneider-Whiting renders obvious this limitation. *Supra* Section IX.D.7.d. As discussed with regard to Element 1.4, Cabrera discloses identifying the presence of data in the staging file responsive to “any triggering event.” EX1005, 11:22-27; *supra* Section IX.D.1 (Claim [1.4]). Cabrera discloses triggering events that include “the expiration of a given time delay.” EX1005, 10:56-60. A POSITA therefore would have found it obvious, with a reasonable expectation of success, to implement Cabrera where the first event comprises a message from a timer because Cabrera provides a finite list of triggering events and indicates that any one of them would be suitable.

17. Claim 27

[27] The method of claim 15, wherein the second event comprises one of the following: a message indicating when the intermediate or the permanent storage location is available, a message indicating when the intermediate or the permanent storage location is inserted, a message indicating high-bandwidth availability, a message from a server associated with the intermediate or the permanent storage location, or a message from a timer.

194. Cabrera-Schneider-Whiting renders obvious this limitation. *Supra* Section IX.D.7.d. Cabrera discloses using “a second triggering event” to trigger transferring the archive file from the staging area to the remote storage, and explains that “this triggering event may be one of several things” such as “a command received from an outside source” or “the expiration of a fixed time.” EX1005, 11:10-16. A POSITA would have understood that “the expiration of a fixed time” is “a message from a timer.”

18. Claim 28

[28] The method of claim 15, wherein the intermediate or the permanent storage location is located in one of a personal attached storage device, a network attached storage device, a Internet storage area network, or a peer-to-peer storage device.

195. Cabrera-Schneider-Whiting renders this claim obvious. *Supra* Section IX.D.7.d. Cabrera discloses that the permanent storage location is located in one of a personal attached storage device, a network attached storage device, an Internet storage area network, or a peer-to-peer storage device. *Supra* Sections IX.D.2-3.

19. Claim 29

[29] A non-transitory computer readable medium comprising program instruction for performing the method of claim 15.

196. Cabrera-Schneider-Whiting renders this claim obvious. *Supra* Section IX.D.7.d. Cabrera also discloses “[e]mbodiments ... includ[ing] computer readable media having executable instructions ... stored thereon.” EX1005, 4:50-67. Whiting

discloses implementing the method as “backup software installed” on a computer.
EX1006, 35:8.

20. Claim 30

[30] The method of claim 15, wherein the intermediate or the permanent storage location is accessible by the computing device through the Internet.

197. Cabrera-Schneider-Whiting renders obvious this limitation. *Supra* Section IX.D.7.d. Cabrera discloses a “networked environment” including a “memory storage device” of a remote computer device logically connected to a computer via a network, such as the Internet. EX1005, 6:9-22; *id.*, FIG. 1.

21. Claim 31

[31] The method of claim 15, wherein the second storage location is the permanent storage location.

198. Cabrera-Schneider-Whiting renders obvious this limitation. *Supra* Section IX.D.7.d. The second storage location of Cabrera is the permanent storage location. *Id.*

I, David Maier, do hereby declare and state, that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: 31 October, 2023

David Maier
Signature

APPENDIX A

DAVID MAIER

Department of Computer Science
Portland State University
PO Box 751
Portland, Oregon 97207-0751
(503) 725-2406
maier@cs.pdx.edu

Born

2 June 1953, Eugene, Oregon

Education

B.A., Honors College	University of Oregon, 1974 Double major: Mathematics and Computer Science
Ph.D.	Princeton University, 1978 Electrical Engineering and Computer Science

Positions Held

2004 - present	Maseeh Professor of Emerging Technologies Department of Computer Science Portland State University
2004 - present	Professor (joint appointment) Environmental and Biomolecular Systems Oregon Graduate Institute
2009	Visiting Research (Jan. - Mar.) Microsoft Research, Redmond, Washington
2007 - 2008	Visiting Researcher (Dec. - Mar.) Microsoft Research, Redmond, Washington
2006 - present	Visiting Professor Department of Medical Informatics & Clinical Epidemiology Oregon Health & Science University
1988 - 2006	Professor (joint appointment as of 2004) Department of Computer Science and Engineering Oregon Graduate Institute
1997 - 1998	Visiting Professor Computer Sciences Department, University of Wisconsin
1989 - 1990	Visiting Scientist GIP Altair, INRIA-Rocquencourt, France (Institute National de Recherche en Informatique et en Automatique)
1988	Acting Chair (Jan. - June)
1983 - 1988	Associate Professor
1982 - 1983	Assistant Professor Department of Computer Science and Engineering Oregon Graduate Institute
1978 - 1982	Assistant Professor Department of Computer Science State University of New York at Stony Brook

AWARDS AND HONORS

1. NSF Presidential Young Investigator Award, *Foundations of Knowledge Management Systems*, IST 83 51730, June 1984 - May 1985, renewed to May 1989. Industrial sponsors: Tektronix Foundation, Intel, Digital Equipment, Servio Logic, Mentor Graphics, Xerox, Beaverton Area Chamber of Commerce.
2. Distinguished Teaching Award, Oregon Graduate Institute, 1985-1986, 1987-1988, 1993-1994.
3. President's Award for Research Excellence, Oregon Graduate Institute, 1988-1989.
4. Outstanding Teacher Award CSE, OGI Student Council, 1987-1988, 1988-1989, 1990-1991, 1993-1994, 1994-1995.
5. 1997 SIGMOD Innovations Award.
6. ACM Fellow, 1998.

GRANTS AND CONTRACTS

1. Department of Energy Emergency Information Management System (EEMIS), *Logical Data Base Design for EEMIS*. Principal Investigator, Jack Heller. February - August 1979. Supervised by Brookhaven National Laboratory.
2. University Awards Committee Fellowship, State University of New York, *Applications of Minimal Covers in Relational Databases*. June - July 1979.
3. NSF, *The Relational Theory of Databases: Research Initiation*, IST 79 18264. September 1979-February 1982.
4. NSF, *Representing Semantic Information within Databases*, IST 81 04834, with David S. Warren and Sharon C. Salveter. May 1981 - October 1983.
5. NSF, *Verbs in Databases*, IST 82 14622, with Sharon C. Salveter, October 1982 -September 1984.
6. NSF, *Theory of Database Systems*, MCS 82 07216, November 1982 - October 1984.
7. Tektronix Computer Research Laboratory, *Data Models for Networked Workstations*, November 1983 - September 1986.
8. Tektronix Information Display Group (Artificial Intelligence Machines), *Smalltalk Short Course*, November 1984 - March 1985.
9. Microelectronics and Computer Technology Corporation (MCC), *Update in the Logical Data Language*, November 1985 - January 1987.
10. NSF, *Automatic Generation of Interactive Displays*, IST 86 04977, with Alan H. Borning and Ralph London, June 1986 - November 1988 (separate grant to Washington).
11. Apple Computer, Graduate Student Support in Object-Oriented Databases, 1988-1989.
12. NSF, *Generating Interactive Displays from Declarative Specifications*, IRI 88 05564, with Alan H. Borning, June 1988 - November 1990 (separate grant to Washington).
13. NSF, *Query Optimization in Object-Oriented Database Management Systems with Encapsulated Behavior*, IRI 89 20642, with Goetz Graefe, September 1989 - February 1991 (separate grant to Colorado).
14. Battelle Pacific Northwest Laboratories, *Establishment of Affiliated Laboratory for Computation Science*, 097714-A-L2, with David Novick, Steve Otto, Jonathan Walpole, Mike Wolfe, January 1991 - September 1991.
15. Battelle Pacific Northwest Laboratories and OACIS, *Database Support for Computational Chemistry*, 097714-A-L2, #5, March - September 1991.
16. Sequent Computer Systems, *Scientific Database Systems*, equipment donation, July 1991.
17. NSF, *Database Support for Scientific Computing*, IRI 91 17008, with Michael Wolfe, James Stanley and Jonathan Walpole, September 1991 - February 1994.

18. DARPA, *Architectures for Query Processing in Persistent Object Bases*, ARPA order #8230, CECOM contract DAAB07-91-C-Q518, with Michael Carey, David DeWitt and Stanley Zdonik, September 1991 - September 1994. Extended to March 1996.
19. OACIS, *Graduate Fellowship: Desktop Material Scientist*, James Stanley, PI, January - December 1992.
20. Battelle Pacific Northwest Laboratories and OACIS. *Computational Proxies: Controlling and Capturing Computational Chemistry Experiments*, 097714-A-L2, #5, rev. 2 with Judy Cushing, January - December 1992.
21. U.S. West Advanced Technologies, *Graduate Student Support in Object Oriented Database Technology*, 1992.
22. NSF, *Collaborative Research: The REVELATION Project: Query Processing in Behavioral Object-Oriented Database*, IRI-9118360, with Goetz Graefe, September 1992-February 1995 (separate grant to Portland State)
23. DOD AASERT Award, *Query Processing in Loosely-Coupled Object Bases*, P-3223-RT-AAS, June 1993-May 1996.
24. Pacific Power and Light and OACIS, *The Capture and Analysis of Application Semantics*, with Lois Delcambre and Leonard Shapiro, April 1993-March 1994.
25. NSF, *Supporting Predictable Latency and Throughput in Large Storage Systems*, IRI-9223788, with Jonathan Walpole, PI, August 1993-January 1995.
26. Battelle Pacific Northwest Laboratories, *Computational Chemistry Database*, Task order 248003, Contract 206010-A-L2, with Judy Cushing, July-December 1993.
27. Battelle Pacific Northwest Laboratories, *Multidisciplinary Scientific Database*, Task Order 246273, Contract 206010-A-L2, with Lois Delcambre, April - September 1994.
28. Battelle Pacific Northwest Laboratories, *HEIS Data Management*, Task Order 266643, Contract 206010-4-L2, with Lois Delcambre, Tim Sheard and Andrew Black, August - September 1994.
29. DOD, *Quality of Service Specification and Use in Massive Digital Data Systems*, MDA904-95-C-5547, with Calton Pu and Jonathan Walpole, March 1995 - February 1998.
30. NSF, *An Effective Framework for Processing OODB Languages*, IRI-9509955, with Leonidas Fegaras, September 1995 - August 1998.
31. NSF, *SGER: Content-Based Connections for Navigating on the NII*, IRI-9502084, with Lois Delcambre, July 1995 - June 1996.
32. ONR, *Control and Scaling of Distributed Agent Systems*, N00014-95-1-1164, with Philip Cohen, Sharon Oviatt, David Novick, Jonathan Walpole and Calton Pu, May 1995 - April 1998.
33. Rik Smoody, *Gift to Support Purchase of Workstation*, with Jonathan Walpole, 1995.
34. Goetz Graefe and Microsoft, *Gift to Support Research and Education*, 1995.
35. Intel, *QoS-Based Adaptive Resource Management for Multimedia Systems*, with Jonathan Walpole and Calton Pu, 1996, 1997.
36. DARPA, *Heterodyne: A Regenerative Receiver for Dynamic Specialization Technology*, F30602-96-1-0302, with Andrew Black, Charles Consel, Lois Delcambre, Calton Pu and Jonathan Walpole, September 1996 - August 1999.
37. DOE, *Information Technology for Coordinated Long Term Care*, subcontract from OHSU 6398171 (DOE F603-94ER61918/A004), with Lois Delcambre, October 1996 - September 1997.
38. NSF, *Enhancing Researcher Site Productivity at Shared Research Facilities: Database Tools for Analyzing Forest Canopy Structure Data*, subcontract from The Evergreen State College on grant 96-30316, with Lois Delcambre, September 1996 - August 1997.

39. NSF, *Query Optimization Engineering*, IRI-9619977, Companion proposal to one by Len Shapiro of PSU, September 1997 - August 2000.
40. DARPA, *Systemic Quality of Service Support for Adaptive Distributed Systems*, N66001-97-C-8522, with Jonathan Walpole, PI, Calton Pu, Dylan McNamee, July 1997 - June 2000.
41. DARPA, *Broadcast Objects for Effective Data Dissemination in BADD*, F30602-97-2-0241, subcontract from Brown University, Stanley Zdonik, PI, July 1997 - November 1998.
42. NSF, *A Wide-Spectrum Infrastructure for Software Research* (Research Infrastructure Award), CDA-9703218, Andrew Black and Calton Pu, PIs, 6 Co-PIs, August 1997 - July 2002.
43. DARPA, *Robust Agent-based Systems Incorporating Teams of Communicating Agents*, F30602-98-2-0098, with Philip Cohen, PI, Calton Pu and Ling Liu, July 1998 - September 2002.
44. NSF/DLI2, *Tracking Footprints Through an Information Space: Leveraging the Document Selections of Expert Problem Solvers*, IIS-9817492, with Paul Gorman (OHSU) and Lois Delcambre, PIs, January 1999 - December 2002.
45. DARPA, *NIAGARA – An Excursion into Net Data Management*, with David DeWitt (Wisconsin) and Jeffrey Naughton, PIs, N66001-99-1-8908, June 1999 - June 2002.
46. NSF, *RUI: Databases and Database Tools for Canopy Science – What Trees Can Teach Us about Integrating Database Use in the Research Process*, with Nalini Nadkarni (Evergreen) and Judy Cushing (Evergreen), PIs, and Lois Delcambre, Co-PI, DBI-9975510, June 1999 - June 2002.
47. DARPA, *Autonomic Information Assurance program, Autonomix: Component, System and Network Autonomy*, with C. Cowan (Wirex) and L. Delcambre, DAAH01-99-C-R206, September 1999 - October 2002.
48. NSF, *Digital Government, Harvesting Information to Sustain Our Forests*, with L. Delcambre, F. Phillips (MST) and P. Toccilino (ECE). In cooperation with Tim Tolle (US Forest Service), Eric Landis (consultant) and Craig Palmer (UNLV), EIA-9983518 plus REU and technology transfer supplements, August 2000 - July 2003.
49. NSF ITR, *A Petabyte in Your Pocket*, with D. DeWitt (Wisconsin) and J. Naughton (Wisconsin), IIS 0086002, September 2000 - August 2005.
50. NSF, *Digital Government: Workshop on Biodiversity Informatics*, EIA-0084541, September 2000 - April 2001.
51. NIH, *National Library of Medicine, Training Program in Health Informatics/Biomedical Information Science and Technology Initiative (BISTI)*, with Christopher Dubay (OHSU), July 2000 - June 2001. Supplement to existing training grant, Kent Spackman (OHSU), PI.
52. NSF ITR, *Quality-Scalable Information Flow Systems for Environmental Observation and Forecasting*, with A. Baptista (ESE), PI, W.-C. Feng, T. Leen, and J. Walpole, ACI 0121475, September 01 - September 06. REU Supplement, Summer 2004.
53. NSF, *DELOS/NSF Study Panel on Information Extraction for Digital Libraries*, IIS 0226501, August 02 - August 03.
54. PNNL, *Stream Queries for Network Monitoring*, (Part of ARDA Network Data Transformation and Analysis Project), with L. Delcambre (PI) and W.-C. Feng, June 02 - August 02.
55. NSF, *Forensix: Larger-Scale Tamper Resistant Computer Forensic System*, with W.-C. Feng (PI), W.-C. Feng, J. Walpole, STI 0230960, October 02 - September 05.
56. Intel-Oregon University System, *Curriculum Fellowship: Undergraduate Research Institute*, with D. Hansen (George Fox University) and W.-C. Feng. January 04 - December 04.
57. Agency for Healthcare Research and Quality (AHRQ), *Using IT to Improve Medication Safety of Rural Elders*, with P. Gorman (PI, OHSU), M. Pavel (OGI), K. Ordelheide (PI, Samaritan North Lincoln Hospital), L. Fagnan (ORPRN), J. Logan (OHSU), others, October 04 - September 08.

58. NSF NSDL, *Superimposed Tools for Active Arrangement and Elaboration of Educational Resources*, with L. Delcambre, E. Fox (Virginia Tech), Lillian Cassel (Villanova), and R. Kelapure (Virginia Tech), DUE 0435496, October 04 – September 06.
59. NSF, *Adapting Information using Superimposed Models and Structures*, with L. Delcambre (PI), IIS 0534762, February 2006 – January 2009.
60. NSF, *Exploiting Live Plus Archive Data for Intelligent Transportation Systems*, with R. Bertini (co-PI) and K. Tufte, IIS 0612311, July 2006 – June 2008.
61. NSF, *Science and Technology Center for Coastal Margin Observation and Prediction*, with A. Baptista (PI, OHSU), D. Martin (Co-PI, UW APL), B. Menge (Co-PI, OSU) and P. Zuber (Co-PI, OHSU), OCE 0424602, July 2006 – June 2011.
62. NSF, *SGER: From Measurement to Management: Evidence-Based Practice in Natural Resource Management*, with J. Cushing (PI, Evergreen), L. Delcambre (Co-PI) and N. Nadkarni (Co-PI, Evergreen), IIS 0639588, August 2006 – January 2008.
63. Oregon Transportation and Research Consortium (OTREC), *Improving Travel Information Products via Robust Estimation Techniques*, with K. Tufte, 07-64, April 2007 – January 2008.
64. AHRQ, *RxSafe: Shared Medication Management and Decision Support for Rural Clinicians*, with L. J. Fagnan (OHSU), P. Gorman (PI, OHSU), Heather Young (OHSU) 1R18HS017102-01, September 2007 – August 2010.
65. Lockheed, *Pay-as-You-Go Information Integration* (Part of DARPA Information Integration seedling program), with L. Delcambre, September 2007 – June 2008.
66. Oregon Transportation and Research Consortium (OTREC), *Exploiting a Next Generation ITS Data Warehouse for Improved System Performance and Congestion Monitoring*, with R. Bertini (PI) and K. Tufte, 09-269, October 2009 – September 2009.

PUBLICATIONS

Papers and Reports

1. A construction of the real numbers. With E.A. Maier. *The Two-Year College Mathematics Journal*, Winter 1973.
2. The complexity of some problems on subsequences and supersequences. *Journal of the ACM* 25(2), April 1978.
3. An efficient method for storing ancestor information in trees. *SIAM Journal on Computing* 8(4), November 1979.
4. On finding minimal length superstrings. With J. Gallant and J.A. Storer. *Journal of Computer and System Sciences* 20(1), February 1980.
5. Representing databases in segmented name spaces. With V. Gligor. In *Databases: Improving Usability and Responsiveness*, B. Shneiderman, editor, Academic Press, 1978.
6. Minimum covers in the relational database model. *Journal of the ACM* 27(4), October 1980.
7. Testing implication of data dependencies. With A.O. Mendelzon and Y. Sagiv. *ACM Transactions on Database Systems* 4(4), December 1979.
8. Generalized mutual dependencies and the decomposition of database relations. With A.O. Mendelzon. *Proceedings of the Fifth International Conference on Very Large Data Bases*, October 1979.
9. Adequacy of decompositions of relational databases. With A.O. Mendelzon, F. Sadri and J.D. Ullman. *Advances in Data Base Theory*, vol. 1, H. Gallaire, J. Minker, and J.M. Nicolas, editors, Plenum, 1981. Also in *Journal of Computer and System Sciences* 21(3), December 1980.
10. Hysterical B-trees. With S.C. Salveter. *Information Processing Letters* 12(4), August 1981.

11. On the complexity of testing implications of functional and join dependencies. With Y. Sagiv and M. Yannakakis. *Journal of the ACM* 28(4), October 1981.
12. Finding augmented set bases. With V. Gligor. *SIAM Journal of Computing* 11(3), August 1982.
13. Discarding the universal instance assumption. Stony Brook Computer Science Report 80/008. Presented at the XP1 Workshop on Relational Database Theory, Stony Brook, June-July 1980.
14. Properties of acyclic database schemes. With C. Beeri, R. Fagin, A.O. Mendelzon, J.D. Ullman and M. Yannakakis. *Proceedings of the Thirteenth Symposium on the Theory of Computing*, May 1981.
15. Maximal objects and the semantics of universal relation databases. With J.D. Ullman. *ACM Transactions on Database Systems*, 8(1), March 1983.
16. Incorporating computed relations in relational databases. With D.S. Warren. *Proceedings of the 1981 ACM Conference on Management of Data*, April 1981.
17. Tools for template dependencies. With R. Fagin, J.D. Ullman and M. Yannakakis. *SIAM Journal on Computing* 12(1), February 1983.
18. On the desirability of acyclic database schemes. With C. Beeri, R. Fagin and M. Yannakakis. *Journal of the ACM* 30(3), July 1983.
19. Connections in acyclic hypergraphs. With J.D. Ullman. *Theoretical Computer Science*, 32(1-2), July 1984.
20. Fragments of relations. With J.D. Ullman. *Proceedings of the 1983 ACM SIGMOD Conference on Management of Data*, May 1983.
21. Specifying connections for a universal relation scheme database. With D.S. Warren. *Proceedings of the 1982 ACM SIGMOD Conference on Management of Data*, June 1982.
22. Toward logical data independence: A relational query language without relations. With D. Rozenshtein, S.C. Salveter, J. Stein and D.S. Warren. *Proceedings of the 1982 ACM SIGMOD Conference on Management of Data*, June 1982.
23. Using write-once memory for database storage. *Proceedings of the ACM Symposium on Principles of Database Systems*, March 1982.
24. Verbs in Databases. With S.C. Salveter. *Proceedings of the Fourth National Conference of the Canadian Society for Computational Studies of Intelligence*, May 1982.
25. Natural language database updates. With S.C. Salveter. *Proceedings of the 20th Meeting of the Association for Computational Linguistics*, June 1982.
26. Natural language updates. With S.C. Salveter. *Proceedings of the Ninth International Conference on Computational Linguistics*, July 1982.
27. Supporting natural language updates in database systems. With S.C. Salveter. *Proceedings of the European Conference on Artificial Intelligence*, July 1982. Also in *Progress in Artificial Intelligence*, L. Steels and J. A. Campbell, editors, Ellis Horwood Ltd., 1985.
28. On the foundations of the universal relation model. With J.D. Ullman and M.Y. Vardi. *ACM Transactions on Database Systems* 9(2), June 1984.
29. Window functions. With D. Rozenshtein and D.S. Warren. In *Advances in Computing Research: The Theory of Databases*, P. C., F. Preparata, editors, JAI Press, 1986.
30. Correcting faults in write-once memory. With D. Dolev, H. Mairson, and J.D. Ullman. *Proceedings of the Sixteenth Symposium on the Theory of Computing*, May 1984.
31. Representing roles in universal scheme interfaces. With D. Rozenshtein and J. Stein. *IEEE Transactions on Software Engineering* SE-11(7), July 1985.
32. Natural language database access. With S.C. Salveter. *Proceedings of the IEEE Fall COMPCON Conference*, September 1983.

33. Capturing more meaning in databases. *Journal of Management Information Systems*, 1(1), Summer 1984.
34. Databases in the Fifth Generation project: Is Prolog a database language? *New Directions for Database Systems*, G. Ariav, J. Clifford, editors, Ablex Publishing Co., 1986.
35. Making Smalltalk a database system. With G. P. Copeland. *Proceedings of the 1984 ACM SIGMOD Conference on Management of Data*, June 1984. Also in *Readings in Database Systems*, M. Stonebraker, editor, Morgan Kaufmann, 1988.
36. Universal scheme interfaces and the theory of window functions. *Proceedings of the Theoretical Issues in Database Systems Workshop*, Benodet, France, July 1984.
37. Data model requirements for engineering applications. With D. Price. *Proceedings of the First International Workshop on Expert Database Systems*, Kiawah Island, South Carolina, October 1984.
38. Relaxing the universal relation scheme assumption. With J. Stein. *Proceedings of the Fourth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, March 1985.
39. Object oriented database systems and knowledge systems. With C. Zaniolo, H. Ait-Kaci, D. Beech, S. Cammarata and L. Kerschberg. *Expert Database Systems*, L. Kerschberg, editor, Benjamin/Cummings Publishing Company, 1986.
40. Object-oriented database development at Servio Logic. With A. Otis and A. Purdy. *Database Engineering*, 8(4), December 1985.
41. The OGC Cognitive Architecture Project. With D. Hammerstrom and S. Thakkar. *SIGARCH Computer Architecture News* 14(1), January 1986.
42. A dynamic tree-locking protocol. With A. Croker. *Proceedings of the 1986 IEEE International Conference on Data Engineering*, February 1986.
43. Magic sets and other strange ways to implement logic programs. With F. Bancilhon, Y. Sagiv, and J.D. Ullman. *Proceedings Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, March 1986.
44. Displaying database objects. With P. Nordquist and M. Grossman. *Proceedings of the First International Conference on Expert Database Systems*, April 1986. Also in *Readings in Object-Oriented Database Systems*, S. Zdonik and D. Maier, editors, Morgan Kaufmann, 1990.
45. Development of an object-oriented DBMS. With J. Stein, A. Otis and A. Purdy. *Proceedings of the 1986 ACM Conference on Object-Oriented Programming Systems, Languages and Applications*, September-October 1986. Also in *Research Foundations in Object-Oriented and Semantic Database Systems*, A.F. Cardenas and D. McLeod, editors, Prentice Hall, 1990.
46. QUICKTALK: A Smalltalk-80 dialect for defining primitive methods. With M. B. Ballard and A. Wirfs-Brock. *Proceedings of the 1986 ACM Conference on Object-Oriented Programming Systems, Languages and Applications*, September-October 1986.
47. The PROTEUS bibliography: Representation and interactive display in databases. With T. Lougenia Anderson and Earl Ecklund. *SIGMOD Record* 15(3), September 1986.
48. Indexing in an object-oriented DBMS. With J. Stein. *Proceedings of the International Workshop on Object-Oriented Databases*, September 1986.
49. PROTEUS: Objectifying the DBMS user interface. With T. L. Anderson and E. F. Ecklund, Jr. *Proceedings of the International Workshop on Object-Oriented Databases*, September 1986.
50. Integrating an object server with other worlds. With A. Purdy and B. Schuchardt. *ACM Transactions on Office Information Systems* 5(1), January 1987. Also in *Object-Oriented Databases with Applications to CASE, Networks and VLSI CAD*, R. Gupta and E. Horowitz, editors, Prentice-Hall, 1991.
51. The filter browser – Defining interfaces graphically. With R. K. Ege and A. Borning. *Proceedings of the European Conference on Object-Oriented Programming*, June 1987.

52. PIQUE: A relational query language without relations. With D. Rozenshtein, S. Salveter, J. Stein and D. S. Warren. *Information Systems* 12(3), 1987.
53. Development and Implementation of an Object-Oriented DBMS. With J. Stein. In *Research Directions in Object-Oriented Programming*, B. D. Shriver, P. Wegner, editors, MIT Press, 1987. Also in *Readings in Object-Oriented Database Systems*, S. Zdonik and D. Maier, editors, Morgan Kaufmann, 1990.
54. Is the disk half full or half empty?: Combining optimistic and pessimistic concurrency mechanisms in a shared, persistent object base. With D. J. Penney and J. Stein. *Proceedings of the Workshop on Persistent Object Systems*, Appin, Scotland, August 1987.
55. Representing database programs as objects. In *Advances in Database Programming Languages*, F. Bancilhon and P. Buneman, editors, ACM Press, 1990.
56. Concepts in object-oriented data management. With J. Stein. *Database Programming & Design* 1(4), April 1988.
57. Abstract objects in an object-oriented data model. With J. Zhu. *Proceedings of the Second International Conference on Expert Database Systems*, Tysons Corner, Virginia, April 1988.
58. Making database systems fast enough for CAD applications. *Object-Oriented Concepts, Databases and Applications*, W. Kim, F. Lochovsky, editors, ACM Press, 1989.
59. The GemStone data management system. With B. Bretl, A. Otis, D. J. Penney, B. Schuchardt, J. Stein, E. H. Williams and M. Williams. *Object-Oriented Concepts, Databases and Applications*, W. Kim, F. Lochovsky, editors, ACM Press, 1989.
60. Associative access support in GemStone. With J. Stein. In *On Object-Oriented Database Systems, Topics in Information Systems* series, K. Dittrich, U. Dayal, A. Buchmann, editors, Springer-Verlag, 1991.
61. PROTEUS: The DBMS user interface as an object. With T.L. Anderson and E.F. Ecklund, Jr. In *On Object-Oriented Database Systems, Topics in Information Systems* series, K. Dittrich, U. Dayal, A. Buchmann, editors, Springer-Verlag, 1991.
62. Query optimization in object-oriented database systems: a prospectus. With G. Graefe. *Advances in Object-Oriented Database Systems: Proceedings of the 2nd International Workshop on Object-Oriented Database Systems*, K. Dittrich, editor, Bad Muenster, West Germany, September 1988. Lecture Notes in Computer Science 334, Springer-Verlag.
63. Multilanguage object-oriented systems: New answer to old database problems? With F. Bancilhon. *Programming of Future Generation Computer II*, K. Fuchi and L. Kott, editors, North-Holland, 1988.
64. Representing CSG solids using a logic-based object data model. With T.L. Anderson, H. Ohkawa, J. Gjovaag and Sheryl Shulman. *Proceedings of the IFIP Working Conference on Visual Database Systems*, Tokyo, Japan, April 1989.
65. Why not change the languages to fit the databases? *Proceedings of the 1989 ACM SIGMOD Workshop on Software CAD Databases*, Napa, California, February 1989.
66. Computational objects in object-oriented data models. With J. Zhu. *Database Programming Languages: Proceedings of the Second International Workshop*, R. Hull, R. Morrison and D. Stemple, editors, Morgan Kaufmann, 1989.
67. Mistaking identity. With T.L. Anderson and J. Stein. In *Database Programming Languages: Proceedings of the Second International Workshop*, R. Hull, R. Morrison and D. Stemple, editors, Morgan Kaufmann, 1990.
68. Why isn't there an object-oriented data model? Invited paper, *Proceedings of the IFIP 11th World Computer Conference (Information Processing 89)*, San Francisco, California, August-September 1989.

69. Features of the TEDM Object Model. With J. Zhu and H. Ohkawa. *Deductive and Object-Oriented Databases, Proceedings of the First International Conference*, Kyoto, Japan, December 1989, W. Kim, J.-M. Nicolas and S. Nishio, editors, Elsevier Science, 1990.
70. The object-oriented database system manifesto. With M. Atkinson, F. Bancilhon, D. DeWitt, K. Dittrich and S. Zdonik. *Deductive and Object-Oriented Databases, Proceedings of the First International Conference*, Kyoto, Japan, December 1989, W. Kim, J.-M. Nicolas and S. Nishio, editors, Elsevier Science, 1990.
71. A study of three alternative workstation-server architectures for object oriented database systems. With D. DeWitt, P. Futtersack and F. Velez. *Proceedings of 16th International Conference on Very Large Databases*, Brisbane, Australia, August 1990. Also in *Building an Object-Oriented Database System: The Story of O2*, F. Bancilhon, C. Delobel and P. Kanellakis, editors, Morgan Kaufmann, 1991.
72. Implementing the O2 object manager: some lessons. With F. Velez, V. Darnis, D. DeWitt, P. Futtersack, G. Harrus and M. Raoux. In *Implementing Persistent Object Bases: Proceedings of the Fourth International Workshop on Persistent Object Systems*, A. Dearle, G.M. Shaw, S.B. Zdonik, eds., Morgan Kaufmann 1991.
73. Comments on the "Third-generation database system manifesto," OGI CSE Technical Report 91-012, April 1991.
74. Efficient assembly of complex objects. With T. Keller and G. Graefe. *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, Denver, Colorado, May 1991.
75. Introduction to languages. With F. Bancilhon. In *Building an Object-Oriented Database System: The Story of O2*, F. Bancilhon, C. Delobel and P. Kanellakis, editors, Morgan Kaufmann, 1992.
76. Query optimization in Revelation, an overview. With S. Daniels, G. Graefe, T. Keller, D. Schmidt and B. Vance. *IEEE Data Engineering Bulletin* 14(2), June 1991.
77. The O2 system. O. Deux et al. *Communications of the ACM* 34(10), October 1991.
78. Specifying a database system to itself. Invited paper, *Specifications of Database Systems*, D.J. Harper and M.C. Norrie, editors, Springer-Verlag, 1992.
79. Supporting display generation for complex database objects. With B.B. Flynn. *ACM SIGMOD Record* 21(1), March 1992.
80. Challenges for query processing in object-oriented databases. With S. Daniels, T. Keller, B. Vance, G. Graefe, and W. McKenna. In *Query Processing for Advanced Database Systems*, J.C. Freytag, D. Maier and G. Vossen, editors, Morgan Kaufmann, 1993.
81. Object-oriented database support for computational chemistry. With J.B. Cushing, M. Rao, D.M. DeVaney and D. Feller. *Proceedings of the Sixth International Working Conference on Scientific and Statistical Database Management*, Ancona, Switzerland, June 1992.
82. Issues in distributed object assembly. With G. Graefe, L. Shapiro, S. Daniels, T. Keller and B. Vance. In *Distributed Object Management*, M.T. Ozsü, U. Dayal and P. Valduriez, editors, Morgan Kaufmann, 1994.
83. The Role of Metadata in Managing Large Environmental Science Datasets: Annotative metadata in scientific applications. With L.M.L. Delcambre. In *Proceedings of SDM'92, Scientific Data Management Workshop*, Pacific Northwest Laboratories, Report PNL-5A-26092, June 1995.
84. An object-oriented heterogeneous database for materials science. With D. Hansen, J. Stanley and J. Walpole. *Scientific Programming* 1(2), Winter 1992.
85. Connecting scientific programs and data using object databases. With J.B. Cushing, D. Hansen and C. Pu. *IEEE Data Engineering Bulletin* 16(1), March 1993.
86. Object data models for shared molecular structures. With J.B. Cushing, D.M. Hansen, G.D. Purvis III, R.A. Bair, D.M. DeVaney, D.F. Feller, M.A. Thompson. In *Computerized Chemical*

- Data Standards: Databases, Data Interchange, and Information Systems*, R. Lysalcowski and D.E. Cragg, editors, ASTM STP 1214, 1994.
87. A call to order. Invited Paper. With B. Vance. *Proceedings of the ACM Symposium on Principles of Database Systems*, May 1993.
 88. Storage system architectures for continuous media data. Invited Paper. With J. Walpole and R. Staehli. *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms (FODO '93)*, Chicago, October 1993.
 89. Specifying rule-based query optimizers in a reflective framework. With L. Fegaras and T. Sheard. *Proceedings of the 3rd International Conference on Deductive and Object-Oriented Databases (DOOD '93)*, Phoenix, December 1993.
 90. Treating programs as objects: The computational proxy experiment. Invited Paper. With J.B. Cushing. *Proceedings of the 3rd International Conference on Deductive and Object-Oriented Databases (DOOD '93)*, Phoenix, December 1993.
 91. Using an object-oriented database to encapsulate heterogeneous scientific data sources. With D.M. Hansen. *Proceedings of the Twenty-seventh Annual Hawaii International Conference on System Sciences*, Wailea, January 1994.
 92. Bambi Meets Godzilla: Object databases for scientific computing. With D.M. Hansen. Invited Paper, *Proceedings of the Seventh International Working Conference on Scientific and Statistical Database Management*, Charlottesville, Virginia, September 1994.
 93. Computational proxies: Modeling scientific applications in object databases. With J.B. Cushing, M. Rao, D. Abel, D. Feller and D. M. DeVaney. *Proceedings of the Seventh International Working Conference on Scientific and Statistical Database Management*, Charlottesville, Virginia, September 1994.
 94. Applications of object database technology in thermodynamics and materials science. With R. Venkatesh, D. Hansen and J. M. Stanley. *Proceedings of the Symposium on Applications of Thermodynamics in the Synthesis and Processing of Materials*, Rosemont, Illinois, October 1994.
 95. Quality of service specifications for multimedia presentation. With R. Staehli and J. Walpole. *Multimedia Systems* 3 (5/6), November 1995.
 96. Towards an effective calculus for object query languages. With L. Fegaras. *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, San Jose, May 1995.
 97. An algebraic framework for physical OODB design. With L. Fegaras. *Proceedings of the 5th International Workshop on Database Programming Languages*, Gubbio, Italy, September 1995.
 98. Interoperating with DIF data. With P. Benninghoff. *Proceedings of RIDE-NDS'96 – Sixth International Workshop on Research Issues in Data Engineering*, New Orleans, February 1996.
 99. Device and physical data independence. With R. Staehli and J. Walpole. *ACM Computing Surveys* 27(4), December 1995.
 100. Bringing knowledge to bear: Challenges for decision support databases. Invited Paper. With M.E. Meredith and L. Shapiro. *Proceedings of the 7th International Hong Kong Computer Society Database Workshop*, 1996.
 101. Rapid bushy join-order optimization with Cartesian products. With B. Vance. *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, Montreal, June 1996.
 102. Proxies in practice: Object Architectures from distributed computational workbenches. With J.B. Cushing, T. Keller, and T. Marr. *Journal of the Brazilian Computer Society* 2(3), November 1996.
 103. Structured maps: Modeling explicit semantics over a universe of information. With L.M.L. Delcambre, R. Reddy and L. Anderson. *International Journal on Digital Libraries* 1(1), 1997.

104. A player for adaptive MPEG video streaming over the Internet. With J. Walpole, R. Koster, S. Cen, C. Cowan, D. McNamee, D. Steere, and L. Yu. *Proceedings 26th Applied Imagery and Pattern Recognition Workshop, AIPR-97, SPIE, Washington DC, October 1997.*
105. Database support for forest canopy researchers – Metadata as a byproduct of the research process. With J. Cushing, N. Nadkarni,, S. Knackstedt, E. Lyons and L. Delcambre, *Proceedings of the Conference on Scientific and Technical Data Exchange (CoDATA), NRC, 1997.*
106. Stackguard: Automatic adaptive detection and prevention of buffer-overflow attacks. With C. Cowan, C. Pu, H. Hinton, P. Bakke, S. Beattie, A. Grier, P. Wagle, and Q. Zhang. *Proceedings of the 7th USENIX Security Conference, San Antonio, January 1998.*
107. Precis-based navigation for familiarization. With L. Delcambre, P. Gorman, and S. Rehfuss. *Proceedings of the 9th World Congress on Medical Informatics, Seoul, Korea, August 1998.*
108. Selected research issues in decision support databases. With M.E. Meredith and L. Shapiro. *Journal of Intelligent Information Systems 11(2), September-October 1998.*
109. Quality of service semantics for multimedia database systems. With J. Walpole, C. Krasic, L. Liu, C. Pu, D. McNamee and D. Steere. *Proceedings of Data Semantics 8: Semantic Issues in Multimedia Systems, IFIP TC-2 Working Conference, Rotorura, New Zealand, January 1999.*
110. The Asilomar report on database research. With P. Bernstein, M. Brodie, S. Ceri, D. DeWitt, M. Franklin, H. Garcia-Molina, J. Gray, J. Held, J. Hellerstein, H. Jagadesh, M. Lesk, J. Naughton, H. Pirahesh, M. Stonebraker and J. Ullman. *SIGMOD Record 27(4), December 1998.*
111. Superimposed information for the Internet (Invited Paper). With L. Delcambre. *Proceedings of the Second International Workshop on the Web and Databases, WebDB'99, Philadelphia, June 1999.*
112. Querying XML data. With A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and Dan Suciu. *IEEE Data Engineering Bulletin 22(3), September 1999.*
113. Models for superimposed information. With L. Delcambre. *Advances in Conceptual Modeling (Proceedings of the ER '99 Workshops on Evolution and Change in Data Management, Reverse Engineering in Information Systems, and the World Wide Web and Conceptual Modeling), LNCS 1727, Paris, November 1999.*
114. Architecting a Network Query Engine for Producing Partial Results. With J. Shanmugasundaram, K. Tufte, D. DeWitt, and J. Naughton. *Proceedings of the Third International Workshop on the Web and Databases, WebDB 2000, Dallas, May 2000, (Lecture Notes in Computer Science 1997, Springer, 2001).*
115. The hybrid technique for reference materialization in object query processing. With Q. Wang and L. Shapiro. *Proceedings of the 4th International Conference on Database Engineering and Applications (IDEAS'2000), Tokyo, Japan, September 2000.*
116. Optimizing object queries using an effective calculus. With L. Fegaras. *ACM Transactions on Database Systems 25(4), December 2000.*
117. Bundles in the wild: Tools for managing information to solve problems and maintain situation awareness. With P. Gorman, J. Ash, M. Lavelle, J. Lyman, L. Delcambre, S. Bowers and M. Weaver. *Library Trends 49(2), pages 266-289, Fall 2000.*
118. Bundles in captivity: An application of superimposed information. With L. Delcambre, S. Bowers, L. Deng, M. Weaver, P. Gorman, J. Ash, M. Lavelle, and J. Lyman. *Proceedings of the IEEE International Conference on Data Engineering, Heidelberg, Germany, April 2001.*
119. Bundles: Meeting Clinical Information Needs. With J. Ash, P. Gorman, M. Lavelle, J. Lyman, L. Delcambre, S. Bowers and M. Weaver. *Bulletin of the Medical Library Association 89(3), July 2001.*
120. Research Directions in Biodiversity and Ecosystem Informatics. With E. Landis, J. Cushing, A. Frondorf, A. Silberschatz, M. Frame and J. L. Schnase. Report of an NSF, USGS, NASA Workshop, Greenbelt, MD, June 2000.

121. The NIAGARA Internet Query System. With J. F. Naughton, D. J. DeWitt, A. Abounaga, J. Chen, L. Galanis, J. Kang, R. Krishnamurthy, Q. Luo, N. Prakash, R. Ramamurthy, J. Shanmugasundaram, F. Tian, K. Tufte, S. Viglas, Y. Wang, C. Zhang, B. Jackson, A. Gupta and R. Chen. *IEEE Data Engineering Bulletin* 24(2), June 2001.
122. Aggregation and Accumulation of XML Data. With K. Tufte. *IEEE Data Engineering Bulletin* 24(2), June 2001.
123. Exploiting upper and lower bounds in top-down query optimization. With L. Shapiro, P. Benninghoff, K. Billings, Y. Fan, K. Hatwal, B. Vance, and Q. Wang. *Proceedings of the 5th International Conference on Database Engineering and Applications (IDEAS'2001)*, Grenoble, France, July 2001.
124. A Superimposed Architecture for Enhanced Metadata. With M. Weaver and L. Delcambre. *Proceedings of the Third DELOS Network of Excellence Workshop on Interoperability and Mediation in Heterogeneous Digital Libraries*, Darmstadt, Germany, September 2001.
125. Mutant Query Plans. With V. Papadimos. *Information & Software Technology* 44(4), March 2002.
126. Enriching Documents in an Information Portal using Superimposed Schematics. With S. Bowers and L. Delcambre. *Proceedings Dg.o 2002 National Conference on Digital Government Research*, Los Angeles, May 2002.
127. Representing, Exploiting, and Extracting Metadata using Metadata++. With M. Weaver, B. Howe, L. Delcambre and T. Tolle. *Proceedings Dg.o 2002 National Conference on Digital Government Research*, Los Angeles, May 2002.
128. Using Controlled Vocabularies as a Knowledge Base for Natural Resource Managers. With T. Tolle, P. Tocalino, L. Delcambre, J. Norman and F. Phillips. *Proceedings Dg.o 2002 National Conference on Digital Government Research*, Los Angeles, May 2002.
129. Spatial Infrastructure for Ecological Research. With J. Cushing, N. Nadkarni and L. Delcambre. *Proceedings Dg.o 2002 National Conference on Digital Government Research*, Los Angeles, May 2002.
130. Distributed Queries without Distributed State. With V. Papadimos. *Proceedings of the Fifth International Workshop on the Web and Databases, WebDB 2002*, Madison, Wisconsin, June 2002.
131. Template-driven End-User Ecological Database Design. With J. Cushing, N. Nadkarni, K. Healy, E. Ordway, and L. Delcambre. *Proceedings of the 6th World Multi-conference on Systemics, Cybernetics and Informatics (SCI 2002)*, Orlando, FL, July 2002.
132. Superimposed Schematics: Introducing E-R Structure for In-Situ Information Selections. With S. Bowers and L. Delcambre, *Proceedings Conceptual Modeling - ER 2002, 21st International Conference on Conceptual Modeling*, Tampere, Finland, October 2002, (Lecture Notes in Computer Science 2503, Springer, 2002).
133. Following Experts at Work in their Own Information Spaces: Using Observational Methods to Develop Tools for the Digital Library. With P. Gorman, M. Lavelle, and L. Delcambre. *Journal of the American Society of Information Science and Technology* 53(14), November 2002.
134. Harvesting Information to Sustain Forests. With L. Delcambre, T. Tolle, and the Forest Project Team. *Communications of the ACM* 46(1), January 2003.
135. Distributed Query Processing and Catalogs for Peer-to-Peer System, With V. Papadimos and K. Tufte. *Proceedings CIDR 2003: First Biennial Conference on Database Systems Research*, Asilomar, CA, January 2003.
136. Exploiting Punctuation Semantics in Continuous Data Streams. With P. Tucker, T. Sheard and L. Fegaras. *IEEE Transactions on Knowledge and Data Engineering* 15(3), May/June 2003.
137. Information System Challenges of Biodiversity and Ecosystem Informatics. With J. L. Schnase, J. Cushing, M. Frame, A. Frondorf, E. Landis and A. Silberschatz. *Information Systems* 28(4), June 2003.

138. Knowledge Management: A Reassessment. With F. Phillips, T. Tolle, E. Landis, L. Delcambre, M. Koch, N. Steckler, P. Toccalino and B. Banga. To appear in *Global Knowledge Management: Challenges and Opportunities*, H. R. Nemati, P. Palvia and R. Ajami, editors.
139. Will Databases Fail Bioinformatics, Too? *OMICS: A Journal of Integrative Biology* 7(1), January 2003.
140. Harvesting information to Sustain Forests. With L. Delcambre, T. Tolle and The Forest Project Team, *Communications of the ACM* 46(1), January 2003.
141. Applying Punctuation Schemes to Queries Over Continuous Data Streams. With P. Tucker and Tim Sheard. *IEEE Data Engineering Bulletin* 26(1), March 2003.
142. The Forest Portal: A Multidisciplinary Project. With M. Koch, L. Delcambre, P. Toccalino, E. Landis, F. Phillips, T. Tolle, L. Shapiro, N. Steckler, M. Weaver, S. Bowers, B. Banga, J. Brewster, A. Gutema, S. Murthy, B. Howe, R. Tummala, J. Norman, K. Zillman, D. Drake, C. Palmer, and A. Burt. *Proceedings Dg.o 2002 National Conference on Digital Government Research*, Boston, MA, May 2003.
143. The Lowell Database Research Self Assessment. With S. Abiteboul, R. Agrawal P. Bernstein, M. Carey, S. Ceri, B. Croft, D. DeWitt, M. Franklin, H. Garcia Molina, D. Gawlick, J. Gray, L. Haas, A. Halevy, J. Hellerstein, Y. Ioannidis, M. Kersten, M. Pazzani, M. Lesk, J. Naughton, H. Schek, T. Sellis, A. Silberschatz, M. Stonebraker, R. Snodgrass, J. Ullman, G. Weikum, J. Widom and S. Zdonik. June 2003. *Communications of the ACM* 48(5), June 2005.
144. Dealing with Disorder. With P.A. Tucker. Presented at the Workshop on Management and Processing of Data Streams, San Diego, June 2003.
145. Environmental Observation and Forecasting Systems: Vision, Challenges and Successes of a Prototype. With A. Baptista, T. Leen, Y. Zhang, A. Chawla, W.-Chi Feng, W.-Chang Feng, J. Walpole, C. Silva and J. Freire. Accepted for International Society for Environmental Information Sciences Annual Conference (ISEIS'2003), Regina, SA, July 2003.
146. Superimposing Spatial Enrichments in Traditional Information [Extended Abstract]. With L. Delcambre, M. Weaver, L. Shapiro and J.B. Cushing. International Workshop on Next Generation Geospatial Information (NG2I'03), Cambridge, Massachusetts, October 2003.
147. A Language for Environmental Data Manipulation. With B. Howe and A. Baptista. *Journal of Environmental Informatics* 2(3), December 2003.
148. Putting Integrated Information in Context: Superimposing Conceptual Models with SPARCE. With S. Murthy, L. Delcambre, and S. Bowers. *Proceedings of the First Asia-Pacific Conference on Conceptual Modeling*, Dunedin, New Zealand, January 2004.
149. Querying Bi-Level Information. With S. Murthy and L. Delcambre. *Proceedings of the Seventh International Workshop on the Web and Databases (WebDB2004)*, Paris, June 2004.
150. Emergent Semantics: Towards Self-Organizing Scientific Metadata. With B. Howe and P. Turner. *Proceedings of the International Conference on Semantics of Networked World: Semantics for Grid Databases*, Paris, June 2004. Springer-Verlag LNCS 3226.
151. Distribution Alternatives for Superimposed Information Services in Digital Libraries. With S. Murthy and L. Delcambre. *Peer-to-Peer, Grid, and Service-Oriented in Digital Library Architectures, 6th Thematic Workshop of the EU Network of Excellence DELOS*, Cagliari, Italy, June 2004. Springer-Verlag LNCS 3664, 2005
152. Algebraic Manipulation of Scientific Datasets. With B. Howe. *Proceedings 30th International Conference on Very Large Databases*, Toronto, August-September 2004. Extended version in *The VLDB Journal* 14(4), November 2005.
153. Linear Road: A Stream Data Management Benchmark. With A. Arasu, M. Cherniack, E. F. Galvez, A. Maskey, E. Ryvkina, M. Stonebraker and R. Tibbetts. *Proceedings 30th International Conference on Very Large Databases*, Toronto, August-September 2004.

154. Logical and Physical Data Independence for Native Scientific Data Repositories. With Bill Howe. *IEEE Data Engineering Bulletin* 27(4), December 2004.
155. Semantics of Data Streams and Operators (Invited Paper), With J. Li, V. Papadimos, P. Tucker and K. Tufte. *Proceedings 10th International Conference on Database Theory*, Edinburgh, Scotland, January 2005.
156. Deriving and Managing Data Products in an Environmental Observation and Forecasting System. With L. Bright. *Proceedings CIDR 2005: Second Biennial Conference on Database Systems Research*, Asilomar, CA, January 2005.
157. No pane, no gain: efficient evaluation of sliding-window aggregates over data streams. With J. Li, K. Tufte, V. Papadimos and P. Tucker. *SIGMOD Record* 34(1), March 2005.
158. Forensix: A Robust, High-Performance Reconstruction System. With A. Goel, W.-C. Feng, W.-C. Feng and J. Walpole. *Proceedings of the Fourth International Workshop on Assurance in Distributed Systems and Networks (ADSN)*, Columbus, Ohio, June 2005.
159. Semantics and Evaluation Techniques for Window Aggregates in Data Streams. With J. Li, K. Tufte, V. Papadimos and P. Tucker. *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, Baltimore, June 2005.
160. Retrofitting a Data Model to Existing Environmental Data. With B. Howe. *Proceedings of the 17th International Conference on Scientific and Statistical Database Management*, Santa Barbara, California, June 2005.
161. Efficient Scheduling and Execution of Scientific Workflow Tasks. With L. Bright. *Proceedings of the 17th International Conference on Scientific and Statistical Database Management*, Santa Barbara, California, June 2005.
162. Digital Library Information-Technology Infrastructures. With Y. Ioannidis, S. Abiteboul, P. Buneman, S. B. Davidson, E. A. Fox, A. Y. Halevy, C. A. Knoblock, F. Rabitti, H.-J. Schek and G. Weikum. *International Journal on Digital Libraries* 5(4), August 2005.
163. Managing the Forecast Factory. With L. Bright and B. Howe. *Proceedings of the Workshop on Scientific Workflows (SciFlo 2006)*, Atlanta, Georgia, April 2006.
164. Principles of Dataspace Systems (keynote). With A. Y. Halevy and M. J. Frankli. *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Chicago, Illinois, June 2006.
165. Smoothing the ROI Curve for Scientific Data Management Applications. With B. Howe and L. Bright. *Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR 2007)*, Asilomar, CA, January 2007.
166. Component-Based End-User Database Design for Ecologists. With J. Cushing, N. Nadkarni, A. Fiala, M. Finch, E. Murphy-Hill and L. Delcambre. *Journal of Intelligent Information Systems* 29(1), August 2007.
167. Using Punctuation Schemes to Characterize Strategies for Querying over Data Streams. With P. Tucker, T. Sheard and P. Stephens. *IEEE Transactions on Knowledge and Data Engineering* 19(9), September 2007.
168. Out-of-Order Processing: A New Architecture for High-Performance Stream Systems. With J. Li, K. Tufte, V. Shkapenyuk, V. Papadimos and T. Johnson. *Proceedings of the VLDB Endowment* 1(1), August 2008.
169. Declaratively Producing Data Mash-ups. With S. Murthy. *Proceedings of the 14th Conference on the Management of Data*, Mumbai, India, December 2008.
170. A Framework for Fine-grained Data Integration and Curation. With D. Archer and L. Delcambre. *Proceedings of the Workshop on the Theory and Practice of Provenance (Usenix)*, San Francisco, February 2009.

171. Scientific Mashups: Runtime-Configurable Data Product Ensembles. With B. Howe and H. Green-Fishback. *Proceedings of the 21st International Conference on Scientific and Statistical Database Management (SSDBM 2009)*, New Orleans, LA, June 2009.

Books and Book Chapters

- The Theory of Relational Databases*. Computer Science Press, 1983, xv + 637 pp. (Russian Translation 1987, collected in the *ACM SIGMOD Anthology*, vol. 4, 2000.)
- Trees. In *The Handbook of Electrical and Computer Engineering*, S.S.L. Chang, editor, Wiley, 1983.
- Computing with Logic: Introduction to Logic Programming*. With D.S. Warren, Benjamin-Cummings, 1988, xxi + 535 pp.
- Readings in Object-Oriented Database Systems*. With S.B. Zdonik, Morgan Kaufmann, San Mateo, California, 1990, x + 629 pp.
- Query Processing for Advanced Database Systems*, J.C. Freytag, D. Maier and G. Vossen, editors, Morgan Kaufmann, 1993, xxiii + 481 pp.
- Persistent Object Systems*, M. Atkinson, D. Maier and V. Benzaken, editors, Springer-Verlag, 1995, xi + 544 pp.

STUDENT SUPERVISION

Ph.D. Theses

- Billie S. Goldstein, April 1982.
Toward More Expressive Database Specifications: Value Constraints on Relations.
- Albert Croker, June 1983.
Approaches to Increasing Database Concurrency Through Locking.
- David Rozenshtein, June 1983.
Query and Role-playing in the Association-Object Data Model.
- Jacob Stein, October 1984.
Constraints in the Association-Object Data Model.
- Mark Grossman, June 1987.
Task Interaction and Control System (TICS).
- Raimund Ege, August 1987.
Automatic Generation of Interactive Displays Using Constraints.
- Harry Porter, December 1988.
A Logic-Based Grammar Formalism Incorporating Feature Structures and Inheritance.
- Jianhua Zhu, July 1989.
Model, Language and Implementation Aspects of a Logic-based Object-Oriented Database System.
- Hitomi Ohkawa, April 1993.
Object-Oriented Database Support for Scientific Data Management: A System for Experimentation.
- Belinda Flynn, February 1994.
A Software Tool for Specifying and Generating Displays in Object-Oriented Database Applications.
- Judith Bayard Cushing, March 1995.
Computational Proxies: An Object-based Infrastructure for Computational Science.
- David Hansen, May 1995.
An Object-Oriented Heterogeneous Database Architecture.
- Bennet Vance, August 1997.
Join-Order Optimization with Cartesian Products.
- Quan Wang, March 2002.

Cost-Based Object Query Optimization.

Paul Benninghoff, April 2003.

Scalable and Efficient Active Service Integration.

Kristin Tufte, December 2004. (Research Supervisor. Thesis Chair: David J. DeWitt)

Partial Results and Structural Aggregation over XML Data Streams.

Peter Tucker, June 2005.

Punctuated Data Streams.

Bill Howe, December 2006.

Gridfields: Model-Driven Data Transformation in the Physical Sciences.

Jin Li, October 2008.

Window Queries over Data Streams.

Sudarshan Murthy, March 2009.

A Framework for Superimposed Applications: Techniques to Represent, Access, Transform, and Interchange Bi-level Information.

Masters Projects and Theses

Paula Wolk, September 1979.

Implementation of a Minimum Cover Algorithm for Relational Databases

Stephen C. Strategos, March 1980.

Implementation of Third Normal Form Relational Schema Generator.

Amos Ziv, October 1981.

The Avocado Problem.

Peter Nordquist, March 1985.

Automatic Display Generation in Smalltalk.

Mark Ballard, April 1986.

QUICKTALK: A Smalltalk-80 Dialect for Describing Primitive Methods.

Juliana Lim, May 1989.

The FilterManager: Maintaining Constraints Among Processes.

Brian Kennedy, August 1991.

Architectural Alternatives for Connecting a Persistent Programming Language and an Object-Oriented Database System.

Meenakshi Rao, November 1995.

Computational Proxies for Computational Chemistry: A Proof of Concept.

Donald Abel, 1996, (PSU, research supervisor).

The Parser Converter Loader: An Implementation of the Computational Chemistry Output Language.

Christopher Sells, May 1997.

Bridging Object Models: The Faux-Object Idiom.

Mark Wong, January 2002.

Logging Clickstream Data Into a Database on a Consolidated System.

PROFESSIONAL ACTIVITIES

Professional and Honorary Societies

Association for Computing Machinery

SIGACT

SIGMOD

Institute for Electrical and Electronic Engineers

Computer Society

Database Engineering Technical Committee
Society for Industrial and Applied Mathematics
Phi Beta Kappa
Sigma Xi
Oregonians for Rationality

Advisory Boards

Genetics Resource and Informatics Program, Oregon National Primate Resource Center, 2004
Center for Biostatistics, Informatics and Computing in Biology and Medicine, OHSU, 2003-2004.

Reviewing and Editing

Associate Editor, *ACM Transactions on Database Systems*, 1987-1994.

Associate Editor, *International Journal of Digital Libraries*, 1997-2000.

Reviewed papers for

ACM Computing Surveys
ACM Transactions on Database Systems
ACM Transactions on Office Information Systems
Acta Informatica
Advances in Computing Research, vol. 3
Advances in Database Theory, vols. 1 and 2
Annals of Mathematics and Artificial Intelligence
Communications of the ACM
Computer Magazine
Data and Knowledge Engineering
IEEE Multimedia
IEEE Software
IEEE Transactions on Computers
IEEE Transactions on Knowledge and Data Engineering
Information Processing Letters
Information Systems
International Journal of Parallel Programming
International Journal of Digital Libraries
Journal of Automated Reasoning
Journal of Computer and System Sciences
Journal of Intelligent Information Systems
Journal of Logic Programming
Journal of the ACM
Journal of the Urban and Regional Information Systems Association
Logic Programming and its Applications
Proceedings of the IEEE
SIAM Journal on Computing
Theoretical Computer Science
Theory and Practice of Object Systems
VLDB Journal

Referee for

2nd International Conference on Databases – Improving

Usability and User Responsiveness;
 IFIP '83;
 1984, 1985, 1986 International Symposia on Logic Programming;
 1986, 1987 ACM Conference on Object-Oriented Programming Systems, Languages and Applications;
 1988 Conference on Extending Database Technology;
 1988 European Conference on Object-Oriented Programming;
 Twelfth ACM Symposium on Operating System Principles;
 1990 3rd International Conference on Database Theory;
 1992 IEEE International Conference on Data Engineering;
 Persistent Object Systems, 1996.

Reviewer for *Zentralblatt für Mathematik*, 1982-1988.

Panelist for NSF Graduate Research Fellowship selections, 1982, 1983, 1985.

Small Business Innovative Research in Information Science and Technology, 1985.

Panelist for NSF Young Investigator Award selections, 1994.

Proposal reviewer for Hong Kong Research Committee (City University of Hong Kong), 2001.

Panelist, NIH Special Emphasis Panel (ZRG1 SSS-H(91)S) on Bioinformatics and Computational Biology Software, 2003.

Program Committees

- Program Organizer, XP4.5 Workshop on Database Theory, Stanford University, August 1983.
- 1984, 1988, 1991 ACM SIGACT-SIGMOD Symposia on Principles of Database Systems.
- 1984, 1986, 1988, 1989, 1993, 2001, 2005 ACM SIGMOD International Conferences on Management of Data (Best Paper sub-committee, 2001).
- Program Co-chair, 1989 ACM SIGMOD Conference on Management of Data.
- 1986 ACM Conference on Object-Oriented Programming Systems, Languages and Applications.
- Tutorial Committee member, 1990 joint European Conference on Object-Oriented Programming – ACM Conference on Object-Oriented Programming Systems, Languages and Applications.
- 1989 International Conference on Extending Database Technology.
- 1991 International Conference on Deductive and Object-Oriented Databases.
- 1991 Third International Workshop on Database Programming Languages.
- 1992 DARPA Software Technology Conference.
- Program Co-chair, Sixth International Workshop on Persistent Object Systems, 1994.
- Program Co-chair, 7th International Hong Kong Computer Society Database Workshop, 1996.
- 1997, 1999, 2004 International Conference on Very Large Data Bases, including Best 10-Year-Old Paper Award subcommittee (1997).
- International Conference on Scientific and Statistical Database Management, 1999, 2003.
- SIGMOD Awards Committee, 1998, 1999, 2000 (chair).
- International Workshop on the Web and Databases (WebDB), 2003.

- 2004 ICDE PhD Workshop.

Conference and Workshop Organization

- Organizer, XP1 Workshop on Relational Database Theory, June-July 1981, Stony Brook, New York.
- Local Arrangements chairman, 1985 ACM SIGACT-SIGMOD Symposium on Principles of Database Systems.
- Local Arrangements committee, 1986 ACM Conference on Object-Oriented Programming Systems, Languages and Applications.
- Director, CRAI International Seminar on Advanced Data and Knowledge Base Systems, 1987.
- Organizer, XP8.5i Workshop on Database Theory, Oregon Graduate Center, July 1987.
- Organizer, Short Course on Next Generation Database Systems, Portland, Oregon, 1988.
- Local arrangements chair, Second International Workshop on Database Programming Languages, 1989.
- Tutorial Arrangements, 1990 ACM SIGMOD International Conference on Management of Data.
- Organizer, Dagstuhl Seminar on Query Processing in Object-Oriented, Complex Object, and Nested Relation Databases, International Conference and Research Center for Computer Science, Germany, 1991.
- Organizer, SDM '92, Scientific Data Management Workshop.
- Session Organizer, Oregon Advanced Computing Conference, 1992.
- Best Paper Co-chair, 1994 ACM SIGMOD Conference on Management of Data.
- Workshop Chair, 1995 International Symposium on Logic Programming.
- General Chair, 9th International Conference on Scientific and Statistical Database Management, Olympia, 1997.
- Organizer, Dagstuhl Seminar on Software Engineering and Database Technology, International Conference and Research Center for Computer Science, Germany, 1997.
- Steering Committee, International Conference on Scientific and Statistical Database Management, 1997 - 2000.
- Organizer, NSF-NASA-USGS Workshop on Biodiversity Informatics, 2000.
- Organizing Committee, 2nd ACM + IEEE Joint Conference on Digital Libraries, 2002.
- Co-leader, DELOS/NSF Study Panel on Information Extraction for Digital Libraries, 2002 - 2003.
- Organizer, DLI2/IMLS/NSDL Principal Investigators Meeting, 2002.
- Organizer, National e-Science Center Workshop on Data Provenance and Annotation, Scotland, 2003.

Consulting

Brookhaven National Laboratories, 1979.
 Bell Laboratories, Holmdel, NJ, 1981.
 Stanford University, Jeffrey D. Ullman (AFOSR grant 80-0212), 1981 - 1984.
 Servio Logic Corporation, 1983 -1989.
 Microelectronics and Computer Technology Corporation, 1985.
 Digital Equipment Corp., 1987 - 1989.
 GiP Altair, 1987, 1990.

MAD Intelligent Systems, 1987.
Object Design, Technical Advisory Board, 1987 - 1993.
Honeywell (including Sweatt Awards selection committee), 1988 - 1989, 1992, 1993.
Mentor Graphics, 1988.
Texas Instruments, 1990, 1994.
Prompt, Inc., 1991 - 1992.
IBM, 1992.
Microsoft, 1995 - 1997.
Informix, 1995 - 1996.
NASA CESDIS, 1995 - 1997.
Alphatech 1996 - 1999.
Oracle, 1997 - 1998.
NCR-Teradata, 1998 - 1999.
University of Texas, Arlington, 1999 - 2001.
Preston | Gates | Ellis LLP, 2000 - 2003.
CrossGain, Technical Advisory Board, 2000 - 2001.
Cycorp, Inc. (for DARPA), 2002.
Klarquist, Sparkman LLP, 2002 - 2008.
Howrey LLP, 2007.
Microsoft Research, 2008.
Perkins Coie LLP, 2008.

Other Professional Activities

Provided questions for Advanced GRE in Computer Science, 1984, 1985.
Member of Working Group on Architectures for Interactive Systems, 1986-1988.
Chairman and Founder, Oregon Database Forum, 1987-1989, sponsor of Database Year in Portland Area, 1987-1988.
Session Leader, ONR Workshop on Critical Research Directions in Programming Languages, 1988.
Participant, Workshop on New Directions in Database Research, International Computer Science Institute, Laguna Beach, California, 1988.
Participant, DARPA Database Systems Workshop, Pacific Grove, California, 1988.
Participant, NSF Invitational Workshop on the Future of Database System Research, Palo Alto, California, 1990.
Participant, NSF-INRIA Workshop on Data Models and Types, New Hope, Pennsylvania, 1990.
Participant, ESPRIT/DARPA/NSF Exploratory Workshop on Information Science and Technology, Washington, DC, 1990.
Participant, OGI/PNL Workshop on Scientific Information Management: Issues and Opportunities, Richland, Washington, 1991.
Participant and Session Leader, Texas Instruments-DARPA Open OODB Workshops (3 workshops, contributed 3 position papers), Dallas, Texas and Boulder, Colorado, 1991-1992.
Participant, DARPA Persistent Object Bases Mini-Workshop, Boulder, Colorado, 1992.
Participant, ARPA Prototech/DSSA Workshop, Steamboat Springs, Colorado, 1993.
Participant, ARPA Persistent Object Bases Program Workshop, Salt Lake City, Utah, 1993.

Participant, TI Workshop on Database Query Optimizer Generators and Rule-Based Optimizers, Dallas, Texas, 1993.

Participant, R & D for the NII: Technical Challenges, Gaithersburg, 1994.

Participant, ARPA Persistent Object Bases Program Workshop, Madison, 1994.

Participant, E.C.-US FIDE Workshop, Tarascon, France, 1994.

Participant, ARPA Interoperability Workshop, LaJolla, 1994.

Participant, ACM/IBM Industrial Advisory Board on Future Applications of Object Technology, Snoqualmie, Washington, 1994.

Participant, NSF Second Invitational Panel on the Future of Database Systems Research (Lagunita 2), San Jose 1995.

Participant, NRC Workshop on HPCC and Crisis Management, Irvine, 1995.

Participant, ONR Tactical Picture Agent Workshops, Washington, DC and San Diego, 1995.

Participant, ARPA Superlink study panel on foreground information systems and ISAT annual meeting, Cape Cod, 1995.

Session Leader, ARPA Persistent Object Base/Intelligent Integration of Information PIs Meeting, San Diego, 1996.

Participant, ACM 50th Anniversary Workshop on Strategic Directions on Computing Research, Database Systems Panel, Cambridge, 1996.

Participant, DARPA study group on Dynamic Databases for Battlefield Awareness, 1996.

Advisor, OGI ACM Student Chapter 1995 - 1999.

Participant, NASA CESDIS Workshop on Data Mapping and Matching, Greenbelt, Maryland, 1996.

Participant, DARPA Working Group on Mediator Languages, College Park, Maryland, 1996.

Participant, Workshop on Curricular Innovations to Promote Software Development Skills, Olympia, WA, 1996.

Participant, DARPA Meeting on High Assurance and Formal Methods, Lake Placid, NY, 1996.

Participant, DARPA/ITO General PI Meeting, Survivability Working Group, Dallas, TX, 1996.

Participant, DARPA Information Management Services Workshop, Miami, FL, 1996.

Participant, DARPA I*3 Workshop, Miami, 1996.

Participant, DARPA Study Group on Architectures for Dynamic Databases, 1997.

Participant, DARPA EDCS High Assurance and Real Time Cluster, 1996 - 1997.

Participant, Rome Laboratory Information Institute Workshop on Next-Generation Information Environments, Hamilton, NY, 1997.

Group Coordinator, NSF Workshop on Information and Data Management, Washington, DC, 1998.

Participant, Infrastructure 98: 1998 NSF CISE/EIA RI and MII PI's Workshop, Snowbird, UT, 1998.

Participant, Database Directions Workshop, Asilomar, CA, 1998.

Participant, CSTB Workshop on Information Technology to Support Crisis Management, Washington, DC, 1998.

Participant, W3C QL'98 - The Query Languages Workshop, Boston, MA, 1998.

Participant, NSF Information and Data Management Workshop, Los Angeles, CA, 1999.

Participant, EU-US Workshop on Large Scientific Databases, Annapolis, MD, 1999.

Participant, NSF DLI2 (Digital Libraries) All Projects Meeting, Ithaca, NY, 1999.

Participant, RM2000 Workshop on Reflective Middleware, Palisades, NJ, 2000.

Participant, DARPA Cyberpanel meeting, New Orleans, LA, 2001.

Participant, IDM 2001 Workshop on Information and Data Management, Forth Worth, TX, 2001.

Participant, DARPA IXO Study Panel on Information Integration, 2001-2002.

Participant, Dagstuhl Seminar on Foundations of Semistructured Data, 2001.

Participant, IRCS Workshop on Linguistic Databases, 2001.

Participant, Workshop on Data Derivation and Provenance, Chicago, IL, 2002.

Participant, OWEB (Oregon Watershed Enhancement Board) Expertise Workshop, Salem, OR, 2002.

Participant and final report contributor, NSF/NLM/DOE Workshop on Data Management for Molecular and Cell Biology, Bethesda, MD, 2003.

Participant, NSF Biodiversity and Ecosystems Informatics PIs Meeting, Washington, DC, 2003.

Participant, Asilomar++ Database Directions Workshop, Lowell, MA, 2003.

Participant, IDM 2003 Workshop on Information and Data Management, Seattle, 2003.

ORION (Ocean Research Interactive Observatory Networks) Workshop, San Juan, Puerto Rico, 2004.

Participant, OOS Tech Workshop on Data Standards for Ocean Observing Systems, Washington, D.C., 2004.

PANELS, TUTORIALS AND PRESENTATIONS

Tutorials

Taught 1-week seminar on Object-Oriented Database Systems (5 times), Berlin Continuing Engineering Education Program, 1987-1991.

Lecturer, Western Institute in Computer Science, 1988-1989, 1992-1995.

Tutorial on Object-Oriented Database Systems, with S.B. Zdonik, IEEE Data Engineering Conference, 1989

Tutorial, "Object-Oriented Database Systems: What's Next?" at the International Conference on Extending Database Technology, Venice, 1990.

Short Course on Object-Oriented Database, OGI Continuing Education, with M.J. Carey, S.B. Zdonik and G. Mitchell, Portland, 1992.

Short Course on Object-Oriented, Distributed and Intelligent Database Systems, with W. W. Chu, A. Cardenas, R. Elmasri and G.M. Lohman, UCLA Extension, 1993.

Short Course on Object-Oriented Database Management Systems, with M.J. Carey, S.B. Zdonik and G. Mitchell, UC Berkeley Extension, 1993-1994.

Tutorial, "Introduction to Object-Oriented Database Systems", OOPSLA '92, Vancouver, BC, 1992.

Seminar, "An Object-oriented Database Approach to Systems Integration," SINTEF, Oslo, Norway, 1994.

Tutorial, "Object-Oriented Databases," 7th International Hong Kong Computer Society Database Workshop, 1996.

Seminar, "Principles of Object-Oriented Databases," Tenth Brazilian School of Computation, Campinas, 1996.

Shortcourse, "Object-Interoperation Models," (3 times), Data-Intensive Systems Center, OGI, 1996 - 1998.

Shortcourse, "XML," (2 times), Data Intensive Systems Center, OGI, 1999.

Panels

Object-Oriented Database Systems and Knowledge Systems, First International Workshop on Expert Database Systems, 1985

Are Data Models Dead? 1986 ACM SIGMOD International Conference on Management of Data

User Interfaces and Database Management Systems, First International Conference on Expert Database Systems, 1986

Database Systems for Engineering Applications, Stanford University, 1987.

Object-Oriented Database Workshop: Implementation Aspects, 1987 ACM Conference on Object-Oriented Programming Systems, Languages and Applications, 1987.

Object-Oriented Database Systems, 1987 ACM Conference on Object-Oriented Programming Systems, Languages and Applications, 1987.

Constraint-Based Systems: Knowledge about Data, Second International Conference on Expert Database Systems, 1988.

Object-Oriented Database Systems, Fourteenth International Conference on Very Large Data Bases, 1988.

Has Theory Brought Anything to Database Systems and Will it in the Future? International Conference on Extending Database Technology, 1990.

Why are Object-Oriented Folks Producing Systems, While Deductive Folks are Producing Papers? International Conference on Extending Database Technology, 1990.

Database Manifestos, ACM-SIGMOD International Conference on Management of Data, 1990.

What Can We Do to Strengthen the Connection between Theory and System Builders, ACM-SIGMOD International Conference on Management of Data, 1992.

Storage Systems. DARPA High Performance Software First Annual PI Meeting, 1992.

What is the "True Path" to Integrating OO and Declarative Languages? ACM Workshop on Combining Declarative and Object-Oriented Databases, 1993.

How Should Declarative and Object-Oriented Databases be Combined? ACM Workshop on Combining Declarative and Object-Oriented Databases, 1993.

What's in the Future of Parallel Architectures? Supercomputing 93.

Research Directions for DOOD Technology, Third International Conference on Deductive and Object-Oriented Databases, 1993.

Strategies and Techniques to Manage System Evolution, Sixth International Workshop on Persistent Object Bases, 1994.

Panel Organizer, The Data You Won't Find in Databases, ACM-SIGMOD International Conference on Management of Data, 1995.

Data Mapping and Matching: Languages for Scientific Datasets, 5th International Workshop on Database Programming Languages, Gubbio, Italy, 1995.

Interoperating with Non-Database Data, Digital Libraries Workshop, CASCON '95, Toronto, 1995.

Data Mapping and Matching: Ad Hoc Art or Well-Grounded Science, NASA CESDIS 1996.

Metadata for Database Interoperations: Why and How? 1997 DBPL Workshop.

Where Are the Objects in Object-Relational Databases? ACM OOPSLA Symposium, 1997.

Future Directions in Database Research, International Conference on Data Engineering, 1998.

Information Technology Context, CSTB Workshop on Information Technology to Support Crisis Management, Washington, DC, 1998.

University R&D and Industry R&D and Commercialization, NSF Workshop on Information and Data Management, Los Angeles, CA, 1999.

Challenges and Opportunities in Scientific and Statistical Databases, 11th International Conference on Scientific and Statistical Database Management, 1999.

Processing Data Streams: Applications, Challenges and Approaches, 18th International Conference on Data Engineering, San Jose, CA, 2002.

Issues in National-Level Data Integration and Analysis. Stanford Database Workshop, Palo Alto, CA, 2002.

Data Streams: Fresh Current or Stagnant Backwater? ACM SIGMOD International Conference on Management of Data, Madison, WI, 2002.

Quality of Service; Benchmarking panels. Streams Winter Meeting (SWiM), Stanford, CA, 2003.

Producers and Consumers of Annotation Tools, National e-Science Center Workshop on Data Provenance and Annotation, Edinburgh, Scotland, 2003.

Papers Presented

Some of the papers listed here later appeared in final form in journals and books. The number in brackets at the end of each item is the number of the final version of the paper given in the section on Papers and Reports.

1. The complexity of some problems on subsequences and supersequences. Conference on Theoretical Computer Science, Waterloo, August 1977. [2]
2. A space-efficient algorithm for the lowest common ancestor problem and an application to finding negative cycles. Eighteenth Symposium on Foundations of Computer Science, October-November 1977. [3]
3. A note on the complexity of the superstring problem. With J.A. Storer. Conference on Information Science and Systems, March 1978. [4]
4. Representing databases in segmented name spaces. With V. Gligor. International Conference on Databases: Improving Reliability and Responsiveness, Technion, Haifa, August 1978. [5]
5. Minimum covers in the relational database model. Eleventh Symposium on the Theory of Computing, April-May 1979. [6]
6. Testing implication of data dependencies. With A.O. Mendelzon and Y. Sagiv. 1979 ACM SIGMOD Conference on Management of Data, May-June 1979. [7]
7. Adequacy of decompositions of relational databases. With A.O. Mendelzon, F. Sadri and J.D. Ullman. Workshop on Formal Bases for Databases, Toulouse, December 1979. [9]
8. Hysterical B-trees. With S.C. Salveter. Conference on Information Science and Systems, March 1980. [10]
9. Capturing more meaning in database systems. Conference on Computers in Small Business, Rutgers University, November 1980. [33]
10. DAGs as Lattices. Symposium on Knowledge Representation, IBM Systems Research Institute, New York, November 1980.
11. Connections in acyclic hypergraphs. With J.D. Ullman. XP2 Workshop on Relational Database Theory, June 1981 and ACM Symposium on Principles of Database Systems, March 1982. [19]
12. Fragments of Relations. With J.D. Ullman. XP2 Workshop on Relational Database Theory, June 1981. [20]

13. Revenge of the JD. With J.D. Ullman and M.Y. Vardi. Second ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, March 1983. [28]
14. Windows on the world. With D. Rozenshtein and D.S. Warren. 1983 ACM SIGMOD Conference on Management of Data, May 1983. [29]
15. Is Prolog a database system? XP4.5 Workshop on Database Theory, August 1983 and NYU Symposium on New Directions for Database Systems, May 1984. [34]
16. Representing roles in universal scheme interfaces. With D. Rozenshtein and J. Stein. 1984 IEEE International Conference on Data Engineering, April 1984. [31]
17. Implementing an object-oriented DBMS. With J. Stein. Object-Oriented Programming Workshop, IBM T. J. Watson Research Center, June 1986. [53]
18. Development of an Object-Oriented DBMS. With J. Stein, A. Otis and A. Purdy. XP 7.52 Workshop on Database Theory, August 1986. [45]
19. A logic for objects. Workshop on Foundations of Deductive Databases and Logic Programming, August 1986.
20. Why object-oriented databases can succeed where others have failed. International Workshop on Object-Oriented Databases, September 1986. [58]
21. Building interfaces with constraints. With R. K. Ege and A. Borning. The 2nd International Conference on Human-Computer Interaction, August 1987.
22. Describing and managing interactive tasks using Horn clauses. With M. Grossman. The 2nd International Conference on Human-Computer Interaction, August 1987.
23. Why database languages are a bad idea. Invited paper, Workshop on Database Programming Languages, Roscoff, France, September 1987. [55]
24. Computational objects in object-oriented data models. With J. Zhu, Second International Workshop on Database Programming Languages, Gleneden Beach, Oregon, June 1990. [65]
25. Mistaking identity. With J. Stein and T.L. Anderson, Second International Workshop on Database Programming Languages, Gleneden Beach, Oregon, June 1990. [67]
26. Object-oriented query processing in Revelation. With G. Graefe, Dagstuhl Seminar on Query Processing in Object-Oriented, Complex Object, and Nested Relation Databases, Germany, 1991. [80]
27. Specifying a database system to itself. Invited paper. Workshop on specifications of Database Systems, Glasgow, Scotland, July 1991. [78]
28. The DARPA persistent object bases initiative. DARPA Software Technology Conference, Los Angeles, April 1992.
29. The Revelation project: Query Processing in object-oriented databases. With S. Daniels, L. Fegaras, T. Keller, B. Vance, G. Graefe, B. McKenna, L. Shapiro. Oregon Advanced Computing Conference, Portland, OR, November 1992.
30. Computational Proxies: Modeling computational applications in object databases. With J.B. Cushing, M. Rao, D. Feller and M. DeVaney. AAAS '93, Boston, February 1993.
31. Specifying rewrite-based query optimizers in a reflective framework. With L. Fegaras and T. Sheard. ARPA Persistent Object Bases Workshop, Salt Lake City, May 1993.
32. Issues in distributed object assembly. With G. Graefe, L. Shapiro, S. Daniels, T. Keller and B. Vance. International Workshop on Distributed Object Management, Edmonton, Alberta, Canada, August 1992. [82]
33. Object data models for shared molecular structures. With J.B. Cushing, D.M. Hansen, G.D. Purvis III, R.A. Bair, D.M. DeVaney, D.F. Feller, M.A. Thompson. ASTM First International Symposium on Computerized Chemical Data Standards, Atlanta, May 1993. [86]

34. Processing OQL. ARPA Persistent Object Bases Program Workshop, Madison, May 1994.
35. Object databases as middleware. Workshop on Curricular Innovations to Promote Software Development Skills, Olympia, June 1996.
36. Approaching object/relational databases from the OODBMS side. The Object/Relational Summit, Burlingame, CA, October 1996.
37. Database desiderata for an XML query language. Invited Paper. W3C QL'98 - The Query Languages Workshop, Boston, MA, December 1998.
38. A petabyte in your pocket: Directions for Net Data Management. Invited talk. DIMACS Workshop Data Processing on the Web, New Brunswick, NJ, March 2000.
39. Automatic coordination and configuration of Net Data Management applications, RM2000 Workshop on Reflective Middleware, Palisades, NY, April 2000.
40. Mutant Query Plans. Invited talk. With V. Papadimos. OOPSLA '01 Workshop on Objects, XML and Databases, Tampa Bay, FL, October 2001. [125]
41. An Architecture for Superimposed Information. Invited talk. IRCS Workshop on Linguistic Databases, Philadelphia, PA, December 2001.
42. Exploiting Punctuation Semantics in Data Streams. Poster. With P. Tucker. 18th International Conference on Data Engineering, San Jose, CA, 2002.
43. Relational Database Theory at Princeton and Beyond. Computer Science Symposium in Honor of Jeffrey D. Ullman, Stanford, CA, 2002.

Invited Talks

1978

University of Maryland, Department of Computer Science
 University of Southern California, Department of Computer Science
 SUNY at Stony Brook, Department of Computer Science
 Columbia University, Department of Electrical Engineering and Computer Science
 University of California at Santa Barbara, Computer Science Program

1979

Bell Laboratories, Murray Hill

1980

IBM, Watson Research Center
 University of Maryland, Department of Computer Science
 IBM, Systems Research Institute
 IBM, San Jose Research Center
 Rutgers University, Newark, School of Business Administration

1981

Pennsylvania State University, Department of Computer Science
 Johns Hopkins University, Department of Electrical Engineering
 Harvard University, Division of Applied Sciences
 University of California at Santa Barbara, Department of Computer Science
 University of Michigan, Department of Computer Science
 Oregon Graduate Center
 Purdue University, Department of Computer Science
 Vanderbilt University, Department of Computer Science
 Bell Laboratories, Murray Hill

1982

University of Toronto, Computer Systems Research Group
Oregon State University, Department of Computer Science

1983

University of Oregon, Department of Computer and Information Science

1984

University of Washington, Department of Computer Science
Brandeis University, Department of Computer Science
Boston University, Department of Computer Science
IASI-CNR (Italian Computer Science Institute)/University of Rome
University of British Columbia, Department of Computer Science
Puget Sound Chapter of the ACM, Seattle

1985

Northwest Computers in Education Conference, Eugene
Microelectronics and Computer Technology Corporation (MCC)
University of Illinois, Department of Computer Science
Mentor Graphics Corporation
Southern Methodist University, Department of Computer Science
Willamette Valley Chapter of the ACM

1986

Oregon State University, Department of Computer Science
Cullinet Software, Inc.
Wang Institute
Brown University, Department of Computer Science
New York University, Graduate School of Business Administration
Rutgers University, Department of Computer Science
University of Pennsylvania, Department of Computer and Information Science
Hewlett-Packard Laboratories, Palo Alto, California
Stanford University, Department of Computer Science

1987

University of California at San Diego, Department of Electrical Engineering and Computer Science
Digital Equipment Corp., Colorado Springs, Advanced Database Development
CRAI Seminar on Advanced Data and Knowledge Base Systems, Capri, Italy
Massachusetts Institute of Technology, Department of Computer Science
GiP Altair, Rocquencourt, France
University of Wisconsin at Madison, Department of Computer Science,
Distinguished Lecture Series in Database Systems

1988

Pike's Peak Chapter of the ACM
Florida International University, Department of Computer Science
Apple Computer, Inc.
Oregon Computer Consultants Association

1989

State University of New York at Stony Brook, Department of Computer Science
Digital Equipment Corporation, Cambridge Research Laboratory.

GIP Altair, O2 Day, Paris, France
INRIA, VERSO-SABRE-Altair joint seminar, Rocquencourt, France
University of Grenoble, IMAG research group, France

1990

University of Zurich, Department of Computer Science, Switzerland
ETH Zurich (Swiss Federal Technical University), Department of Computer Science,
Switzerland
University of Aachen, School of Applied Mathematics, Institute for Computer
Science, Germany
IBM Almaden Research Center
University of Colorado, Boulder, Department of Computer Science
Digital Equipment Corp., Colorado Springs, Advanced Database Development
US West Advanced Technologies, Edgewood, Colorado
Texas Instruments, Information Technologies Laboratory
Senter for Industriforskning (Center for Industrial Research), Oslo, Norway
University of Helsinki, Department of Computer Science, Finland

1991

Battelle Pacific Northwest Laboratories, Molecular Sciences Research Center
Saturday Academy, Beaverton, Oregon
University of Colorado, Colorado Springs, DEC Distinguished Lecture Series
University of Aberdeen, Scotland, Department of Computing Science
University of Glasgow, Scotland, Department of Computing Science, Distinguished Lecture
Series
Structured Development Forum, Portland, Oregon
University of Texas, Austin, Distinguished Lecture Series in Systems Research

1992

Boeing Computer Services, Advanced Technology Center, Seattle, Washington
University of Denver, Colorado Advanced Software Institute
The Evergreen State College, Olympia, Washington
Oregon State University, Department of Oceanography, Corvallis, Oregon

1994

University of Toronto, Canada, Department of Computer Science, Distinguished Lecture Series
University of Wisconsin - Madison, Department of Computer Science
Massachusetts Institute of Technology, Department of Computer Science
IEEE Computer Society, Portland, Oregon
University of Hamburg, Germany, Department of Informatics

1995

Brown University, Department of Computer Science
Pittsburgh Supercomputing Center
The Evergreen State College, Olympia, Washington
NASA Center of Excellence in Space Data and Information Sciences (CESDIS)
University of Utah, Department of Computer Science
XEROX, Palo Alto Research Center (PARC)

1996

Southern Methodist University, School of Engineering and Applied Science
Hong Kong University of Science and Technology, Department of Computer Science
Informix, Portland, Oregon

The Evergreen State College, Olympia, Washington

1997

National Science Foundation, Division of Information, Robotics and Intelligent Systems
University of Wisconsin, Database Affiliates
University of Wisconsin, Computer Sciences Department

1998

University of Pennsylvania, Department of Computer and Information Science
Oregon Graduate Institute, ADC Kentrox Perspectives on Technology Series
Case Western Reserve University, Department of Computer and Information Science.

1999

INRIA, VERSO group, Rocquencourt, France.

2000

University of Washington, Department of Computer Science
GemStone Systems, Beaverton, Oregon.
Intel Architecture Labs, Hillsboro, Oregon
NASA Ames Research Center, Moffett Field, California

2001

NASA Johnson Space Center, Houston, Texas

2002

University of Edinburgh, School of Informatics, Centre for Cognitive Science
University of Edinburgh, School of Informatics, Laboratory for the Foundations of Computer Science
University of British Columbia, Department of Computer Science

2003

New England Database Seminar, Brandeis University, Waltham, Massachusetts
Microsoft Research, Redmond, Washington

2005

Lawrence Livermore National Laboratory, Center for Applied Scientific Computing
University of California, Riverside, Department of Computer Science & Engineering

2007

University of California, Davis, Department of Computer Science
Northwest Database Society, Seattle, Washington

2008

Microsoft Research, Redmond, Washington
New England Database Seminar, Brandeis University, Waltham, Massachusetts
University of Pittsburgh, Department of Computer Science

2009

Microsoft Research, Redmond, Washington
Southern Methodist University, Department of Computer Science and Engineering