

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.: 10,585,850

Issued: March 10, 2020

Application No.: 14/970,044

Filed: December 15, 2015

Title: Automatic Real-Time File Management Method And Apparatus

DECLARATION OF DAVID MAIER

DROPBOX, INC., ET AL.
EXHIBIT 1004

TABLE OF CONTENTS

	Page(s)
I. INTRODUCTION AND ENGAGEMENT	6
II. BACKGROUND AND QUALIFICATIONS	6
III. MATERIALS CONSIDERED AND INFORMATION RELIED UPON REGARDING '850 PATENT.....	10
IV. UNDERSTANDING OF PATENT LAW	12
V. PERSON OF ORDINARY SKILL IN THE ART	16
VI. TECHNOLOGY OVERVIEW.....	17
A. File-Based Backup and Recovery Systems Were Known Since At Least 1978	17
B. From The Perspective Of The Backup/Restore Application, There Was No Distinction Between Local and Remote File Storage.....	18
VII. THE '850 PATENT.....	20
A. The '850 Patent's Specification	20
B. Claim Similarities Analysis.....	24
VIII. GROUND 1: CLAIMS 1-21 ARE OBVIOUS OVER SCHNEIDER.....	31
A. Schneider.....	31
1. Claims 1 & 10	37
a) [1.1]/[10.1].....	37
b) [1.2.i]/[10.2.i].....	38
c) [1.2.ii]/[10.2.ii]	39
d) [1.2.iii]/[10.2.iii]	46
e) [1.3]/[10.3].....	46

f)	[1.4]/[10.4].....	49
2.	Claims 2 & 11	50
3.	Claim 3.....	51
4.	Claims 4 & 12	53
5.	Claims 5 & 13	54
6.	Claims 6 & 14	55
a)	[6.1/14/1]	55
7.	Claims 7 & 15	57
8.	Claims 8 & 16	57
9.	Claims 9 & 17	58
10.	Claim 18.....	59
a)	[18.1].....	59
b)	[18.2].....	60
c)	[18.3.i], [18.3.ii], [18.4], [18.5]	61
d)	[18.3.iii]	61
11.	Claim 19.....	62
a)	[19.1].....	62
b)	[19.2].....	62
c)	[19.3].....	62
d)	[19.4].....	63
a)	[19.5].....	63
12.	Claims 20 & 21	64

IX.	GROUND 2: CLAIMS 1-21 ARE OBVIOUS OVER SCHNEIDER IN VIEW OF CRAWFORD	65
X.	GROUND 3: CLAIMS 1-8, 10-16, AND 18-21 ARE OBVIOUS OVER SWEAT.....	69
A.	Sweat	69
1.	Claims 1 & 10	73
a)	[1.1] / [10.1]	73
b)	[1.2.i] / [10.2.i]	74
c)	[1.2.ii] / [10.2.ii]	75
d)	[1.2.iii] / [10.2.iii]	77
e)	[1.3] / [10.3]	77
f)	[1.4] / [10.4]	79
2.	Claims 2 & 11	81
3.	Claim 3	82
4.	Claims 4 & 12	84
5.	Claim 5 & 13	84
6.	Claim 6 & 14	85
a)	[6.1/14.1]	85
b)	[6.2/14.2]	86
7.	Claims 7 & 15	87
8.	Claims 8 & 16	87
9.	Claim 18	88
a)	[18.1]	88
b)	[18.2]	88

c)	[18.3.i], [18.3.ii], [18.4], [18.5]	89
d)	[18.3.iii]	89
10.	Claim 19	90
a)	[19.1]	90
b)	[19.2]	90
c)	[19.3]	90
d)	[19.4]	91
e)	[19.5]	91
11.	Claims 20 & 21	93
XI.	GROUND 4: CLAIMS 9 AND 17 ARE OBVIOUS OVER SWEAT IN VIEW OF SCHNEIDER	93
1.	Claims 9 & 17	93

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. 10,585,850 (hereinafter “the ’850 Patent”).

2. I am being compensated at my usual and customary hourly rate for the time I 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 1-21 (each a “Challenged Claim” and collectively the “Challenged Claims”) of the ’850 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, U Penn, U 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 '850 PATENT

15. In preparing this declaration, I have reviewed the materials discussed herein when offering my opinions, including the following Exhibits that I understand are being referenced in and filed with the IPR which my declaration accompanies:

LIST OF EXHIBITS

No.	Description
1001	U.S. Patent No. 10,585,850 (“’850 Patent”)
1002	File History of U.S. Patent No. 10,585,850 (“’850 FH”)

No.	Description
1003	U.S. Provisional Application No. 60/234,211 (“ 211 Provisional ”)
1005	PCT Publication No. WO 99/12101 (“ Schneider ”)
1006	U.S. Patent No. 7,062,532 (“ Sweat ”)
1007	U.S. Provisional Application No. 60/126,170 (“ Sweat Provisional ”)
1008	U.S. Patent No. 5,953,729 (“ Cabrera ”)
1009	U.S. Patent No. 5,771,354 (“ Crawford ”)
1010	Joost S.M. Verhofstad, <i>Recovery Techniques for Database Systems</i> , Computing Surveys, Vol. 10, No. 2 (June 1978) (“ Verhofstad ”)
1011	Ann L. Chervenak, et al., <i>Protecting File Systems: A Survey of Backup Techniques</i> , Proceedings of Joint NASA and IEEE Mass Storage Conference (1998) (“ Chervenak ”)
1012	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 ”)
1013	Marc Farley, <i>Building Storage Networks</i> (McGraw-Hill 2000) (excerpts) (“ Farley ”)
1014	Microsoft Computer Dictionary, Fifth Edition (Microsoft Press 2002) (excerpts)
1015	Dropbox, Inc.’s Proposed Claim Constructions, Case No. 6:22-cv-01142-OLG (W.D. Tex.), July 7, 2023
1016	Datanet LLC’s Proposed Claim Constructions, Case No. 6:22-cv-01142-OLG (W.D. Tex.), July 7, 2023
1017	Dropbox, Inc.’s Opening Claim Construction Brief, Case No. 6:22-cv-01142-OLG (W.D. Tex.), July 26, 2023
1018	Datanet LLC’s Response to Opening Claim Construction Brief, Case No. 6:22-cv-01142-OLG (W.D. Tex.), August 16, 2023

No.	Description
1019	Rebuttal Decl. of Dr. Sam Malek in Support of Datanet LLC's Response to Opening Claim Construction Brief, Case No. 6:22-cv-01142-OLG (W.D. Tex.), August 16, 2023
1020	Dropbox, Inc's Reply Claim Construction Brief, Case No. 6:22-cv-01142-OLG (W.D. Tex.), August 31, 2023
1021	Joint Claim Construction Statement, Case No. 6:22-cv-01142-OLG (W.D. Tex.), September 18, 2023
1022	List of Challenged Claims
1023	Comparison Chart of Claims

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 '850 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 '850 patent or any assignee of the '850 patent that any secondary considerations tend to rebut the obviousness of any Challenged Claim of the '850 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 '850 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. The technology overview provided above (*infra* Section VI) reflects some of the information within a POSITA's knowledge.

31. Having reviewed and considered the '850 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 '850 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 PhD 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.

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. TECHNOLOGY OVERVIEW

A. File-Based Backup and Recovery Systems Were Known Since At Least 1978

33. Systems for providing redundant backup for computer data files, including storage of multiple versions of a given file and capability to recover prior file versions were well known in the art at the time of the purported invention. For example, as early as 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.” EX1010, 1 (Verhofstad, J. *Recovery Techniques for Database Systems*, Computing Surveys, Vol. 10, No. 2). Verhofstad describes a “Backup/current version” system for files,

in which “[t]he files containing the present values of existing files form the current version of the database. Files containing previous values form a consistent, backup version of the database. Backup versions can be used to restore files to previous values.” *Id.*, 6; *see also* EX1011 (Chervenak, A. et al., *Protecting File Systems: A Survey of Backup Techniques*), 2-3 (“Backup software can operate either on files or on physical disk blocks. **File-based backup** systems understand file structure and copy entire files and directories to backup devices. ... Then backup software writes each file contiguously to the backup medium; this facilitates fast individual file recovery.”) (emphasis original)¹.

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

34. It was also well known to access storage devices, including those used for backup, over a network, allowing such storage devices to be located remotely and accessed by a local computer. By the 1990s, Network File Systems (such as NFS, first released by Sun Microsystems in 1984) erased the difference between how users accessed a local file and a remote file. EX1012, (showing first release); EX1009, 3:64-66 (“Drives on a remote system defined using NFS and RFS 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

¹ Emphases throughout are added, unless otherwise noted.

location of the file, a property known as “location transparency.” Therefore, by 1999, with respect to backup applications, there was no significant technical barrier between storing a file or its backup version locally and storing it remotely.

35. This lack of a meaningful distinction is reflected in the prior art. For example, the Sweat reference relied on herein disclosed a distributed network architecture connecting local client computers to remote file servers and file storage over the Internet. EX1006, 3:12-26. Such an architecture allows a user to “view or download files from anywhere in the world to work on them.” *Id.*, Abstract, 2:52-53. And Cabrera, which issued in 1999 from an application filed in 1997, described that then-state of the art backup and archival “systems typically copy 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.” EX1008, 1:38-42. Further, the “[a]rchive or backup storage may comprise any storage medium suitable for such a purpose. The 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.” EX1008, 3:3-7. Cabrera explains that “commonplace” remote networking systems may utilize a wide area network (WAN) and that “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.

VII. THE '850 PATENT

36. The '850 Patent issued March 10, 2020, from an application filed December 15, 2015 (the "'044 Application") and alleges priority through several continuation applications to a non-provisional application filed September 21, 2001, and also claims the benefit of U.S. Provisional Application No. 60/234,221 ("'221 Provisional"), filed September 21, 2000.

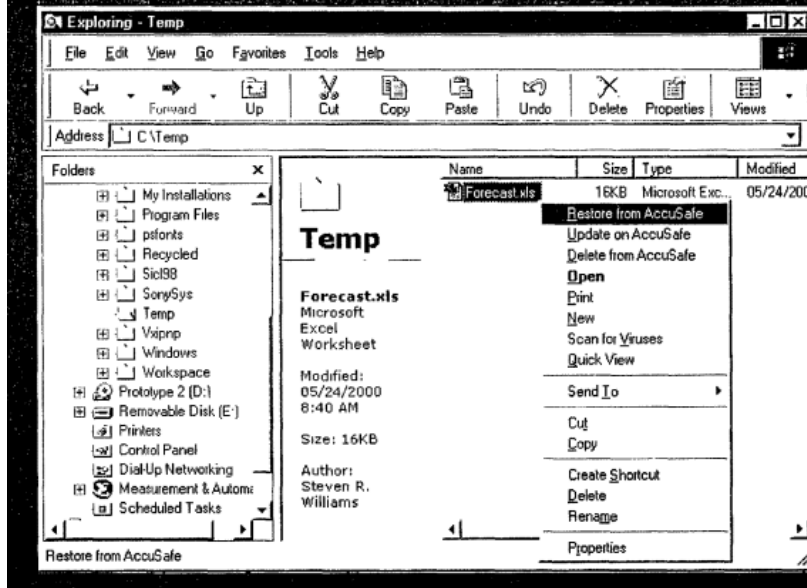
A. The '850 Patent's Specification

37. While the claims of the '850 Patent relate to "restoring a file to a previous version," (EX1001, claims 1, 10, 18, 19) the written description of the '850 patent does not describe restoring a file to a previous version. Nor does the written description describe any of the claimed "presenting" or "retrieving" steps.

38. The '850 Patent incorporates the '221 Provisional by reference. EX1001, 1:4-10. The '221 Provisional includes presentation slides for a prototype version of a product called "AccuSafe™" (e.g., EX1003, 3-60)² that supposedly allows a user to "right click on the file he's been working with, then select AccuSafe's restore function to retrieve the lost versions," as shown in the slide below. *Id.*, 31 (excerpt).

² Unless otherwise noted, '221 Provisional cites are to the page numbers added to the exhibit filed herewith.

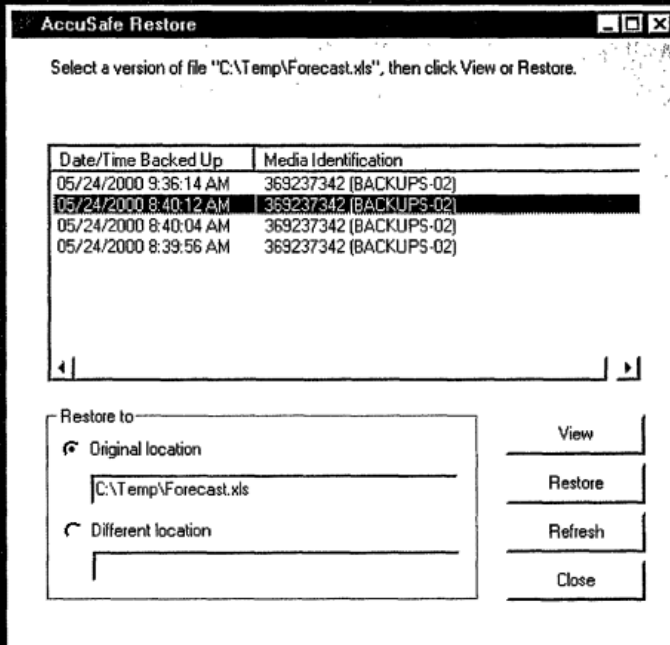
Example use-case scenario



The user realizes that he's overwritten his file. Via explorer (or any other file selection dialog), the user can right-click on the file he's been working with, then select AccuSafe's restore function to retrieve the lost versions.

39. As shown in the slides below, the '221 Provisional states that the AccuSafe product displays a list of captured revisions of the desired file and a user selects a previous file version to display that file using the same application that created the file. *Id.*, 32-34. The user may iteratively view multiple captured versions of the file until the user finds the correct prior version that the user wanted. *Id.* (excerpted below).

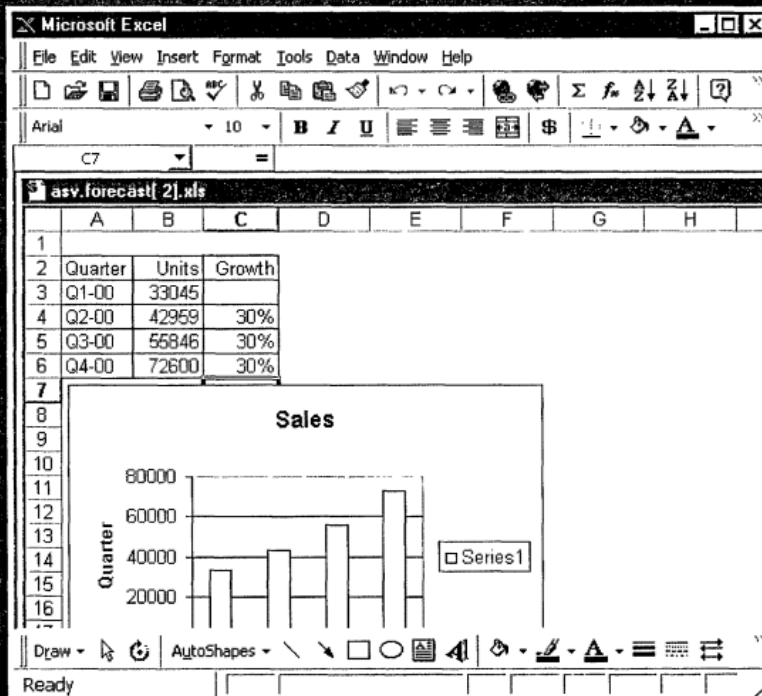
Example use-case scenario



AccuSafe displays the list of revisions it has captured.

The user selects a version from this list, then clicks the View button to determine if this is the version he wishes to restore...

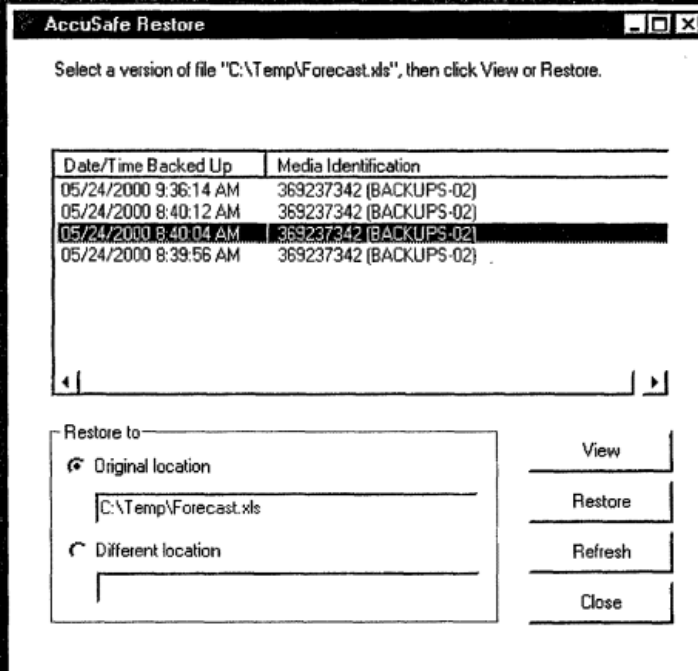
Example use-case scenario



AccuSafe displays the revision selected, using the application that created the file.

If this isn't the version that the user wanted...

Example use-case scenario

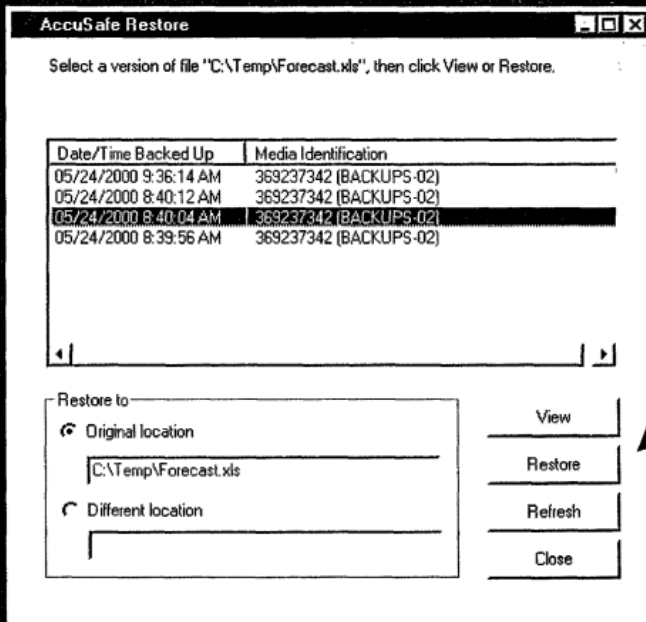


The user selects another revision to view...

Note: Most backup applications don't have such a preview capability, which significantly enhances ease-of-use.

40. Finally, as shown below, once a user verifies the prior file version to restore and clicks "Restore," the product overwrites the current version of the file with the prior version selected by the user, while saving a copy of the replaced current version. *Id.*, 36 (excerpt).

Example use-case scenario



The user has verified this to be the proper revision to restore, so the user clicks Restore to retrieve his file.

B. Claim Similarities Analysis

41. Various limitations of the claims use substantially the same or identical claim language. EX1023 presents a chart of the '850 patent's claims, with comparable limitations in the same row, showing the similarity between claim limitations.

42. Independent claims 1 and 10 are substantially similar, as shown in the table below. The only difference between the two claims is that claim 1 includes additional language regarding a "computational machine" with the additional language shown [**bracketed and bolded**]. Accordingly, all limitations of claim 10

are included in claim 1 and any reference or combination that includes all elements of claim 1 would necessarily also do so for claim 10.

Claim 1	Claim 10
[1.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:	[10.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:
[1.2] (A) presenting information for a collection of one or more previous versions of the file, the information for the collection including information indicative of at least one or more of previous versions of the file, wherein a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location, the restorable representation having at least information required for recovering the version V [by a computational machine with the current version being accessible] , the remote storage location being accessible through a network;	[10.2] (A) presenting information for a collection of one or more previous versions of the file, the information for the collection including information indicative of at least one or more of previous versions of the file, wherein a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location, the restorable representation having at least information required for recovering the version V, the remote storage location being accessible through a network;
[1.3] (B) responsive to a selection to preview a selected previous version of the file based on the presented information for the collection of (A), presenting a presentable representation of the selected previous version, the selected previous version being one of the previous versions of the file in the presented information for the collection, the presentable representation having at least	[10.3] (B) responsive to a selection to preview a selected previous version of the file based on the presented information for the collection of (A), presenting a presentable representation of the selected previous version, the selected previous version being one of the previous versions of the file in the presented information for the collection, the presentable representation having at least

information required for presenting at least a portion of the selected previous version [by the computation machine with the current version being accessible; and]	information required for presenting at least a portion of the selected previous version;
[1.4] (C) responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version [by the computational machine] as the current version on the local storage location, the selected previous version available from the restorable representation of the selected previous version.	[10.4] (C) responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version as the current version on the local storage location, the selected previous version available from the restorable representation of the selected previous version.

43. Dependent claims 2 and 4-9 (dependent from claim 1) are identical to respective dependent claims 11-17 (dependent from claim 10), as shown in the table below, with the sole difference being the independent claims from which they depend.

Claim 1 Dependent Claims	Claim 10 Dependent Claims
[2] The method of claim 1, wherein, for the presenting of (B), the presentable representation is presented using an application that created the file or the selected previous version.	[11] The method of claim 10, wherein, for the presenting of (B), the presentable representation is presented using an application that created the file or the selected previous version.
[4] The method of claim 1, wherein the information includes at least two previous versions of the file.	[12] The method of claim 10, wherein the information includes at least two previous versions of the file.

[5] The method of claim 1, wherein the information for the collection includes information indicative of at least one or more of a timestamp, an original file location, and a comment of the previous versions or the file.	[13] The method of claim 10, wherein the information for the collection includes information indicative of at least one or more of a timestamp, an original file location, and a comment of the previous versions or the file.
[6.1] The method of claim 1, further comprising the steps of: (B2) after the presenting of (B) and responsive to a selection to not restoring the previous version and responsive to a second selection of a second previous version of the previous versions, version presenting a second presentable representation of the selected second previous version, the second presentable representation having at least information required for presenting at least a portion of the selected previous version; and	[14.1] The method of claim 10, further comprising the steps of: (B2) after the presenting of (B) and responsive to a selection to not restoring the previous version and responsive to a second selection of a second previous version of the previous versions, version presenting a second presentable representation of the selected second previous version, the second presentable representation having at least information required for presenting at least a portion of the selected previous version; and
[6.2] (C2) responsive to a selection to restore the second selected previous version, retrieving the restorable representation of the second selected previous version from the remote storage location and storing the second selected previous version as the current version on the local storage location, the second selected previous version available from the restorable representation of the second selected previous version.	[14.2] (C2) responsive to a selection to restore the second selected previous version, retrieving the restorable representation of the second selected previous version from the remote storage location and storing the second selected previous version as the current version on the local storage location, the second selected previous version available from the restorable representation of the second selected previous version.
[7] The method of claim 1, wherein the restorable representation of the selected previous version comprises one or more of a compressed and encrypted version of the selected previous version.	[15] The method of claim 10, wherein the restorable representation of the selected previous version comprises one or more of a compressed and encrypted version of the selected previous version.

[8] The method of claim 1, wherein the restorable representation of the selected previous version comprises an incremental change to the file.	[16] The method of claim 10, wherein the restorable representation of the selected previous version comprises an incremental change to the file.
[9] The method of claim 1, wherein the method has a substantially imperceptible impact on system performance from a user's point of view by delaying one or more operations to prevent a perceptible impact on system performance from a user's point of view.	[17] The method of claim 10, wherein the method has a substantially imperceptible impact on system performance from a user's point of view by delaying one or more operations to prevent a perceptible impact on system performance from a user's point of view.

44. Independent claims 1 and 18 are also substantially similar, with the only differences being claim 18's additional element [18.2], internal numbering of steps, and minor language in element [1.2]. Different language between the two claims is shown in the table below [**bracketed and bolded**]. Accordingly, all limitations of claim 18, except those in element [18.2], are included in claim 1 and any reference or combination that includes all elements of claim 1 would necessarily also do so for all elements claim 18 except for element [18.2].

Claim 1	Claim 18
[1.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:	[18.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:
	[18.2] [(A) iteratively transmitting a representation of at least a portion of a current version of a file to a remote storage location for archiving the current version

	responsive to a determination of an availability of the remote storage location for accepting a transmission of the representation, the current version being accessible at a computational machine, the remote storage location being accessible through a network;]
[1.2] [(A)] presenting information for a collection of one or more previous versions of the file, the information for the collection including information indicative of at least one or more of previous versions of the file, wherein a restorable representation of each version, V, of the previous versions, is retrievable from [a] remote storage location, the restorable representation having at least information required for recovering the version V by [a] computational machine [with the current version being accessible] , the remote storage location being accessible through a network;	[18.3] [(B)] presenting information for a collection of one or more previous versions of the file, the information for the collection including information indicative of at least one or more of previous versions of the file, wherein a restorable representation of each version, V, of the previous versions, is retrievable from [the] remote storage location, the restorable representation having at least information required for recovering the version V by [the] computational machine, the remote storage location being accessible through a network;
[1.3] [(B)] responsive to a selection to preview a selected previous version of the file based on the presented information for the collection of [(A)], presenting a presentable representation of the selected previous version, the selected previous version being one of the previous versions of the file in the presented information for the collection, the presentable representation having at least information required for presenting at least a portion of the selected	[18.4] [(c)] responsive to a selection to preview a selected previous version of the file based on the presented information for the collection of [(B)], presenting a presentable representation of the selected previous version, the selected previous version being one of the previous versions of the file in the presented information for the collection, the presentable representation having at least information required for presenting at least a portion of the selected

previous version by the computation machine with the current version being accessible; and	previous version by the computation machine with the current version being accessible; and
[1.4] [(C)] responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version by the computational machine as the current version on the local storage location, the selected previous version available from the restorable representation of the selected previous version.	[18.5] [(D)] responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version by the computational machine as the current version on the local storage location, the selected previous version available from the restorable representation of the selected previous version.

45. Dependent claims 20 and 21 are substantially the same. As shown in the table below, the only differences between the claims (shown [**bracketed and bolded**]) do not affect their scope. Accordingly, the claims are meaningfully the same and my analysis herein treats them as such.

Claim 20	Claim 21
[20] The method of claim 19, [wherein said method comprises] decompressing said selected revision of said file prior to (d) and decompressing said selected another revision of said file prior to (f).	[21] The method of claim 19, [further comprising the step of] decompressing said selected revision of said file prior to [step] (d) and decompressing said selected another revision of said file prior to [step] (f).

46. The preambles for all four independent claims 1, 10, 18, and 19 are identical: “A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:.”

VIII. GROUND 1: CLAIMS 1-21 ARE OBVIOUS OVER SCHNEIDER

47. As explained below, it is my opinion that claims 1-21 are obvious over Schneider.

A. Schneider

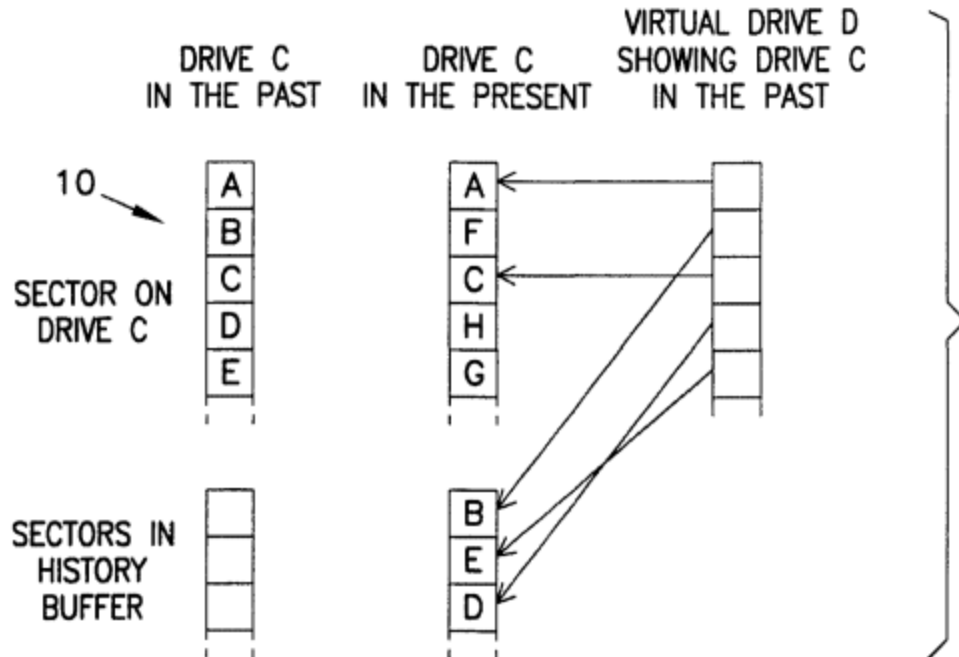
48. Schneider (EX1005) published March 11, 1999, and thus qualifies as prior art under at least Sections 102(a-b).

49. Schneider is analogous art to the '850 Patent because it is from the same field of endeavor. The '850 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." EX1005, 1:15-17. Thus, the '850 Patent and Schneider are both within the same field of endeavor of computer file storage, management, and recovery.

50. With Schneider's technology, "if the information on disk changes and it is determined the original state was desired, a user uses the backup to recover this original state." Id., 1:28-2:1. Schneider provides examples of such scenarios, including "a computer system 'crashing' during an update of a piece of information, thus leaving it in neither the original or 'new' state, the user changing information only later to desire to restore (or just reference) the original state, a computer virus altering information, or a file being deleted accidentally." Id., 2:6-10.

51. Among its “essential components,” Schneider discloses the ability to preview a prior state of information: “the ability to simulate a time reverted disk while at the same time allowing the user to continue using the current disk (thus, for example, allowing you to copy forward information from the past into the current).” *Id.*, 11:6-17. The user may also “completely revert a disk to a prior state in time.” *Id.* It also provides a “management process” that “[p]rovid[es] utilities that operate on the saved information to determine available versions of a file.” *Id.*

52. Schneider’s process uses a dedicated storage structure called a “history buffer.” “The history buffer works by recording the original state of sectors on a disk prior to being changed” and recording the time associated with that change of state. *Id.*, 12:1-4. The history buffer allows a user to recover data of a Drive C by looking back to a prior reference time by creating a simulated Drive D that combines unchanged data from the current Drive C with archived data from the history buffer. *Id.*, 15:20-29. Then, if the user wishes to revert the Drive C back to its state as it was at the prior reference time, the data in Drive D is copied over to Drive C so as to take the place of the present drive state. *Id.*, 16:9-21. This process of data recovery is depicted in Figure 2, reproduced below, where the constructed Drive D combines unaltered data (sectors A and C) with archived data (sectors B, E, and D) so as to reconstruct Drive C as it was in the past at the reference time (consisting of A, B, C, D, and E). *Id.*, Fig. 2.

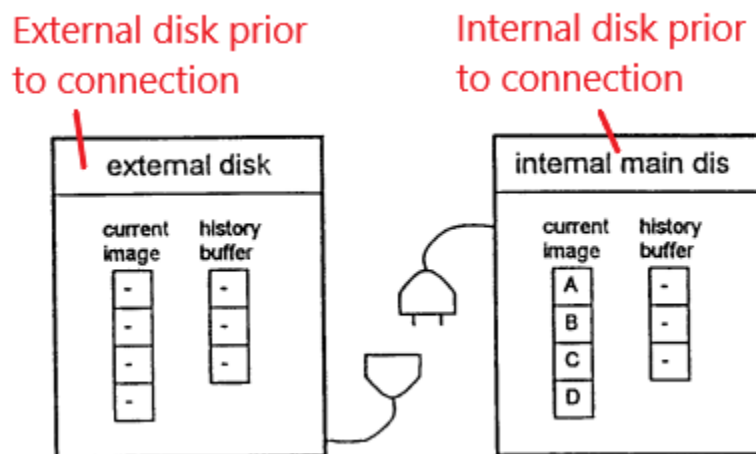
**FIG. 2****EX1005, FIG. 2**

53. Schneider discloses that its techniques are applicable to file-level data recovery, allowing a user to select a prior version of a specific file and recover only the data for that file at a specific point in time. *Id.*, 23:29-24:6. This file-level implementation is “based on the principal of saving original states of recent changes” and allows the operating system to “simply present the user with a list of prior states for a given file, and allow any to be copied forward.” *Id.*, 24:7-12. For example, Schneider discloses “an extension to [Microsoft] Explorer wherein the user can right click on a specific file and have the option to view a list of old versions of the file.” EX1005, 60:3-7. It also discloses a process that “shows all available old versions of

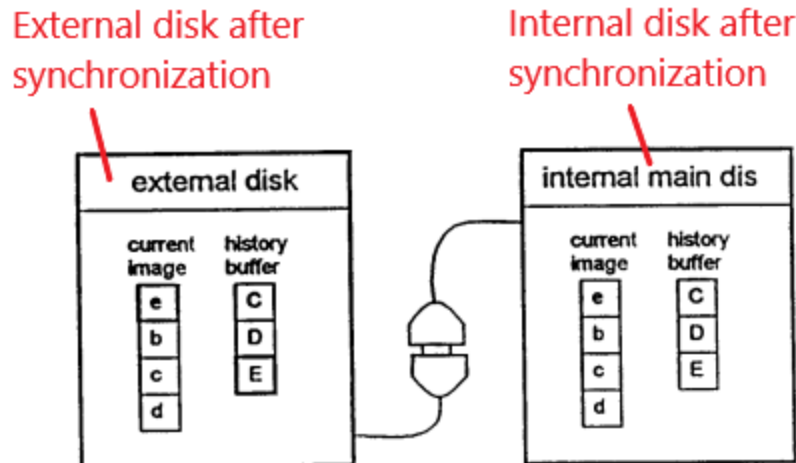
files and directories ... in the form of a hierarchy.” *Id.*, 60:15-20. Schneider also “allows the user to not only know of the existence of an old version of a file, but to view its contents without actually formally retrieving the file,” e.g., by copying the file to a temporary directory that is later purged. *Id.*, 60:21-29; *see also id.*, 130:10-17 (disclosing application “supplementing the directory and file presentation means of an OS, and allowing the user to view a list of earlier versions of a file. A selection can be made from the list and the recovered file either replaces the current version or is copied to a new file.”).

54. Finally, 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. *Id.*, 112:26-114:7. For example, “each PC has an internal disk for which a portion represents OS visible data” and the internal disk may also have “zero” or more space “set aside for historic data.” *Id.*, 113:9-13. It also teaches an external disk with “additional storage used for historic states.” *Id.*, 113:15-19. In this manner, compared to the internal disk, “the external disk may be able to reach further back in time in re-creating prior states.” *Id.*, 113:19-21. Schneider teaches that the user’s “PC’s software would divert to and take advantage of an external disk that had more historic information.” *Id.*, 113:26-28. An example of this storage of files to local and external disks is shown in Figures 47A and 47I, reproduced and annotated below. 47A depicts the internal and external

disks before being connected, while 47I shows the state of the disks after they have been fully synchronized. *Id.*, 112:1-3, 112:24. As shown, over time the external disk mirrors the same data contained in the internal disk, thereby providing an effective backup of the disk. While 47A-47F and 47H-47I depict synchronization of disk pages between an internal drive and external drive (*id.*, 112:1-24), Schneider explains that its “present invention can be implemented to work either with managing data at the disk sector level or the file level” (*id.*, 10:25-27). A POSITA would have therefore understood that use of an external disk would have been equally applicable to Schneider’s file-level implementation. *See id.*, 28:2-6 (explaining that in the file-level implementation, data pages in the history buffer “would correspond to file information instead of disk sectors”).



EX1005, FIG. 47A (annotated)



EX1005, FIG. 47I (annotated)

55. Based on these and other teachings in Schneider, a POSITA would have found it obvious to implement a system with the current versions of files stored locally on the user's PC and the historic versions on an external drive accessible via a network. EX1005, 10:25-26, 24:1-2. This obvious system would employ the software taught by Schneider to allow users to, for a given file, (i) access the historic data on the external drive and "view a list of old versions of the file" (*id.*, 113:26-28, 60:3-5); (ii) for a given previous file version, preview that version, e.g., "view its contents without actually formally retrieving the file" (*id.*, 60:21-29; *id.*, 16:1-5, 130:10-17); and (iii) "replace[] the current version" of the file with the recovered earlier version of the file (*id.*, 130:10-17). Schneider describes, and a POSITA would recognize, the compatibility of each of these features and the POSITA would have no issue implementing a file recovery system that included each of them.

1. Claims 1 & 10

56. As demonstrated above at paragraph 42 and in Exhibit 1023, the limitations of claims 1 and 10 are identical except that Claim 1 includes certain additional limitations not included in claim 10. Accordingly, my declaration below presents and analyzes the limitations of these claims together and identifies the additional limitations contained in claim 1 and not claim 10 [*bracketed and in italics*].

a) [1.1]³/[10.1]

[1.1]/[10.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:

57. To the extent the preamble is limiting, Schneider discloses it.

58. Schneider discloses “allowing the user to view a list of earlier versions of a file,” select an earlier version, and then “replace[] the current version” with the recovered earlier version of the file. EX1005, 130:10-17; *see also* 23:29-24:12 (the system “present[s] the user with a list of prior states for a given file, and allow[s] any to be copied forward”); *id.*, 31:19-24 (discussing ability “to locate and recover versions only of desired files”). Schneider teaches that a file’s current version is

³ Reference numbers in the format of [claim#.limitation#] are added throughout for ease of reference.

available at a local storage location, such as the internal disk of the user's PC as shown in FIGS. 47A-G. *See id.*, 113:9-10 (“each PC has an internal disk for which a portion represents OS visible data”); *see also* 112:1-114:5 (describing state of current version on internal disk, over time); FIGS. 47A-47F, 47H-47I (showing “current image” on “internal main” disk); 110:13-14 (describing “changes that were made to the current image” on the main internal disk); FIG. 47G; 135:6-8, FIG. 63. Schneider explains that the internal disk contains non-historic system data, such as information visible to the operating system. *Id.*, 113:9-13.

b) [1.2.i]/[10.2.i]

[1.2.i]/[10.2.i] (A) presenting information for a collection of one or more previous versions of the file, the information for the collection including information indicative of at least one or more of previous versions of the file,

59. Schneider discloses presenting information indicative of previous versions of a file, e.g., using an extension to the “industry standard Windows95 Explorer utility” that allows the user to “right click on a specific file and have the option to view a list of old versions of the file.” EX1005, 59:27-60:5; *see also id.*, 58:20-23 (describing history log portion of history buffer), 58:24-27 (describing general log portion of history buffer), 130:10-17 (describing system “allowing the user to view a list of earlier versions of a file”), 31:19-24 (user may “select a file and request the present invention's software to locate all available versions of that file,

over time”). The presented list of old file versions is information indicative of at least one or more previous versions of the file.

60. In order to provide such previous file versions, Schneider discloses capturing and storing a collection of one or more previous versions of a file. Specifically, “[t]he file level implementation of present invention automatically keeps backup copies of files prior to their being changed in a circular system.” *Id.*, 24:12-14. Schneider performs such automatic backup by storing data to the history buffer in response to an operation, such as a write operation. For example, in response to a write request for a file, “[t]he history buffer works by recording the original state of sectors on a disk prior to being changed.” *Id.*, 12:1-212:5-8, 24:7-10, 24:12-14, 24:21-23, 36:12-17. Accordingly, a POSITA would recognize that the data backup system of Schneider would store in its history buffer multiple versions of a given file, where multiple such versions exist, with each version corresponding to a write request altering the file.

c) [1.2.ii]/[10.2.ii]

[1.2.ii]/[10.2.ii] wherein a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location,

61. The file-level data recovery implementation taught by Schneider maintains a restorable representation of each version presented to the user. For example, a “selection can be made from the list [of earlier versions] and the

recovered file either replaces the current version or is copied to a new file.” EX1005, 130:10-13; *see also id.*, 23:29-24:12 (the system “present[s] the user with a list of prior states for a given, file and allow[s] any to be copied forward”). The presented earlier file versions are restorable because such file versions are stored in the history buffer and are used to recover versions of desired files. *Id.*, 31:19-24.

62. The restorable representations are retrievable from a remote storage location because Schneider teaches locating its historic data (from which the files are restored) on an external device, such as an external disk via a network. *Id.*, 108:15, 112:26; *see also id.*, 109:7-8 (describing “external disk drive that, like the main disk, employs a method of saving original disk states”). To provide redundant backup via an external disk, Schneider discloses a process whereby “[w]hen an external disk that has been set up is connected to the computer, the engine seeks to synchronize it with the internal disk.” *Id.*, 111:11-12; *see also id.*, 111:3-112:24 (describing general process for transferring/storing historic data on the external disk), 112:26-114:7 (describing how to implement the external disk as “a disk interfaced to a target computer through a network”). Schneider teaches use of an external disk to ensure “a guaranteed usable backup image is available” even if the local drive crashes or otherwise becomes unavailable. EX1005, 109:19-29. Schneider teaches one implementation in which the external (remote) disk maintains the same historic data as the internal (local) disk. *Id.*, 113:15-17; FIGS. 47A-F, 47H-

I.⁴ Schneider also teaches, alternatively, implementing the external disk with more “additional storage used for historic states,” allowing the external disk “to reach further back in time in re-creating prior states.” *Id.*, 113:17-21; *see also* FIG. 47G.

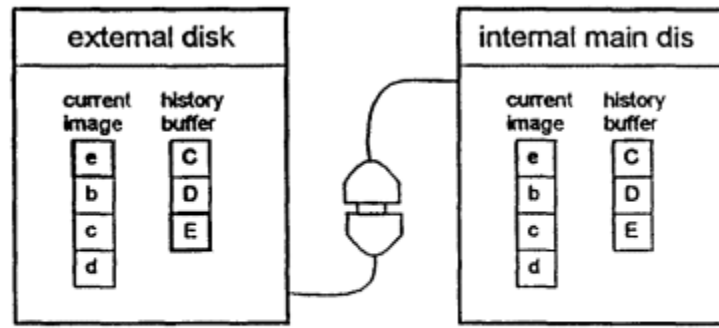


Figure 47I

EX1005, FIG. 47I

⁴ A POSITA would have understood that Schneider’s teachings regarding an external disk were equally applicable to Schneider’s file level implementation as to disk level implementation. Schneider discloses that its “present invention can be implemented to work either with managing data at the disk sector level or the file level.” EX1005, 10:25-27; *see also id.*, 28:2-6. A POSITA would implement Schneider’s use of an external disk in the file level implementation in order to provide hardware redundancy so as to be able to recover historic file versions even in the case of a lost data on the internal drive, e.g., through a system crash. *Id.*, 132:20-22.

63. Each prior version of a file stored on the external disk is retrievable because Scheider teaches that the external disk's historic information is used to "re-creat[e] prior states." *Id.*, 113:19-21, 24:21-24 (describing saving a prior file state "just before a file is opened and the intention to modify is clear"), 112:29-113:2 (describing transfer "from" the external disk). Schneider explains how the user "PC's software would divert to and take advantage of an external disk that had more historic information than available on the internal disk, and whose access is desired." *Id.*, 113:26-28; *see also id.*, 114:5-6 ("Note the figure shows data flowing from the PCs to the server, but data actually flows in both directions."). Therefore an external disk can allow for storage of more historic information including additional prior versions than could be backed up on the internal drive alone.

64. Schneider's network-connected external disk is a remote⁵ storage location at least because it is not internal to the user's PC, as shown in FIG. 47G, which depicts the historic data as backed up by a "server with engine software" that is connected to the user PCs via a LAN. While the figure labels its server as "local disk storage," a POSITA would understand this notation to mean that the disk storage is local to the server, not necessarily the user PCs.

⁵ The '850 patent never uses the term "remote" outside the claims.

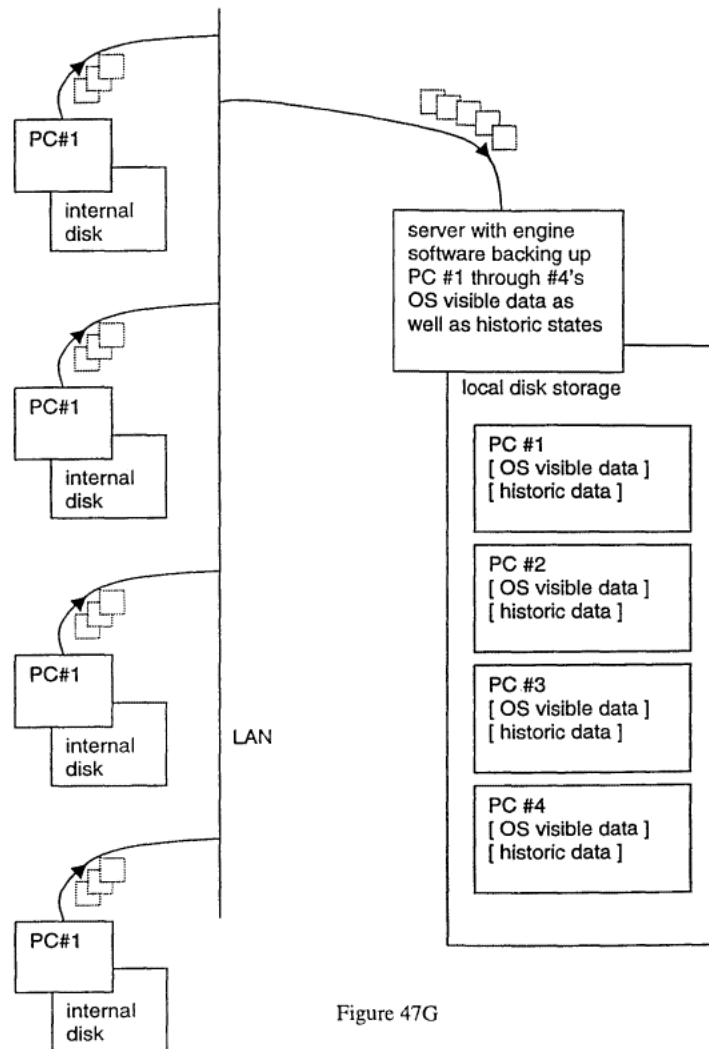


Figure 47G

EX1005, FIG. 47G

65. To the extent Patent Owner argues that Schneider’s network-connected external disk is not a “remote storage location,” e.g., because it is depicted as connected via a Local Area Network (“LAN”), it would have been obvious to connect to that external disk using a wide-area network such as the Internet instead of a LAN. As early as 1993, it was known that the Internet was “commonly used to transfer ... files between computers.” EX1009, 3:13-19 (background provided in

patent issued from application filed November 4, 1993). Even then, the Internet was a known alternative to a LAN: “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.” EX1009, 3:37-39; *see also* EX1008, 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”). It was also known to access external drives using the Internet, as “[d]rives on a remote system defined using NFS and RFS [we]re available for mounting by other remote systems on the Internet.” EX1009, 3:64-66; EX1008, 6:9-36; *see also supra* ¶¶34-35. By 2000, people had been “sharing data on the Internet for decades already” (EX1013, 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 alternative to LAN-connected drives.

66. Given that it was a known alternative to LAN-accessible external disks, it would have been an obvious design choice to use a remote Internet-accessible disk when implementing Schneider’s networked external disk. Indeed, the POSITA would have been motivated to do so, e.g., to reduce the chances of the backup information being destroyed. EX1013, 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.”). Moreover, an Internet-accessible drive would conserve local storage space and power and allow computers in different remote locations to access a single external storage disk. A POSITA would easily implement this known alternative, as Schneider already explains how a network server “can communicate with more than one PC” and “can independently maintain the OS visible disk image and historic states for a set of PC’s.”). EX1005, 113:3-5; *see also id.*, 11:18-28 (describing different types of “disk” and explaining how “[t]he processes described herein” apply “regardless of” which type of disk is used). The obviousness of this alternative is confirmed by the ’850 patent itself, which states that the storage device of the alleged invention “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, 5:35-40. It treats them as interchangeable, because they were. And during prosecution, applicants did not dispute the Examiner’s finding that it would have been obvious to receive prior revisions of a file from an Internet storage area network. EX1002, 144. Accordingly, element [1.2.ii/10.2.ii] is obvious.

d) [1.2.iii]/[10.2.iii]

[1.2.iii]/[10.2.iii] the restorable representation having at least information required for recovering the version V [*by a computational machine with the current version being accessible*], the remote storage location being accessible through a network;

67. As discussed for claim element [1.2.ii/10.2.ii] (¶¶61-63) Schneider discloses that its historic data, e.g., as stored in a history buffer, is used to recover prior versions of files to the user's computer. *See also* EX1005, 24:12-14. When a specific file is recovered from the historic data, "the recovered file either replaces the current version or is copied to a new file." *Id.*, 130:10-13. As further explained above for [1.2.ii/10.2.ii] (¶¶64-65), Schneider discloses accessing the remote storage location through a network and it would have been at least obvious for access to occur via the Internet to a physically or geographically remote location.

68. The user's computer also provides access to the current version of a given file. For example, Schneider discloses that the current image is accessed via the user's computer and the current image constitutes "the non-historic view of the disk," i.e., the "data last written by the user." *E.g., id.*, 35:26-36:3. Accordingly, a POSITA would understand that a current version of a given file is available and accessible to a user on the user's hard disk.

e) [1.3]/[10.3]

[1.3]/[10.3] (B) responsive to a selection to preview a selected previous version of the file based on the presented information for the collection of (A), presenting

a presentable representation of the selected previous version, the selected previous version being one of the previous versions of the file in the presented information for the collection, the presentable representation having at least information required for presenting at least a portion of the selected previous version [*by the computation machine with the current version being accessible*];
[and]

69. As discussed for [1.1/10.1] and [1.2.i/10.2.i], Schneider discloses displaying a list of previous file versions by the computational machine (i.e., the user's computer). In response to user selection of a particular version, the Schneider system previews the version, e.g., by copying the previous file to a temporary directory and allowing the user to view that temporary copy:

Note that it is useful to allow the user to select a file that can be retrieved from the past, and to automatically launch the appropriate application to view the file, referencing either the file on a simulated disk or copying the file from a simulated disk to a temporary directory. (The contents of this temporary directory, when no longer in use, are eventually automatically purged.) **This allows the user to not only know of the existence of an old version of a file, but to view its contents without actually formally retrieving the file,** as the viewed file is automatically discarded. Therefore the viewed file's existence in terms of being retrieved is hidden from the user in that the user does not have to manage the viewed file on disk.

EX1005, 60:21-29; *see also id.*, 60:3-5. A POSITA would understand that such a selection would be made using the list of previous file versions displayed to the user. For example, Schneider's Explorer extension identifies to a user a list of what old versions are available and the POSITA would implement the view selection mechanism by allowing the user to select a past file version from the list that the user wishes to view. Similarly, Schneider further teaches that its "invention" allows users to "simulate" historic versions "while at the same time allowing the user to continue using the current disk" and thus "copy forward information from the past into the current." EX1005, 11:10-13; *see also id.*, 22:28-23:2; 130:10-13 (the "recovered file" may be "copied to a new file" (allowing for preview) or replace the current version), 142:17-18, 160:27-28. At least by providing a temporary view of prior versions that can be viewed simultaneously with the current version of a file, the system taught by Schneider presents a presentable representation of the selected previous version, the selected previous version being one of the previous versions of the file in the information presented to the user.

70. At least because Schneider discloses users viewing and copying from the prior versions, the presentable representation (i.e., the previewed file) includes "information required for presenting at least a portion of the selected previous version by the computation machine." Schneider enables a user to select a stored prior version of a file to view. A POSITA would understand that this file includes

information required for presenting the selected previous file because Schneider “automatically launch[es] the appropriate application to view the file.” *Id.*, 60:21-23. As explained above for element [1.2.iii/10.2.iii] (*supra* ¶68) Schneider discloses the current version of a file being accessible, and such current version would remain accessible during the preview process disclosed by Schneider.

f) [1.4]/[10.4]

[1.4]/[10.4] (C) responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version [*by the computational machine*] as the current version on the local storage location, the selected previous version available from the restorable representation of the selected previous version.

71. Schneider teaches that, from a “list of earlier versions of a file” displayed to the user, a “selection can be made from the list and **the recovered file ... replaces the current version.**” *Id.*, 130:10-13. A POSITA would have understood this to involve retrieving the restorable representations from the remote storage location and storing it locally because Schneider discloses restoring past files by replacing the current file with the recovered file. EX1005, 130:11-13, 15:20-16:21, Fig. 2. Schneider discloses selecting a previous version of a file in order to restore it and store it as the current file version, e.g. by allowing a user to select the previous version from a displayed list of available stored previous versions. EX1005, 130:10-13. A POSITA would have understood that recovering a previous file version

from a remote storage location would include processes for searching, locating, and returning that file version in order to transfer the file to the user's computer. As explained above for element [1.1/10.1], Schneider discloses storing the current version by the local computer on the computer's local drive, i.e., a local storage location.

2. Claims 2 & 11

[2/11]. The method of claim [1/10], wherein, for the presenting of (B), the presentable representation is presented using an application that created the file or the selected previous version.

72. As I demonstrated above at paragraph 43 and in Exhibit 1023, dependent claims 2, 4, 5, 6, 7, 8, and 9 (depending from claim 1) are substantively identical to dependent claims 11, 12, 13, 14, 15, 16, and 17 (depending from claim 10). Accordingly, this declaration presents and analyzes each such corresponding pair of dependent claims together.

73. Schneider teaches this limitation. Schneider teaches that when a previous file version is previewed, Schneider "automatically launch[es] the appropriate application to view the file." *Id.*, 60:21-23. A POSITA knew that the most appropriate application to view a specific file was the application that created the file, making that application at least an obvious design choice. A POSITA understood that computer files are typically viewable by the same software application that created the file and such an application would be certain to work for

viewing the files it created. As just one example, in 2000 Microsoft Word was the most widely used word progressing program. Microsoft Word saved files created using the software using a file format specific to that application, using the “.doc” extensions, and documents stored using that extension are viewable using Microsoft Word. It would be evident to a POSITA that when a user created a specific previous version of a file using a software application on her computer, with the application creating the file according to the application’s specific file format, that same software application on the user’s computer would be used to view that file.

3. Claim 3

3. The method of claim 2, wherein the application is in a read-only mode.

74. Schneider teaches this limitation. Schneider teaches that when a user selects a listed previous version of a file to preview, the file may be viewed using an appropriate application by copying the file to a temporary directory. *Id.*, 60:21-29. The contents of this temporary directory, “when no longer in use, are eventually automatically purged.” *Id.*, 60:24-25. Thereby, the “viewed file’s existence in terms of being retrieved is hidden from the user in that the user does not have to manage the viewed file on disk.” *Id.*, 60:28-29. Accordingly, a POSITA would understand Schneider to, when previewing a file with an appropriate application, operate the application in read-only mode because the previewed file is stored in a temporary directory with its existence hidden from the user and is subsequently “automatically

discarded” (*id.*, 60:27-28). Accordingly, a POSITA would not desire to perform any write operation on such a file that is meant to be discarded and would implement the teachings of Schneider so as to have the viewing application open the file as read-only; that is, a POSITA would restrain the user from updating files that are being previewed, knowing that any such changes would be lost after viewing unless other action were taken.

75. It would have also at least been an obvious design choice to implement Schneider such that the preview of a given previous version was opened in read-only mode. A POSITA knew that applications had three access permission modes for opening or viewing files: “read-only” (i.e., enabling only viewing of the file), “read/write” (i.e., enabling viewing and editing the file) and read, write, and delete (i.e., enabling viewing, editing, and file deletion). EX1014, 398, 440. The most obvious design choice for implementing Schneider’s temporary preview feature was for the application to open the file in read-only mode. Further, the POSITA would have been additionally motivated to open the file in read-only mode so as to avoid inadvertent overwriting of the contents of the previous file version. Read-only mode would thus maintain the underlying purpose of Schneider’s file recovery system to retain previous versions for later viewing. Schneider specifically presents a list of previous versions of files to a user and allows a user to select a previous version and to select a previous version to retrieve and restore. A POSITA would understand that

if the viewing application is operated in a mode other than read-only (e.g., read-write mode), this would allow the possibility that the application overwrites data in the previous file version. The POSITA would understand that allowing writes to a previous file version would defeat the purpose of Schneider's file recovery system, because writing new data to a previous file version would turn that file into a new file with new data, and would thereby prevent a user from accessing and restoring the data of the previous file version. A POSITA would therefore implement Schneider to preview previous file versions in read-only mode in order to ensure the data integrity of stored previous versions of files. The POSITA had a reasonable expectation of success with such an implementation because read-only is one of the three fundamental permission modes for accessing a file and all three such modes have at least read capability.

4. Claims 4 & 12

[4/12]. The method of claim [1/10], wherein the information includes at least two previous versions of the file.
--

76. Schneider discloses this limitation. Schneider presents multiple files in “a list of earlier versions of a file.” *Id.*, 130:10-13. *See also* 31:21-24 (“In essence, from the user's point of view, he or she could select a file and request the present invention's software to locate all available versions of that file, over time, that are still in the history buffer.”). A POSITA would have understood that Schneider's

plural “versions” constitutes at least two versions because that would have been a common usage scenario that is expressly contemplated by Schneider.

5. Claims 5 & 13

[5/13]. The method of claim [1/10], wherein the information for the collection includes information indicative of at least one or more of a timestamp, an original file location, and a comment of the previous versions or the file.

77. Schneider discloses this limitation. As explained, Schneider discloses storing previous file versions for an archived file in a history buffer. EX1005, 24:12-14, 31:12-24. Schneider further discloses that its history buffer is “organized in[to] blocks” with each block having a header containing a table of mapping entries. *Id.*, 26:19-24. Each table entry contains: “**1) a type, 2) a time stamp, and 3) an original disk location.**” *Id.*, 26:25-27. And Schneider teaches implementing this organization scheme “at the file level, where the data pages would correspond to file information [and o]ther details would also change as are obvious.” *Id.*, 28:3-6. As explained above, Schneider also teaches “show[ing] all available old versions of files.” *Id.*, 59:27-60:20. A POSITA would have been motivated to include all of this available file information when presenting the previous versions, so that the user could make a more informed decision in selecting prior versions for viewing. Accordingly, a POSITA would have implemented Schneider to present information that included a timestamp, an original file location, and a comment indicating a file type, for each archived previous version of a file.

6. Claims 6 & 14**a) [6.1/14.1]**

[6.1/14.1] The method of claim [1/10], further comprising the steps of: (B2) after the presenting of (B) and responsive to a selection to not restoring the previous version and responsive to a second selection of a second previous version of the previous versions, version presenting a second presentable representation of the selected second previous version, the second presentable representation having at least information required for presenting at least a portion of the selected previous version; and

78. It would be common sense to a POSITA that a user would repeat the steps of claim [1.2.i-iii/10.2.i-iii] to iterate between previewing and rejecting multiple previous file versions until the user finds the correct file version the user wishes to restore.

79. As explained, Schneider presents to a user a list of previous file versions and allows a user to select a file to preview and subsequently select to retrieve and restore that file to save it as the current version. *See, e.g.*, EX1005, 60:3-5. A POSITA would understand that a user would implement Schneider's file retrieval system iteratively. That is, a POSITA would have understood the purpose of Schneider's file preview functionality of allowing a user to view the contents of a previously stored version, without having to actually retrieve and restore that file version. EX1005, 60:25-27 ("This allows the user to not only know of the existence of an old version of a file, but to view its contents without actually formally retrieving the file."). Indeed, the capability of previewing a file before restoring it

indicates an expectation that the user will not necessarily want to restore the initially selected previous version. A user searching for a specific previous file version may not know which listed previous version is the one she is looking for without first previewing the content of the file. Thus, a user might go through multiple iterations of selecting a first previous file version to preview, determining that the previewed version is not the correct version, selecting to not restore that version (e.g., by closing the preview display or selecting another version in the presented list of file versions), and selecting to preview the second version, until the user finds the correct file version that she is searching for.

[6.2/14.2] (C2) responsive to a selection to restore the second selected previous version, retrieving the restorable representation of the second selected previous version from the remote storage location and storing the second selected previous version as the current version on the local storage location, the second selected previous version available from the restorable representation of the second selected previous version.

80. Schneider teaches this limitation. As explained above for claim [6.1/14.1], it would have been obvious to a POSITA that a user using the file recovery system of Schneider would iterate through previewing and rejecting multiple versions of stored previous file versions until the user finds the correct previous file version the user is searching for. As explained above for element [1.3/10.3] (*supra* ¶¶69-70), Schneider discloses a process of responsive to a selection to restore a selected previous version, retrieving the restorable representation of the

selected version from the remote storage location and storing that version as the current version on the local storage location, with the selected previous version available from the restorable representation of that version. A POSITA understood that a user would employ that same procedure to select, retrieve, and restore a second selected previous version (i.e., where the user determines that the second selected previous version is the version the user is searching for).

7. Claims 7 & 15

[7/15]. The method of claim [1/10], wherein the restorable representation of the selected previous version comprises one or more of a compressed and encrypted version of the selected previous version.

81. Schneider discloses this limitation. Schneider states that it “is anticipated that data compression techniques could be applied to the history buffer to reduce the space it requires.” EX1005, 26:2-4. Accordingly, a POSITA would implement data compression to the previous file versions stored in the history buffer to reduce the space required to store such data.

8. Claims 8 & 16

[8/16]. The method of claim [1/10], wherein the restorable representation of the selected previous version comprises an incremental change to the file.

82. Schneider discloses this limitation. As explained, Schneider discloses performing automatic backup in response to an operation such as a write operation. EX1005, 12:1-2, 12:5-8, 24:7-10, 24:12-14, 24:21-23, 36:12-17. Therefore, a

POSITA would understand that each version in the list of previous file versions presented by Schneider would contain incremental changes introduced by such a write operation to the relevant file.

9. Claims 9 & 17

[9/17]. The method of claim [1/10], wherein the method has a substantially imperceptible impact on system performance from a user's point of view by delaying one or more operations to prevent a perceptible impact on system performance from a user's point of view.

83. Schneider discloses this limitation.⁶ The '850 patent indicates that this step can be performed during a data backup process and is not limited to the stage of retrieving stored data. The patent discloses a method of automatically storing a backup version of a file at the time the file is opened by a user in response to an instruction to open the file. "In step 210, file capture block 10 detects the instruction and captures the operating file.... In step 215 the resident program opens the

⁶ I understand that in the Parallel Litigation, Petitioner Dropbox contends that the term "a "substantially imperceptible impact ... from a user's point of view" is indefinite for failing to inform a POSITA with reasonable certainty about the scope of the claimed invention. EX1017, 9-11; *see also* EX1021, 3. I provide no opinion regarding potential indefiniteness of the term because a POSITA would recognize the disclosures of Schneider to be within the bounds of the term as explained for this limitation regardless of any uncertainty as to those bounds.

operating file.... Step 210 is performed by momentarily delaying the execution of step 215 in such a manner as to have little or no perceptible impact on system performance from the users [sic] point of view.” EX1001, 7:9-28.

84. Similarly, Schneider discloses that when files are written to archive, “[i]n order to least impact the user’s use of a computer, a goal is implementing the present invention to minimize the performance impact on reading data and rely on writes being a background activity. If writes are done in the background then overhead can be added without impacting response to the user.” EX1005, 32:9-12. Accordingly, Schneider teaches delaying a write operation such that its data backup operations have a substantially imperceptible impact on system performance from user’s point of view. *See also id.*, 37:22-29 (“Caching writes in memory and committing them to disk during free time can reduce or eliminate the impact on the user, even though there is a tripling of time in the actual writing of the data.... [I]n theory the user had been free to continue working and so should not notice the performance degradation.”).

10. Claim 18

a) [18.1]

[18.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:

85. The preamble to claim 18 is identical to that of claim 1 and Schneider therefore discloses this limitation for the same reasons as explained for element [1.1]. *See* ¶¶57-58.

b) [18.2]

[18.2] (A) iteratively transmitting a representation of at least a portion of a current version of a file to a remote storage location for archiving the current version responsive to a determination of an availability of the remote storage location for accepting a transmission of the representation, the current version being accessible at a computational machine, the remote storage location being accessible through a network;

86. Schneider discloses this limitation. As explained, Schneider discloses performing automatic backup in response to an operation such as a write operation. EX1005, 12:1-2, 12:5-8, 24:7-10, 24:12-14, 24:21-23, 36:12-17. Accordingly, a POSITA would understand that Schneider iteratively transmits such backed-up file versions to Schneider's remote storage location. Schneider teaches to a POSITA a determination of whether a remote storage location is available in that it discloses that the external storage disk can be removable and that transfer of information does not begin until the external storage is reattached. EX1005, 110:24-28. It would further be obvious to a POSITA to make a determination of the availability of the remote storage location prior to transmitting file back-up versions. A POSITA would understand that if the remote storage location is unavailable, for example if the remote storage is faulty, without power, or disconnected from the network,

transmission of file back-ups would be unsuccessful and the user should be notified of the problem so that saved files may be backed up using a separate method or and so the user can fix the problem with the remote backup location.

87. Schneider teaches a current file version being accessible at a computational machine (i.e., a local computer) and a remote storage location being accessible through a network for the same reasons explained for elements [1.2.i-ii/10.2.i-ii] (*supra* ¶¶59-66).

c) [18.3.i], [18.3.ii], [18.4], [18.5]

88. As I demonstrated at paragraph 44 and in Exhibit 1023, claim elements [18.3.i], [18.3.ii] [18.4], and [18.5] are textually identical to respective claim 1 elements [1.2.i], [1.2.ii], [1.3], and [1.4] except for internal headers (i.e., A, B, C, D). Accordingly, Schneider discloses each of elements [18.3.i], [18.3.ii], [18.4], and [18.5] for the same reasons explained for respective elements [1.2.i], [1.2.ii], [1.3], and [1.4].

d) [18.3.iii]

[18.3.iii] the restorable representation having at least information required for recovering the version V by the computational machine, the remote storage location being accessible through a network;

89. Schneider discloses this limitation for the same reasons explained for limitation [1.2.iii] (*supra* ¶¶67-68). As demonstrated at paragraph 44, limitations

[18.3.iii] and [1.2.iii] are textually identical except that [1.A.2.iii] is narrower in requiring the additional language of “with the current version being accessible.”

11. Claim 19

a) [19.1]

[19.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:

90. The preamble to claim 19 is identical to that of claim 1 and Schneider therefore discloses this limitation for the same reasons as explained for element [1.1].

b) [19.2]

[19.2] (a) receiving a request to display a list of captured revisions for a file; (b) displaying a list of captured revisions for said file;

91. Schneider discloses this limitation for the same reasons explained for element [1.2.i/10.2.i] (*supra* ¶¶59-60). Schneider discloses allowing the user to “right click on a specific file and have the option to view a list of old versions of the file.” EX1005, 59:27-60:5; *see also id.*, 58:20-23, 58:24-27, 130:10-17, 31:19-24.

c) [19.3]

[19.3] (c) receiving a selection from said list indicating a revision of said file to restore; (d) previewing selected revision of said file, wherein said selected revision of said file is received from an Internet storage area network;

92. Schneider teaches this limitation for the same reasons explained for element [1.3/10.3]. The '850 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, 4:45-48. In response to user selection of a particular version, the Schneider system previews the version by allowing the user to view a temporary copy. EX1005, 60:21-29; *see also id.*, 60:3-5.

d) [19.4]

[19.4] (e) receiving another selection from said list indicating another revision of said file to restore; (f) previewing selected another revision of said file wherein said selected another revision of said file is received from a network attached storage location; and

93. Schneider teaches this limitation for the same reasons as explained for element [6.1/14.1]. That is, a POSITA would understand that a user attempting to restore a particular file version would utilize Schneider's file preview functionality iteratively to select and preview successive previous versions of a file until the user finds the correct version the user was searching for.

a) [19.5]

[19.5] (g) restoring selected another revision of said file following said previewing step in (f), optionally, decrypting said selected revision of said file prior to step (d) and decrypting said selected another revision of said file prior to step (f).

94. Schneider teaches this limitation for the same reasons as explained for element [1.4/10.4] and [6.2/14.2]. A user seeking to restore a particular file version, after finding the correct version through iteratively previewing multiple versions would then select to restore the file. Schneider discloses that after displaying its list of prior file versions “a selection can be made from the list and **the recovered file ... replaces the current version.**” *Id.*, 130:10-13.

95. A POSITA would understand the entirety of the limitation following the word “optionally” to be modified by the word and therefore be an optional element, because the remainder of the limitation following “optional” is a single clause undivided by any punctuation.

12. Claims 20 & 21

96. As I demonstrated in paragraph 45, claims 20 and 21 are substantially the same, with no meaningful differences in their claim language. Differences between the claims are shown below in brackets. Accordingly, my declaration analyzes the claims together.

[20/21]. The method of claim 19, [wherein said method comprises/further comprising the step of] decompressing said selected revision of said file prior to [step] (d) and decompressing said selected another revision of said file prior to [step] (f).

97. Schneider discloses this limitation. As explained above for claims 7/15 (*supra* ¶81), Schneider anticipates performing data compression for files *stored* in

its history buffer. A POSITA would understand that compression of archived data would naturally also require decompressing an archived file version in order to display the file.

**IX. GROUND 2: CLAIMS 1-21 ARE OBVIOUS
OVER SCHNEIDER IN VIEW OF CRAWFORD**

98. To the extent Patent Owner argues that Schneider does not disclose that its external storage constitutes a remote storage location accessible via a network (independent claims 1, 10, 18) or an internet storage area network and network attached storage location (independent claim 19), such remote network and internet storage would have been obvious over Crawford (EX1009). Crawford published on June 23, 1998, and is thus prior art under at least Sections 102(a) and 102(b). Crawford is analogous art to the '850 Patent because it is from the same field of endeavor. Crawford discloses an "Internet online backup system provid[ing] remote storage." EX1009, Title. Crawford thereby allows computer data recovery, when recovery is necessary. *Id.*, 8:13-22. Thus, the '850 Patent and Crawford are both within the same field of endeavor of computer file storage, management, and recovery.

99. The invention of Crawford allows a local customer computer to connect to a remote server over the internet "and obtain additional processing and storage resources for the customer's computer. The resources can take the form of virtual storage and processing capabilities." *Id.*, Abstract.

100. Crawford discloses a preferred embodiment as depicted in Figure 1, reproduced below. In this embodiment, a computer 50 is connected to an on-line service system 100 over a data link 150. *Id.*, 12:40-45, 14:18-21, Fig. 1. This data link “may comprise a dial up telephone line or other similarly convenient telecommunications link that allows customer computer 50 to be located remotely to the on-line service system 100.” *Id.*, 14:22-25. Both the computer 50 and on-line service system 100 include a modem to facilitate the data link. *Id.*, 16:3-9.

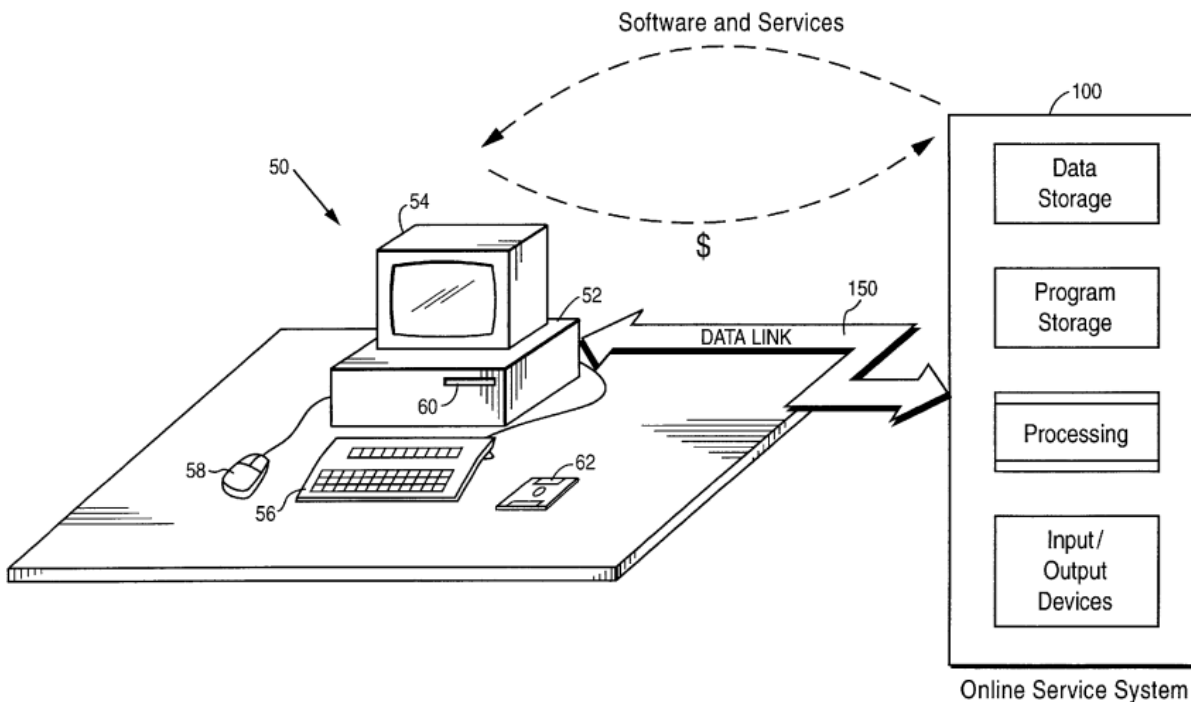


Fig. 1

101. Via this data link, the system of Crawford provides offsite archival services whereby files on the local customer computer “are automatically copied to on-line service virtual disks for offsite archiving.” *Id.*, 14:45-48. When needed, this on-line virtual disk archive is used to restore files on the local customer computer.

Id., 14:50-53 (“Virtual disks inactive for a specified period can be copied to on-line service tape for long term offsite archival.”). Crawford therefore discloses computer file backup and restoration with a remote storage location over an Internet network connection. *See id.*, Title (“Internet Online Backup System Provides Remote Storage”), 61:20-24 (claim 12) (“allowing the customer to restore data from the online backup service provider storage over the Internet connection”).

102. A POSITA would have been motivated to modify the file back-up and restore system of Schneider to include an Internet data link to a remote storage device as taught by Crawford. A POSITA would have done so to achieve the benefits explained for claim [1.2.ii/10.2.ii] for Ground 1 above (*supra* ¶¶65-66). Crawford explains additional benefits for its Internet based storage system that a POSITA would want to achieve. Crawford explains that a user’s local computer may run out of space, thereby limiting the amount of data that can be stored. EX1009, 13:28-31. Crawford solves this problem by connecting the local computer to additional processing and memory resources over the internet, thereby providing a user with “additional disk space, or access to a more powerful processor.” *Id.*, Abstract. A POSITA would have therefore recognized that the teachings Crawford could improve the computing system of Schneider by providing it with additional computing resources including additional memory storage and processing power. A POSITA would have had a reasonable expectation of success of implementing

remote storage over an Internet link as taught by Crawford, because Crawford implements such connecting using a conventional modem (*id.*, 16:3-7) and teaches that communications over the modem are performed using “conventional data packet transmission techniques” (*id.*, 57:44-48).

103. Further, to the extent Patent Owner argues that the term “optionally, decrypting said selected revision of said file prior to step (d) and decrypting said selected another revision of said file prior to step (f)” (claim [19.8]) is not merely an optional limitation, such limitation is also rendered obvious via the proposed combination of Schneider and Crawford. Crawford discloses encrypting data transmitted by the local customer computer to the remote online backup system. EX1009, 60:42-46 (claim 1) (“(1) encrypting backup data provided by the customer’s computer, (2) transmitting the encrypted backup data to the online backup service provider”). Because the system of Crawford encrypts data to send from the customer computer to remote backup, a POSITA would understand that such encrypted data would be decrypted at the user’s computer following transmission in the other direction from the online backup to the local computer. A POSITA combining the file backup system of Schneider with the remote Internet data link between local and remote storage as taught by Crawford would also be motivated to incorporate Crawford’s teachings of encrypting and decrypting data transmitted between the local computer and remote storage in order to give a user

the option to encrypt files backed up to remote storage and decrypt such files when they are retrieved in order to ensure security of data in transmission and while it is being stored.

**X. GROUND 3: CLAIMS 1-8, 10-16,
AND 18-21 ARE OBVIOUS OVER SWEAT**

104. As explained below, claims 1-8, 10-16, and 18-21 are obvious over Sweat.

A. Sweat

105. Sweat (EX1006) is a U.S. patent issued on June 13, 2006, and issued from U.S. Patent Application No. 09/534,757 (the “’757 application”) filed on March 24, 2000, and thus qualifies as prior art under at least Sections 102(e).

106. Sweat claims priority to provisional application No. 60/126,170 (EX1007, the “Sweat provisional”), filed on March 25, 1999.

107. Sweat is analogous art to the ’850 Patent because it is from the same field of endeavor. Sweat discloses “a project hosting service” allowing users to view and download prior versions of files. EX1006, Abstract. Sweat allows a user to recover prior file versions from a server to the user’s local computer. *Id.*, 10:51-55. Thus, the ’850 Patent and Sweat are both within the same field of endeavor of computer file storage, management, and recovery.

108. Sweat relates to “a method, apparatus, and article of manufacture for a project hosting service that a user can access with a web browser. With it, a user can

communicate and collaborate with the members of a design team or construction project. Embodiments of the invention let a user share CAD drawing files (DWG, DWF, etc.), document files, or any other type of file.” EX1006, 1:43-49. The hardware and software environment of Sweat is depicted in Figure 1, reproduced below, which shows a user’s client computer 100 connected to a server computer 106 via a network 102. *Id.*, Fig. 1, 3:2-6. The server 106 is connected, among other things, to a database 116 and a file server 120 that accesses file storage 124. *Id.*, 3:18-26.

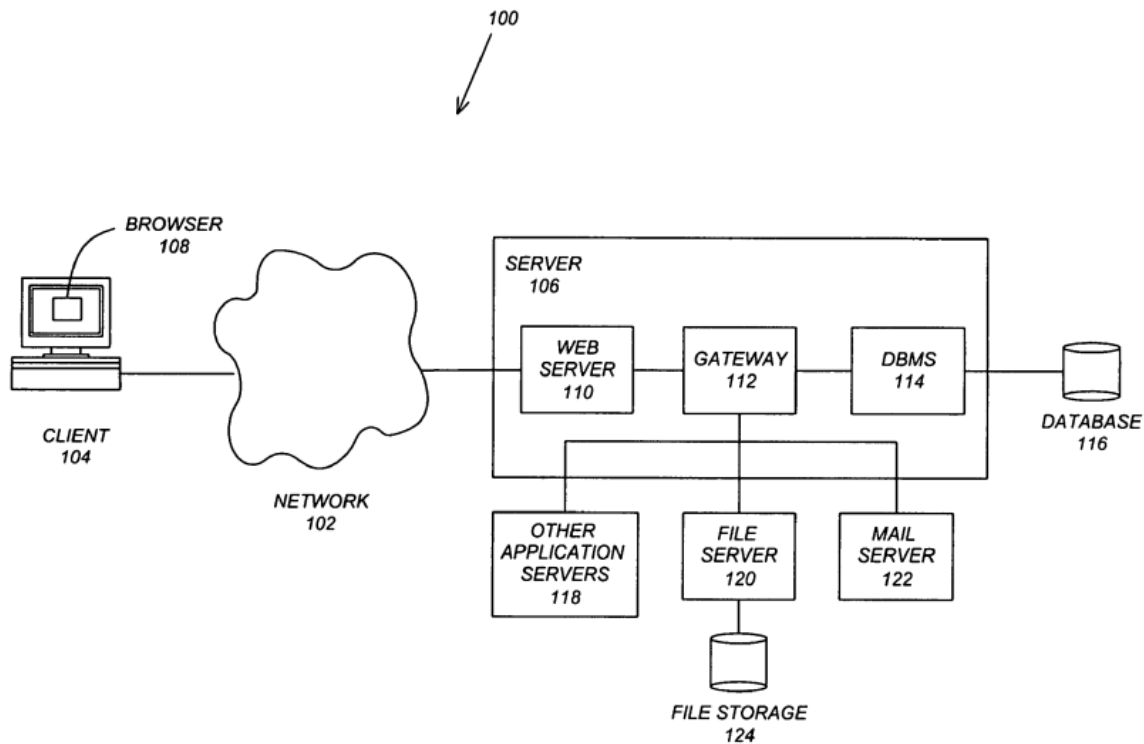


FIG. 1

109. Sweat’s distributed file management system manages projects that can be accessed by multiple users connected via the network. A project is managed in

the form of a project folder, which “may be used to organize and store folders, project files, text files, other files, drawings, links, notes, etc.” *Id.*, 16:40-44; *see also id.*, 17:5-8 (“A user can create and use standard folders and sub-folders to organize and store project information (text files, drawings, links, and notes) to the best advantage for the user's work environment.”). Sweat implements incremental backup of files from the client to the server as users make changes to files. “When a member saves changes or adds new information to files in the client 104... the new data is uploaded to the server 106.” *Id.*, 17:16-19. Sweat allows multiple versions of a file to be saved, including allowing for an unlimited number of prior versions. *Id.*, 9:19-25, Fig. 4. As Sweat explains, “[f]ile versions allow the opportunity to create multiple versions of a single document wherein prior versions are available for viewing. When a user creates a new file version, an updated version of a file that already exists on the server 106 is updated.... The updated version of the file becomes the new version (e.g., version 2), and the older version remains the same (e.g., version 1).” *Id.*, 19:42-52.

110. Sweat provides an interface to a user for viewing, previewing, and downloading previous versions of a file. Figure 5, reproduced below, shows an interface window displayed to a user when the user selects a specific file using Sweat's file management system on the user's client computer. *Id.*, Fig. 5, 2:16-18.

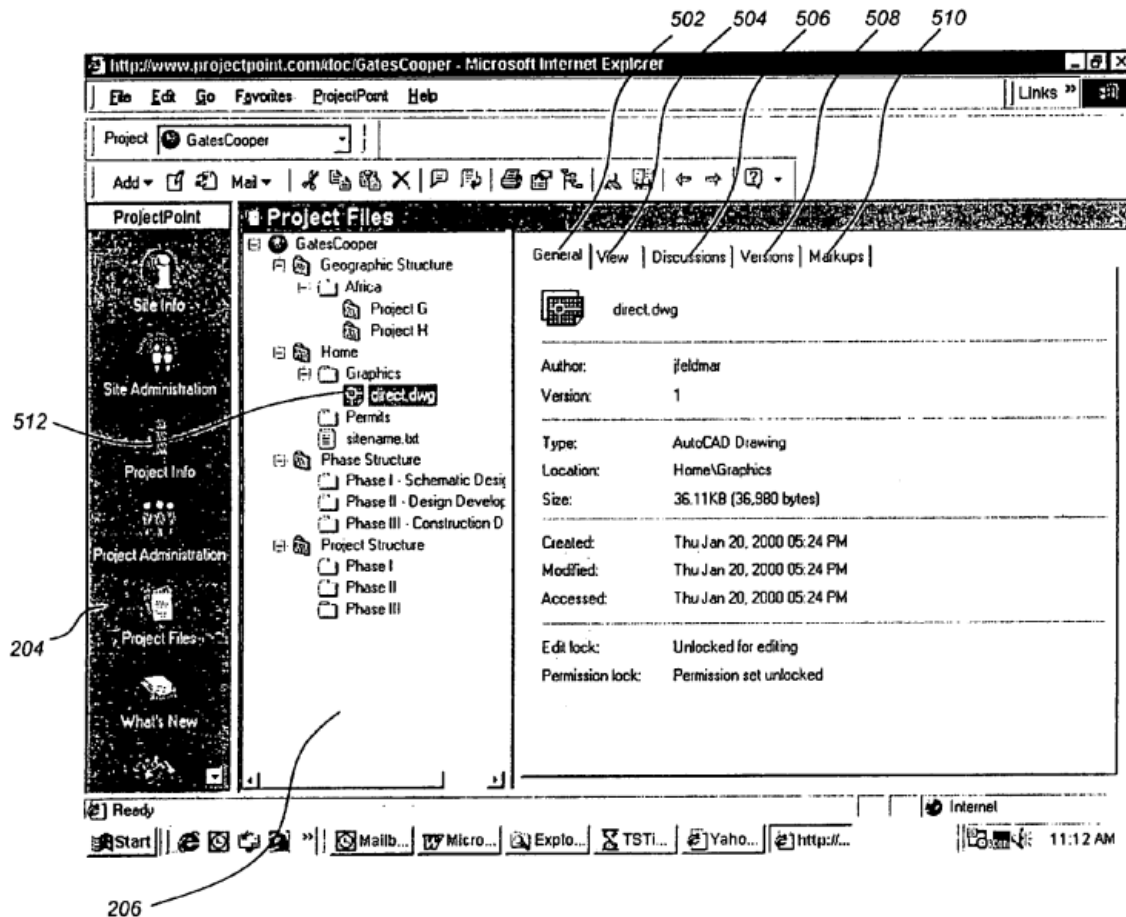


FIG. 5

111. A user can select the “versions tab 508,” which “provides a list of all of the currently selected files” and “is used to open and view any file version [and] download a selected file to a local computer 104.” *Id.*, 10:51-55.

112. The Sweat provisional is incorporated in its entirety in Sweat. EX1006, 1:7-12. The Sweat provisional supports at least claim 1 of Sweat, as shown in the table included in Appendix B.

113. The Sweat provisional supports the disclosures in Sweat’s specification, as shown in the table included in Appendix C.

1. Claims 1 & 10

114. As demonstrated above at paragraph 42 and in Exhibit 1023, the limitations of claims 1 and 10 are identical except that Claim 1 includes certain additional limitations not included in claim 10. Accordingly, my declaration below presents and analyzes the limitations of these claims together and identifies the additional limitations contained in claim 1 and not claim 10 [*bracketed and in italics*].

a) [1.1] / [10.1]

[1.1]/[10.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:

115. To the extent the preamble is limiting, Sweat teaches this limitation.

116. Sweat teaches having a current version of a file being available at a local storage location. For example, Sweat discloses that its versions tab 508 “is used to open and **view any file version [and] download a selected file to a local computer.**” EX1006, 10:51-55; EX1007, 9, 31⁷. Any file version would also include the most recent, i.e., the current version of the file. Sweat enables the user to download any file version, including this current version from the server to their

⁷ Unless otherwise noted, Sweat Provisional cites are to the page numbers added to the corresponding exhibit filed herewith.

local computer by clicking a “Download” button. EX1006, 10:56-61; EX1007, 9, 31. Accordingly, a POSITA would recognize a user would have a current version of a file stored to their local client computer, for example to work on and edit the file. Further, Sweat discloses that the user can use an “Update” command to designate a current file version at the user’s local computer as the newest file version at the server. Sweat discloses that “[i]f the user elects to update a file (e.g., by selecting the ‘Update’ option from the right-click menu), the user is prompted to select the file on the local computer 104 that corresponds to the file 512 to be updated on server 106.” EX1006, 11:5-8; EX1007, 7, 63. This further indicates that a user would have the current version available at their local computer and the user can also upload that current version to the server if desired.

117. As explained with respect to the remainder of claim 1 below, Sweat discloses restoring a file available in local storage to a previous version of the file.

b) [1.2.i]/[10.2.i]

[1.2.i]/[10.2.i] (A) presenting information for a collection of one or more previous versions of the file, the information for the collection including information indicative of at least one or more of previous versions of the file,

118. Sweat discloses this limitation. Sweat discloses that its file management system backs up previous file versions to a remote storage server. EX1006, 17:16-19 (“When a member saves changes or adds new information to files in the client 104... the new data is uploaded to the server 106.”), 5:56-58 (“Use [the Update]

command to select and upload a new version of a file to the server.”); EX1007, 63, 104. The server can store any number of previous versions of a file. EX1006, 9:19-25, Fig. 4; EX1007, 21, 25, 30.

119. Sweat discloses presenting this collection of previous file versions to a user. Specifically, Sweat discloses a “a list displayed in versions tab” that “is used to open and view any file version, download a selected file to a local computer 104, and review or edit comments that are attached to the selected file version.” EX1006, 10:52-57; EX1007, 31, 9. Sweat also describes an alternative “view tab” to view the previous versions EX1006, 10:57-58, 10:59-11:2; EX1007, 21, 25, 30. Accordingly Sweat discloses presenting information for a collection of one or more previous versions of a file (i.e., by presenting a list of those collected files), with the information for the collection including information indicative of at least one or more previous versions (i.e., by presenting each previous version in the list).

c) [1.2.ii]/[10.2.ii]

[1.2.ii]/[10.2.ii] wherein a restorable representation of each version, V, of the previous versions, is retrievable from a remote storage location,

120. Sweat discloses this limitation. Sweat makes each previous version displayed in the list of versions tab 508 restorable from the remote storage location of server 106. As Sweat shows in Fig. 1, reproduced below, server 106 is accessible by a user’s local computer over network 102. EX1006, Fig. 1, 3:2-6; EX1007, 3, 53.

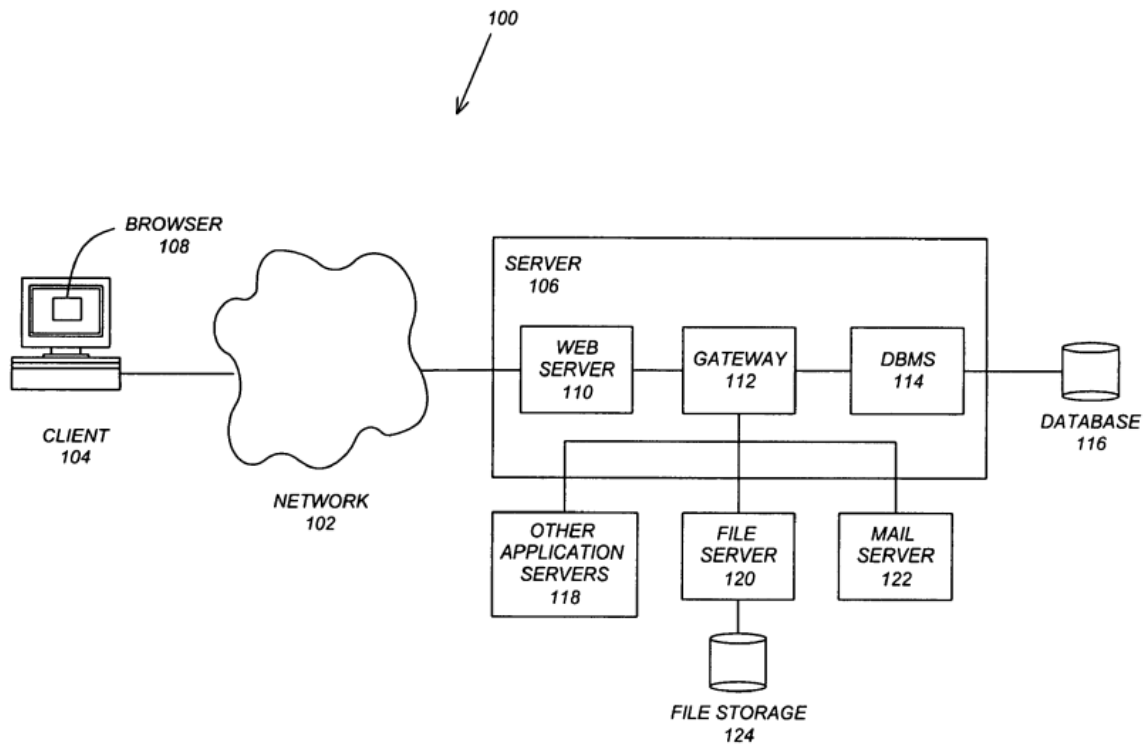


FIG. 1

Sweat specifically describes the network as “Internet 102.” EX1006, 3:12-14; EX1007, 3, 103. A POSITA would therefore recognize Sweat’s server 106 to be remotely connected to the local computer over the Internet, and server 106 (and storage connected thereto such as file storage 124) constitutes a remote storage location.

121. A user can restore a previous version by downloading a prior stored version from server 106 to the user’s local computer. “To download a version of the file from the server 106 to the local computer 104, the user selects the desired version and clicks a ‘Download’ button. Thereafter, the user identifies/selects the target location to save the selected file. The file is then copied to the location specified.”

EX1006, 10:58-63; EX1007, 31, 9. When a user clicks the “Download” button, the system “[c]opies the file or folder to the user’s local computer” and the user can “overwrite files that exist on the user’s local computer, as desired.” EX1006, 5:52-55; EX1007, 33, 79. A POSITA would therefore recognize that a user would store the retrieved file version as the current operative version of the file by using the “Download” command to overwrite the existing current version on the user’s computer with the desired previous version of the file.

d) [1.2.iii]/[10.2.iii]

[1.2.iii]/[10.2.iii] the restorable representation having at least information required for recovering the version V [*by a computational machine with the current version being accessible*], the remote storage location being accessible through a network;

122. Sweat discloses this limitation. As explained for claim [1.2.ii/10.2.ii], each listed previous file version in the versions tab 508 of Sweat includes information required for recovering that version to the user’s local computer because it allows a user to select the version to download it from the server 106 to the user’s local computer 104. The remote storage location of server 106 is accessible through network 102, which may comprise the Internet. EX1006, Fig. 1, 3:2-8; EX1007, 3, 53.

e) [1.3]/[10.3]

[1.3]/[10.3] (B) responsive to a selection to preview a selected previous version of the file based on the presented information for the collection of (A), presenting

a presentable representation of the selected previous version, the selected previous version being one of the previous versions of the file in the presented information for the collection, the presentable representation having at least information required for presenting at least a portion of the selected previous version [*by the computation machine with the current version being accessible*];
[and]

123. Sweat teaches this limitation. Sweat further discloses allowing a selection to preview a previous file version in the presented list of versions and then presenting a presentable representation of that version. As explained, Sweat presents a list of previous file versions to a user. For example, “[t]o open a version, the user selects the version from a list displayed in versions tab 508 and selects a ‘View in Window’ button.” EX1006, 10:55-58; EX1007, 31, 9. When a user selects the “View in Window” button, “the selected file appears in a separate window” such that the “user sees the contents of the file.” EX1006, 24:13-19; *see also id.*, 24:6-11; EX1007, 31, 9. A POSITA would understand that the presentable representation of each file (i.e., the file presented in the list of Sweat’s versions tab) include information required for presenting the selected previous version by the user’s computer, because the user selects the specific previous version from the list to view the content of that version.

124. It would be obvious to a POSITA that a user would carry out the teachings of Sweat so as to perform the file preview functionality of Sweat with the current version being accessible. As explained for claim [1.1] (*supra* ¶116) a user

would have a current version stored in their local computer and it would therefore be accessible. That is, Sweat discloses that a user would download the newest version of a file from the remote server to their local computer, or, alternatively, use the update command to set a file already stored to the computer as the most current version of the file. Such a user with the current version accessible on their computer would desire to restore a previous version of the file to replace the current version, for example, in the case where the newest version of the file has become corrupted or where desired data has been deleted from the newest version. EX1005, 4:26-29 (explaining the long standing practice of providing backup files to recover a file that has been corrupted or lost). A POSITA would recognize that such a user would utilize Sweat's previous file version preview functionality (e.g., through selecting the "View in Window" button in the versions tab 508) to determine whether a given previous version is the correct version the user is seeking prior to restoring the file in order to conserve time, computer memory, and processing resources.

f) [1.4]/[10.4]

[1.4]/[10.4] (C) responsive to a selection to restore the selected previous version, retrieving the restorable representation of the selected previous version from the remote storage location and storing the selected previous version [*by the computational machine*] as the current version on the local storage location, the selected previous version available from the restorable representation of the selected previous version.

125. Sweat teaches this limitation. Specifically, Sweat allows a user to restore a selected previous version of a file by presenting a selection to the user to download the previous version from the restorable representation of that version (i.e., the previous version displayed in the list of previous versions to the user via versions tab 508). Sweat discloses that “[t]o download a version of the file from the server 106 to the local computer 104, the user selects the desired version and clicks a ‘Download’ button. Thereafter, the user identifies/selects the target location to save the selected file. The file is then copied to the location specified.” EX1006, 10:58-63; EX1007, 31, 9. Sweat explains that when a user clicks the “Download” button, the system “[c]opies the file or folder to the user’s local computer” and that the user can “overwrite files that exist on the user’s local computer, as desired.” EX1006, 5:52-55; EX1007, 3, 103. A POSITA would therefore find it obvious that a user implementing the system of Sweat would use the “Download” button to restore a previous version of a file by copying the file to the user’s local computer and overwriting the current version of the file on the user’s local computer, so as to save the restored file as the new current version of that file on the local computer. A POSITA would have understood that recovering a previous file version from a remote storage location would include processes for searching, locating, and returning that file version in order to transfer the file to the user’s computer.

126. Alternatively, Sweat discloses a user selecting a file stored on the user's local computer and copying that file to the server as the newest version of the file. Specifically, Sweat discloses a user selecting an "Update" option to select a file on the local computer and store that file to the server as the newest version of the file. EX1006, 11:5-9, 19:46-56; EX1007, 63.

2. Claims 2 & 11

[2/11]. The method of claim [1/10], wherein, for the presenting of (B), the presentable representation is presented using an application that created the file or the selected previous version.

127. As I demonstrated at above at paragraph 43 and in Exhibit 1023, dependent claims 2, 4, 5, 6, 7, 8, and 9 (depending from claim 1) are substantively identical to dependent claims 11, 12, 13, 14, 15, 16, and 17 (depending from claim 10). Accordingly, this declaration presents and analyzes each such corresponding pair of dependent claims together.

128. Sweat teaches this limitation. A POSITA would understand that an appropriate application to view a specific file includes an application that created the file. A POSITA understood that computer files are typically viewable by the same software application that created the file. As Sweat explains, "[t]o view project files including drawing files and markup files, a user must install the appropriate viewer to open and view drawing files." EX1006, 23:38-41; *see also id.* 5:46-48 (opening files "using the application registered to that file type on your computer"), 1:63-67;

EX1007, 59, 73. Sweat relates specifically to the management of CAD drawing files and acknowledges that “[t]he use of Computer Aided Design (CAD) application programs is well known in the art.” EX1006, 1:24-26; *see also id.*, Abstract. Because Sweat discloses project members using CAD applications on their local computers to create and edit project files, it would be obvious to a POSITA that the same CAD application would be used to view previewed files of the same file type generated by that CAD application.

3. Claim 3

3. The method of claim 2, wherein the application is in a read-only mode.

129. Sweat teaches this limitation. As explained, Sweat discloses using an appropriate application to view a previous file version in order to provide a preview of that version to a user. Sweat discloses that members of a project using its system can be granted different levels of permission by a system administrator. EX1006, 14:43-52; EX1007, 35-36, 38. One such permission setting is “[v]iew permission,” which only “provides the ability to view project files without the ability to add, edit, or delete them.” EX1006, 14:60-61, 13:18-21; EX1007, 35-36, 38. Therefore, a POSITA would understand that where a user has been granted view permission by a system administrator and where the user selects to preview a given file, the corresponding application would display the selected file in read-only mode, so as

to affect the view permission and prevent the ability to add, edit, or delete viewed files.

130. Further, A POSITA understood that a computer system had only three potential access permission modes of operation in which the computer could view data within a file: A POSITA knew that files had three access permission modes for opening or viewing files: “read-only” (i.e., enabling only viewing of the file), “read/write” (i.e., enabling viewing and editing the file) and “read, write, and delete” (i.e., enabling viewing, editing, and file deletion). EX1014, 398, 440. Accordingly, it would be obvious to a POSITA to implement the teachings of Sweat by operating the application used to preview a previous file version in read-only mode because a POSITA understood that there were a finite number of alternative ways a computer system could read data from a file, and the POSITA would have had a reasonable expectation of success of using an appropriate application in read only mode to view the contents of a previous file version.

131. Further, a POSITA would be motivated to operate the viewing application for the previewing step of Sweat in read-only mode to maintain the underlying purpose of Sweat’s file recovery system. Sweat specifically presents a list of previous versions of files to a user and allows a user to select a previous version and to select a previous version to preview. A POSITA would understand that if the viewing application is operated in a mode other than read-only (e.g., read-

write mode), this would allow the possibility that the application overwrites data in the previous file version. The POSITA would understand that allowing writes to a previous file version would defeat the purpose of Sweat's file recovery system, because writing new data to a previous file version would turn that file into a new file with new data and would thereby prevent a user from accessing and restoring the data of the previous file version. *Id.* A POSITA would therefore implement Sweat to preview previous file versions in read-only mode in order to ensure the data integrity of stored previous versions of files. *Id.*

4. Claims 4 & 12

[4/12]. The method of claim [1/10], wherein the information includes at least two previous versions of the file.

132. Sweat discloses this limitation. Sweat stores multiple file versions at its server. EX1006, 19:42-46. Sweat presents a list of previous file versions and allows a user to open and view any file version. EX1006, 10:52-57; EX1007, 31.

5. Claim 5 & 13

[5/13]. The method of claim [1/10], wherein the information for the collection includes information indicative of at least one or more of a timestamp, an original file location, and a comment of the previous versions or the file.

133. Sweat discloses this limitation. Sweat explains that project files include discussion threads containing comments that are attached to a project file. EX1006, 10:34-36; EX1007, 29. Sweats version tab 508 provides a user the ability to edit

comments for a given file using the “Edit Comment” button and to review such comments. EX1006, 10:52-55, 10:63-66; EX1007, 31, 9. Further, Sweat discloses a “general tab 502” that “displays information about the selected project file 512” including the location, date the file was created, last modified and accessed. EX1006, 10:10-14; EX1007, 30, 32. This general tab 502 is part of the information for the collection of previous versions because the general tab provides information for the specific file that the listed previous versions relate to.

6. Claim 6 & 14

a) [6.1/14.1]

[6.1/14.1] The method of claim [1/10], further comprising the steps of: (B2) after the presenting of (B) and responsive to a selection to not restoring the previous version and responsive to a second selection of a second previous version of the previous versions, version presenting a second presentable representation of the selected second previous version, the second presentable representation having at least information required for presenting at least a portion of the selected previous version; and

134. Sweat teaches this limitation. As explained, Sweat presents to a user a list of previous file versions and allows a user to select a file to preview and subsequently select to retrieve and restore that file to save it as the current version. And as explained above, for claims [6.1/14.1] for the Schneider reference in Ground 1 (*supra* ¶¶78-79), a POSITA would understand that a user would implement preview steps of a file retrieval system iteratively. Accordingly, a POSITA searching for a specific previous version of a file using the system of Sweat would undergo the

process of selecting a previous file version to preview, determining that the previewed version is not the desired version, selecting to close the previewed version, and then iteratively previewing additional previous file versions in the same manner until the user determines that the previewed file version is indeed the file version the user is seeking.

b) [6.2/14.2]

[6.2/14.2] (C2) responsive to a selection to restore the second selected previous version, retrieving the restorable representation of the second selected previous version from the remote storage location and storing the second selected previous version as the current version on the local storage location, the second selected previous version available from the restorable representation of the second selected previous version.

135. Sweat teaches this limitation. As explained above for claim [6.1/14.1] (*supra* ¶134), it would have been obvious to a POSITA that a user using the file recovery system of Sweat would iterate through previewing and rejecting multiple versions of stored previous file versions until the user finds the correct previous file version the user is searching for. As explained above for claim [1.3] (*supra* ¶¶123-24), Sweat discloses a process of responsive to a selection to restore a selected previous version, retrieving the restorable representation of the selected version from the remote storage location and storing that version as the current version on the local storage location, with the selected previous version available from the restorable representation of that version. A POSITA understood that a user would

implement that same procedure to select, retrieve, and restore a second selected previous version (i.e., where the user determines that the second selected previous version is the version the user is searching for).

7. Claims 7 & 15

[7/15]. The method of claim [1/10], wherein the restorable representation of the selected previous version comprises one or more of a compressed and encrypted version of the selected previous version.

136. Sweat discloses this limitation. Specifically, Sweat explains that “[f]iles may be automatically compressed when they are being transported to the local computer 104 or the server 106.” EX1006, 19:67-20:2. Accordingly, a POSITA would understand restorable previous file versions stored at the server to be compressed so as to reduce space of such files in storage.

8. Claims 8 & 16

[8/16]. The method of claim [1/10], wherein the restorable representation of the selected previous version comprises an incremental change to the file.

137. Sweat discloses this limitation. As explained, Sweat saves numerous file versions, with new data uploaded to the server upon a user selection to “update” an edited file to the server. EX1006, 5:54-58, 10:4-10, 17:16-21; EX1007, 63, 104. Accordingly, a POSITA would understand that successive versions of a particular file would reflect incremental changes made to that file by one or more users between such versions.

9. Claim 18

a) [18.1]

[18.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:

138. The preamble to claim 18 is identical to that of claim 1 and Sweat therefore discloses this limitation for the same reasons as explained for element [1.1].

b) [18.2]

[18.2] (A) iteratively transmitting a representation of at least a portion of a current version of a file to a remote storage location for archiving the current version responsive to a determination of an availability of the remote storage location for accepting a transmission of the representation, the current version being accessible at a computational machine, the remote storage location being accessible through a network;

139. Sweat discloses this limitation. As explained, Sweat discloses performing incremental backup in response to a user saving or editing a project file. EX1006, 5:56-58, 17:16-19; EX1007, 31. Accordingly, a POSITA would understand that Sweat iteratively transmits such backed-up file versions to Sweat's remote storage location. It would further be obvious to a POSITA to make a determination of the availability of the remote storage location prior to transmitting file back-up versions. A POSITA would understand that if the remote storage location is unavailable, for example if the remote storage is faulty, without power,

or disconnected from the network or down for maintenance, transmission of file back-ups would be unsuccessful and the user should be notified of the problem so that saved files may be backed up using a separate method and so the user can fix the problem with the remote backup location.

c) [18.3.i], [18.3.ii], [18.4], [18.5]

140. As I demonstrated at paragraph 44 and in Exhibit 1023, claim elements [18.3.i], [18.3.ii] [18.4], and [18.5] are textually identical to respective claim 1 elements [1.2.i], [1.2.ii], [1.3], and [1.4] except for internal headers (i.e., A, B, C, D). Accordingly, Sweat discloses each of elements [18.3.i], [18.3.ii], [18.4], and [18.5] for the same reasons explained for respective elements [1.2.i], [1.2.ii], [1.3], and [1.4].

d) [18.3.iii]

[18.3.iii] the restorable representation having at least information required for recovering the version V by the computational machine, the remote storage location being accessible through a network;

141. Sweat discloses this limitation for the same reasons explained for limitation [1.2.iii] (*supra* ¶122). As demonstrated at paragraph 44, limitations [18.3.iii] and [1.2.iii] are textually identical except that [1.2.iii] is narrower in requiring the additional language of “with the current version being accessible.”

10. Claim 19

a) [19.1]

[19.1] A method of restoring a file to a previous version of the file, a current version of the file being available at a local storage location, comprising the steps of:

142. The preamble to claim 19 is identical to that of claim 1 and Sweat therefore discloses this limitation for the same reasons as explained for element [1.1].

b) [19.2]

[19.2] (a) receiving a request to display a list of captured revisions for a file; (b) displaying a list of captured revisions for said file;

143. Sweat discloses this limitation for the same reasons explained for element [1.2.i/10.2.i]. Sweat discloses receiving a selection to view a “versions tab” and presenting a list of previous file versions to a user. EX1006, 10:52-57, 24:13-19; EX1007, 31, 9.

c) [19.3]

[19.3] (c) receiving a selection from said list indicating a revision of said file to restore; (d) previewing selected revision of said file, wherein said selected revision of said file is received from an Internet storage area network;

144. Sweat discloses this limitation for the same reasons explained for element [1.3/10.3]. Sweat discloses a user selection to preview a previous file

version from the presented list of previous versions by selecting the “View in Window” button. EX1006, 10:55-58; EX1007, 31, 9.

d) [19.4]

[19.4] (e) receiving another selection from said list indicating another revision of said file to restore; (f) previewing selected another revision of said file wherein said selected another revision of said file is received from a network attached storage location; and

145. Sweat teaches this limitation for the same reasons as explained for element [6.1/14.1] (*supra* ¶134). That is, a POSITA would understand that a user attempting to restore a particular file version would utilize Sweat’s file preview functionality iteratively to select and preview successive previous versions of a file until the user finds the correct version the user was searching for.

e) [19.5]

[19.5] (g) restoring selected another revision of said file following said previewing step in (f), optionally, decrypting said selected revision of said file prior to step (d) and decrypting said selected another revision of said file prior to step (f).

146. Sweat teaches this limitation for the same reasons as explained for element [1.4/10.4] and [6.2/14.2]. A user seeking to restore a particular file version, after finding the correct version through iteratively previewing multiple versions would then select to restore the file. Sweat discloses that after displaying its list of prior file versions, a user selects the “Download” button to save a selected file from the server to copy to the user’s local computer, including overwriting a current

version of that file on the user's computer. EX1006, 5:52-55, 10:58-63; EX1007, 3, 9, 31 103.

147. Sweat also discloses optionally decrypting the selected and second selected revision of the files prior to steps (d) and (f), respectively. Specifically, for all communications between the client and the server, Sweat provides the user the option to perform such data transfer in secure or non-secure mode by specifying whether the data is transferred using the HTTP or HTTPS protocol. EX1006, 8:10-16. A POSITA understood that HTTPS is an encrypted Internet communication protocol and that therefore files transferred between the client computer and remote server using this protocol would be encrypted for data transfer and then decrypted for use of the files (e.g., for viewing files). EX1014, 259-260 (describing HTTPS as a Web protocol "that provides for encryption and transmission through a secure port"). Accordingly, Sweat teaches providing an option to a user to encrypt and decrypt files prior to the file previewing steps of claim 19 steps (d) and (f).

148. Regardless, POSITA would understand the entirety of the limitation following the word "optionally" to be modified by the word and therefore be an optional element, because the remainder of the limitation following "optional" is a single clause undivided by any punctuation and Sweat therefore meets those portions of the limitation.

11. Claims 20 & 21

149. As I demonstrated in paragraph 45, claims 20 and 21 are substantially the same, with no meaningful differences in their claim language. Differences between the claims are shown below in brackets. Accordingly, my declaration analyzes the claims together.

[20/21]. The method of claim 19, [wherein said method comprises/further comprising the step of] decompressing said selected revision of said file prior to [step] (d) and decompressing said selected another revision of said file prior to [step] (f).

150. Sweat discloses this limitation. As explained above for claims 7/15 (supra ¶136), Sweat discloses automatically compressing files being transported to the local computer server. EX1006, 19:67-20:2. A POSITA would understand that compression of archived data during transmission would naturally also require decompressing an archived file version in order to display the file.

XI. GROUND 4: CLAIMS 9 AND 17 ARE OBVIOUS OVER SWEAT IN VIEW OF SCHNEIDER

151. As explained below, claims 9 and 17 are obvious over Sweat in view of Schneider. As I demonstrated in paragraph 43, claims 9 and 17 are substantively identical. Accordingly, my declaration analyzes the claims together.

1. Claims 9 & 17

[9/17]. The method of claim [1/10], wherein the method has a substantially imperceptible impact on system performance from a user's point of view by

delaying one or more operations to prevent a perceptible impact on system performance from a user's point of view.

152. As explained above, Sweat discloses all limitations of claim 1/10 (*supra* Section X.A.1). Claims 9/17 are obvious in further view of the teachings of Schneider. Specifically, Schneider discloses that when files are written to archive, “[i]n order to least impact the user's use of a computer, a goal is implementing the present invention to minimize the performance impact on reading data and rely on writes being a background activity. If writes are done in the background then overhead can be added without impacting response to the user.” EX1005, 32:9-12. Accordingly, Schneider teaches delaying a write operation such that its data backup operations have a substantially imperceptible impact on system performance from user’s point of view for the same reasons explained for claims [9/17] in Ground 1 (*supra* Section VIII.A.9). *See also id.*, 37:22-29 (“Caching writes in memory and committing them to disk during free time can reduce or eliminate the impact on the user, even though there is a tripling of time in the actual writing of the data.... [I]n theory the user had been free to continue working and so should not notice the performance degradation.”).

153. As explained, Sweat performs iterative backup of file versions by transferring saved versions to its remote storage. It would have been obvious to a POSITA to combine such file backup transmissions with Schneider’s procedure of

delaying transmissions to remote storage and transmitting only during free time in order to reduce or eliminate the impact on the user. A POSITA would have been motivated to make such a combination because Sweat and Schneider are directed to the same technology, mainly file storage, management, and retrieval systems. Further, a POSITA would make such a combination in order to provide the file backup system of Sweat with the benefit explicitly described by Schneider of making its file transmissions impact-free to the user such that the user does not notice any performance degradation. A POSITA would have had a reasonable expectation of success in combining the teachings of Sweat and Schneider as it would be simple for a POSITA to program the system of Sweat to perform transmissions of data to remote storage as a background operation and to delay such transmissions to a time when such operation would go unnoticed by the user.

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: October 30, 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. Ordelleide (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, Rotorua, 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-Founded 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

APPENDIX B

Sweat Claim Limitation	Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
1.pre. A computer-implemented apparatus comprising:	3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project.”
[1.1] a computer server;	53 - “When you add a file to your project, you select a file from your local computer which is uploaded across the Web and placed inside a project folder within your ProjectPoint Site. The file is stored on the ProjectPoint server and is available to other project members (level of interaction depends on the assigned permission).” 61 – “After you make changes to your version of the file, rename it, and then upload it to the ProjectPoint server. Send an email or attach a discussion item to let others know about the name change.”
[1.2] an interactive web site hosted on the computer server, wherein the interactive web site is configured to provide access to architectural project information, wherein one or more areas of the interactive	3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project.”

Sweat Claim Limitation	Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
web site provide for:	39 - List Project Members can see a list of project folders, but cannot view or edit them. This permission level is assigned so a member has the ability to navigate through several levels within a project, and then see and interact with a single file, as needed. 53 - “When you add a file to your project, you select a file from your local computer which is uploaded across the Web and placed inside a project folder within your ProjectPoint Site. The file is stored on the ProjectPoint server and is available to other project members (level of interaction depends on the assigned permission).”
[1.3.a] modification and organization of: a display of the interactive web site;	3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project.”
[1.3.b] site members of the interactive web site;	3 – “You can view or download files from anywhere in the world to work on them. It's easy to prevent others from overwriting files while you are working on them. All you need to do is lock the file and other project members are notified that you are changing it. When you are done, you can upload and unlock the file, while simultaneously notifying other project members of the change by email.” 10 – “For the Site Administrator, this button provides a fast way to get to frequently used administrative tasks and activities, including adding members and groups for the first time, and changing permissions.”

Sweat Claim Limitation	Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support				
[1.3.c] one or more projects including storing, organizing, and displaying a listing of drawings and text files in project folders and standard folders, and displaying drawings and text files,	12 – “You can view or download files from anywhere in the world to work on them. It's easy to prevent others from overwriting files while you are working on them. All you need to do is lock the file and other project members are notified that you are changing it. When you are done, you can upload and unlock the file, while simultaneously notifying other project members of the change by email.” 37 – “A project folder is automatically generated when you create a project, and by default, appears in the Tree view. The project folder is used to store all additional (standard) folders, project files, drawings, links and notes that are associated with the project. Project folders are easy to spot because they always appear beneath the Site level, but at the outermost level in the hierarchial Tree view.” <table> <tr> <td data-bbox="762 878 1188 951">Review</td><td data-bbox="1188 878 1612 951">Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.</td></tr> <tr> <td data-bbox="762 951 1188 1044">Edit</td><td data-bbox="1188 951 1612 1044">Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.</td></tr> </table> 38 -	Review	Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.	Edit	Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.
Review	Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.				
Edit	Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.				
[1.3.d] wherein project related administration tasks are available if a user is working with the project folders and such project related administration tasks are not available when working with	3 – “You can view or download files from anywhere in the world to work on them. It's easy to prevent others from overwriting files while you are working on them. All you need to do is lock the file and other project members are notified that you are changing it. When you are done, you can upload and unlock the file, while simultaneously notifying other project members of the change by email.” 38 - “Control over access to project information is a central function of				

Sweat Claim Limitation	Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support												
the standard folders; and	ProjectPoint. Permission levels identify the access and use of folders and files for every member assigned to project(s) in the Site. See below for permission level definitions.” Permission Level Definitions <table border="1" data-bbox="766 532 1591 992"> <tr> <td data-bbox="766 532 1178 607">No Access</td><td data-bbox="1178 532 1591 607">Keeps the Project Members from seeing the project (unless an Administrator gives them permission to view a folder or file).</td></tr> <tr> <td data-bbox="766 607 1178 716">List</td><td data-bbox="1178 607 1591 716">Project Members can see a list of project folders, but cannot view or edit them. This permission level is assigned so a member has the ability to navigate through several levels within a project, and then see and interact with a single file, as needed.</td></tr> <tr> <td data-bbox="766 716 1178 764">View</td><td data-bbox="1178 716 1591 764">Project Members can view project files, but cannot add, edit, or delete them.</td></tr> <tr> <td data-bbox="766 764 1178 834">Review</td><td data-bbox="1178 764 1591 834">Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.</td></tr> <tr> <td data-bbox="766 834 1178 922">Edit</td><td data-bbox="1178 834 1591 922">Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.</td></tr> <tr> <td data-bbox="766 922 1178 992">Administrator</td><td data-bbox="1178 922 1591 992">Project Members have full rights to a project, including adding or removing other Project Members and assigning permission levels.</td></tr> </table> 38 -	No Access	Keeps the Project Members from seeing the project (unless an Administrator gives them permission to view a folder or file).	List	Project Members can see a list of project folders, but cannot view or edit them. This permission level is assigned so a member has the ability to navigate through several levels within a project, and then see and interact with a single file, as needed.	View	Project Members can view project files, but cannot add, edit, or delete them.	Review	Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.	Edit	Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.	Administrator	Project Members have full rights to a project, including adding or removing other Project Members and assigning permission levels.
No Access	Keeps the Project Members from seeing the project (unless an Administrator gives them permission to view a folder or file).												
List	Project Members can see a list of project folders, but cannot view or edit them. This permission level is assigned so a member has the ability to navigate through several levels within a project, and then see and interact with a single file, as needed.												
View	Project Members can view project files, but cannot add, edit, or delete them.												
Review	Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.												
Edit	Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.												
Administrator	Project Members have full rights to a project, including adding or removing other Project Members and assigning permission levels.												
[1.3.e] project members of the one or more projects including defining access permissions for project members to access the project folders, the standard folders, the drawings, and the text files.	See 1.3.d.												

Appendix C

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
Abstract: “Embodiments of the invention let a user share CAD drawing files (DWG DWF, etc.), document files, or any other type of file. A user can add comments or reply to discussion threads about the file or other aspects of the project. A user can view or download files from anywhere in the world to work on them, while locking the file to prevent others from overwriting the file.” 1:52-56: “A user can view or download files from anywhere in the world to work on them, while locking the file to prevent others from overwriting the file. When the user is done, the file can be uploaded and unlocked while simultaneously notifying other Project members of the change by e-mail.”	3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project. ProjectPoint easily installs on your computer and works inside Microsoft Internet Explorer. It requires Internet access, Microsoft Windows and minimum system requirements. You can view or download files from anywhere in the world to work on them. It's easy to prevent others from overwriting files while you are working on them. All you need to do is lock the file and other project members are notified that you are changing it. When you are done, you can upload and unlock the file, while simultaneously notifying other project members of the change by email.”

APPENDIX C

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
Abstract: “Embodiments of the invention let a user share CAD drawing files (DWG DWF, etc.), document files, or any other type of file. A user can add comments or reply to discussion threads about the file or other aspects of the project. A user can view or download files from anywhere in the world to work on them, while locking the file to prevent others from overwriting the file.” 1:52-56: “A user can view or download files from anywhere in the world to work on them, while locking the file to prevent others from overwriting the file. When the user is done, the file can be uploaded and unlocked while simultaneously notifying other Project members of the change by e-mail.”	3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project. ProjectPoint easily installs on your computer and works inside Microsoft Internet Explorer. It requires Internet access, Microsoft Windows and minimum system requirements. You can view or download files from anywhere in the world to work on them. It's easy to prevent others from overwriting files while you are working on them. All you need to do is lock the file and other project members are notified that you are changing it. When you are done, you can upload and unlock the file, while simultaneously notifying other project members of the change by email.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
1:57-67: “Embodiments of the invention provide users with the ability to create their own site, define users, specify a user's access permissions for projects and files, and determine granular access levels (e.g., access for the entire site, a project, a folder, or individual files/documents) without interaction from any third party. Further, all customer activities can be logged. Additionally, one or more embodiments of the invention provide an application programming interface for third parties to create custom clients or permit direct access to a server. Such an API operates with both thick (.exe) and thin (.html) clients.	14 – “Provides the Project Administrator a way to get to frequently used administrative tasks and activities, including adding members and groups, changing permissions, and accessing a log of all project activities” 24 - “Project Administration provides the Project Administrator (or Project Manager) with easy access to frequently used administrative tasks and activities. You can name the project, indicate a "not to exceed" version limit to the files in the project, add members and groups, change permissions and access a log of all project activities.” 38 - “Control over access to project information is a central function of ProjectPoint. Permission levels identify the access and use of folders and files for every member assigned to project(s) in the Site. See below for permission level definitions.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support												
	38 - Permission Level Definitions <table border="1" data-bbox="961 418 1831 899"> <tr> <td data-bbox="961 418 1398 496">No Access</td><td data-bbox="1398 418 1831 496">Keeps the Project Members from seeing the project (unless an Administrator gives them permission to view a folder or file).</td></tr> <tr> <td data-bbox="961 496 1398 610">List</td><td data-bbox="1398 496 1831 610">Project Members can see a list of project folders, but cannot view or edit them. This permission level is assigned so a member has the ability to navigate through several levels within a project, and then see and interact with a single file, as needed.</td></tr> <tr> <td data-bbox="961 610 1398 659">View</td><td data-bbox="1398 610 1831 659">Project Members can view project files, but cannot add, edit, or delete them.</td></tr> <tr> <td data-bbox="961 659 1398 737">Review</td><td data-bbox="1398 659 1831 737">Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.</td></tr> <tr> <td data-bbox="961 737 1398 821">Edit</td><td data-bbox="1398 737 1831 821">Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.</td></tr> <tr> <td data-bbox="961 821 1398 899">Administrator</td><td data-bbox="1398 821 1831 899">Project Members have full rights to a project, including adding or removing other Project Members and assigning permission levels.</td></tr> </table> 61 – “After the file is downloaded, the application registered as the file type's editor with Microsoft Windows opens the file on your computer (if a document is a redline, link or a note, it is edited within ProjectPoint). If you want to change the registered application, you can make the change at this point.”	No Access	Keeps the Project Members from seeing the project (unless an Administrator gives them permission to view a folder or file).	List	Project Members can see a list of project folders, but cannot view or edit them. This permission level is assigned so a member has the ability to navigate through several levels within a project, and then see and interact with a single file, as needed.	View	Project Members can view project files, but cannot add, edit, or delete them.	Review	Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.	Edit	Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.	Administrator	Project Members have full rights to a project, including adding or removing other Project Members and assigning permission levels.
No Access	Keeps the Project Members from seeing the project (unless an Administrator gives them permission to view a folder or file).												
List	Project Members can see a list of project folders, but cannot view or edit them. This permission level is assigned so a member has the ability to navigate through several levels within a project, and then see and interact with a single file, as needed.												
View	Project Members can view project files, but cannot add, edit, or delete them.												
Review	Project Members can view all files in a project and can redline drawings and add or edit notes, comments, and links.												
Edit	Project Members can add, edit, or delete files for a project. These files include drawings, folders, notes, comments, links, and external files, such as images and Microsoft Word files.												
Administrator	Project Members have full rights to a project, including adding or removing other Project Members and assigning permission levels.												

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
3:2-11: “FIG. 1 schematically illustrates a hardware and software environment in accordance with one or more embodiments of the invention, and more particularly, illustrates a typical distributed computer system 100 using a network 102 to connect client computers 104 to server computers 106. A typical combination of resources may include a network 102 comprising the Internet, LANs, WANs, SNA networks, or the like, clients 104 that are personal computers or workstations, and servers 106 that are personal computers, workstations, minicomputers, or mainframes.”; <i>see also</i> 3:12-55.	3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project. ProjectPoint easily installs on your computer and works inside Microsoft Internet Explorer. It requires Internet access, Microsoft Windows and minimum system requirements.” 53 - “When you add a file to your project, you select a file from your local computer which is uploaded across the Web and placed inside a project folder within your ProjectPoint Site. The file is stored on the ProjectPoint server and is available to other project members (level of interaction depends on the assigned permission).”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
3:12-18: “The Internet 102 connects client computers 104 executing Web browsers 108 to server computers 106 executing Web servers 110. The Web browser 108 is typically a program such as Netscape's Navigator or Microsoft's Internet Explorer. The Web server 110 is typically a program such as IBM's HyperText Transport Protocol (HTTP) Server or Microsoft's Internet Information Server.”	3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project. ProjectPoint easily installs on your computer and works inside Microsoft Internet Explorer. It requires Internet access, Microsoft Windows and minimum system requirements. You can view or download files from anywhere in the world to work on them. It's easy to prevent others from overwriting files while you are working on them. All you need to do is lock the file and other project members are notified that you are changing it. When you are done, you can upload and unlock the file, while simultaneously notifying other project members of the change by email.” 103 – “Server Computer that hosts the ProjectPoint service. This remote computer, database, and file storage system stores nroject information and provides access to it over the Web. The server manages this access using requests from a client.”

3:57-4:19: “Software (referred to as ProjectPoint™) provides for the unique organization of folders as containers for storing, managing, and sharing files and discussions for an architectural project. The ProjectPoint™ software may be downloaded and installed on a computer. Further, ProjectPoint™ may work in combination with or independently from a user's internet browser such as the Internet Explorer available from Microsoft™. Once installed, a user may initialize the ProjectPoint™ software by starting the ProjectPoint™ application or by attempting to access a web site (or an area of a web site) that is configured to work with ProjectPoint™ (at which point, the ProjectPoint™ application will automatically initialize on the user's browser 108). Once the software is initialized, the user is prompted to log in using a user name and password (see description below). Upon successful login, information is downloaded from the web site/server 106 to the local computer 104 and displayed in accordance with the ProjectPoint™ software. Thus, ProjectPoint™ may be viewed as an application and as a web site and applet with a series of pages and programming modules utilized by a web server 110 and web browser 108.

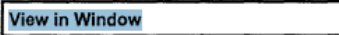
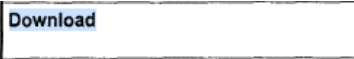
3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project.

ProjectPoint easily installs on your computer and works inside Microsoft Internet Explorer. It requires Internet access, Microsoft Windows and minimum system requirements.

You can view or download files from anywhere in the world to work on them. It's easy to prevent others from overwriting files while you are working on them. All you need to do is lock the file and other project members are notified that you are changing it. When you are done, you can upload and unlock the file, while simultaneously notifying other project members of the change by email.”

9 -

Download	Copies the file to your local computer. Does not lock or open the file.
----------	---

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
5:43-44: "View in Window: Opens the item for viewing in a new window"	9 -  Opens the item for viewing in a new window. 31 – “Note: When you choose View in Window, the file will open and be displayed in a separate window outside of ProjectPoint and your browser.”
5:52-55: "Download: Copies the file or folder to the user's local computer. The user can include subfolders and overwrite files that exist on the user's local computer, as desired. Does not lock or open the file."	33 – “Project members see only the project(s) to which they belong. If a user is not assigned to any project, they will not see any projects when they log in. This gives you complete control of security for any sensitive material. Assign permissions carefully - once the user has edit permission for a project, they have the ability to overwrite files.” 79-  Copies the file to your local computer. <i>Does not lock or open the file.</i>

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
5:56-58: “Update - Use this command to select and upload a new version of a file to the server (typically for a file that has an older version on the server). Usually used after editing.”	61 – “After you complete an editing session, it's a good idea to save the file to your local computer. If the file is ready to be shared with other project members, the updated version can be copied back to the ProjectPoint server using the Update File Wizard (In the Standard Toolbar, click the Update File icon). The current version of the file is replaced by the updated version, and the file is unlocked. Otherwise, you can continue to edit the file until you are ready to share the changes with others.” 104 – “Update: To choose a local document and add it to the project site as a new version of the currently selected document. The local document does not have to have the same name as the document it is updating.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
7:67-8:9: “To specify a file to be used for the site information page, the user selects the choose button 302. Choose button 302 present the user with options to use the default page (uses the supplied site information page URL address), to browse the site for a page (allows the user to select a file by browsing through a directory of all files on the current site), or to browse local drives (allows the user to select a file by browsing through a directory of all of the files on the user's local computer and the internet), or to enter the name of the desired file.”	21 - “To create a link to the Site Information page 1. In Information click the right arrow, and then choose one of the following: • Default - accepts the current URL address. • Browse in Home - provides a directory of all ProjectPoint files. • Browse - provides a directory of all the files on your local computer and the Internet.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
9:19-34: “As illustrated in FIG. 4, to change the number of file versions that can be saved, the user merely selects button 406. In response thereto, a dialog is displayed that presents the user with options to “save all” or to set the number of versions allowed in a “limit to” area where the user can enter a number. By selecting “save all”, an unlimited number of versions can be saved. Once the number of saved versions is limited, each time a version is saved, the oldest version is “bumped” up to the next number. Once the version limit is reached, saving a new version deletes the oldest version. For example, if a user uploads 10 versions of a document and the version limit is set equal to 5. The versions available would be 10, 9, 8, 7, and 6. If a new version is then added, versions available would be 11, 10, 9, 8, and 7. Version 6 is removed from the system (since the maximum version limit would be exceeded if version 6 remained).”	21 – “To specify a number of file versions 1. Use the arrows in Versions to indicate the maximum number of file versions to be saved.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
10:10-16: “Project files general tab 502 displays information about the selected project file 512 including the file name, author, version, file type (text, drawing, or note), location, file size, date the file was created, last modified and accessed, edit lock, and the permission set lock. The modification of information in general tab 502 may be limited to a site or project administrator.”	28 - Project Files  Provides a hierarchial list (a Tree View, similar to Explorer) of all the folders and files in the project. To reveal or hide levels, click the + and - symbol to the left of the folder in the list. When you select an item in the <i>Tree</i> View, the details (organized in four tabs) appear in the Details view to the right.  To learn more about working with the information in the tabs, see General, View, Discussions, and Versions. 30 – “Project File General Tab The General Tab is used to: • View and change the Project name. • Determine the "not to exceed" limit on file versions that are saved with each file in the project. • Change a link (URL address) to the Project Information page. • Change a link (URL address) to the Project Directory. To rename a project 1. Select the current name in Project Name, and then enter a new one. To specify a number of file versions 1. Use the arrows in Versions to indicate the maximum number of file versions to be saved.

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
	To create a link to the Project Information page 1. In Information click the right arrow, and then choose one of the following: • Default - accepts the current URL address. • Browse in Home - provides a directory of all ProjectPoint files. • Browse - provides a directory of all the files on your local computer and the Internet.”
10:22-27: “Project files view tab 504 is used to open and display the contents of a project file in the details view portion of the screen. A user can also add lines and text to drawing files, and save the file with the markups (redlines) as a DWF file to a location of the user's choice (e.g., on the server 106 or on the local computer 104).”	32 - “Project File View Tab The View tab is used to open and display the contents of a project file in the Details view (located on the right side of the screen). Depending on your permission level, you can use the right-click menu to get to many of the most often used commands (such as edit, download, Properties, and so on). For detailed procedures, see the Viewing a File section.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
10:34-41: “The project files discussion tab 506 displays all of the discussion threads that are attached to the selected project file 512. A discussion thread contains comments, questions, response, or other information input by a user. Additionally, each user entry is identified by the user that entered the entry, the date of the entry, and the version of the entry. Project members do not have to be active participants in discussions to read the comments in a discussion thread.”	29 – “Discussion Tab The Discussion tab is used to display all the discussion threads relating to the selected project file. Discussions are especially useful if you want to chain messages together to exchange information and make decisions as a team. Discussion threads are organized and stored along with the related file. Project Members do not have to be active participants in discussions to review the information contained in them. For detailed procedures, see the Starting a Discussion Thread and Replying to a Discussion Thread sections.”

10:51-11:3: "The project files versions tab 508 provides a list of all of the currently selected files. The versions tab 508 is used to open and view any file version, download a selected file to a local computer 104, and review or edit comments that are attached to the selected file version. To open a version, the user selects the version from a list displayed in versions tab 508 and selects a "View in Window" button. Alternatively, the user can click the view tab 504. To download a version of the file from the server 106 to the local computer 104, the user selects the desired version and clicks a "Download" button. Thereafter, the user identifies/selects the target location to save the selected file. The file is then copied to the location specified. To edit a comment, the user selects the version desired, and clicks an "Edit Comment" button. Thereafter, the user can make the desired changes to the comments as needed. The Edit Comment option may only be available to administrators and the owner of the file. Each of the functions (i.e., download, view in window, and edit comment) may also be available from a menu displayed by clicking the right-mouse button."

11:4-9: "An additional function that may be

31 - "Project File Versions Tab

The Versions tab provides a list of all versions associated with the currently selected file. You can easily open and view any file version from this tab.

To open a previous version

1. Select the version you want, right-click, and then choose View.

Note: When you choose View in Window, the file will open and be displayed in a separate window outside of ProjectPoint and your browser.

To learn more about creating a new file version or editing a version, read the Creating a New Version and Editing a Version sections."

9 -	View in Window	Opens the item for viewing in a new window.
	Download	Copies the file to your local computer. <i>Does not lock or open the file.</i>

61 – "After you complete an editing session, it's a good idea to

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
available from the version tab 508 is that of updating a file. If the user elects to update a file (e.g., by selecting the “Update” option from the right-click menu), the user is prompted to select the file on the local computer 104 that corresponds to the file 512 to be updated on server 106.”	save the file to your local computer. If the file is ready to be shared with other project members, the updated version can be copied back to the ProjectPoint server using the Update File Wizard (In the Standard Toolbar, click the Update File icon). The current version of the file is replaced by the updated version, and the file is unlocked. Otherwise, you can continue to edit the file until you are ready to share the changes with others.”
13:10-21: "Project member information can be changed over the life of a project. As needed, administrators can: add or remove members from a project; notify members that they have been added or removed from a project, or, that there is a change in their assigned permission level; and change members' permission levels. Project members have viewing and access permission only to the projects which they belong. If a site member is not assigned to any project, the project files tree view display 206 is empty. Additionally, members can have read-only permissions for one project and edit permission for another project (as defined by the project or site administrator)."	18 – “For the Site Administrator, provides a quick way to get to frequently used administrative tasks and activities including adding members and groups for the first time, changing permissions, and accessing a log of all site activities.” 20 – “Site Administration Members The Members tab is used to: • Add or remove members from a site. • Change members permission levels. • Notify members that they have been added or removed from a site, or, that there is a change in their assigned permission level.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
	35 – “Project members see only the project(s) to which they belong. If a user is not assigned to any project, they will not see any projects when they log in. This gives you complete control of security for any sensitive material. Assign permissions carefully - once the user has edit permission for a project, they have the ability to overwrite files. Many project members need just read-only permission (View or List permission). Note that a member can have read-only permission for one project and edit permission for another project. It is up to you as the Administrator to decide who gets permission for each project, folder, and document.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
13:10-13:21: “Project member information can be changed over the life of a project. As needed, administrators can: add or remove members from a project; notify members that they have been added or removed from a project, or, that there is a change in their assigned permission level; and change members' permission levels. Project members have viewing and access permission only to the projects which they belong. If a site member is not assigned to any project, the project files tree view display 206 is empty. Additionally, members can have read-only permissions for one project and edit permission for another project (as defined by the project or site administrator).” 14:43-14:61: “If a member has project administrator permission, the member can add folders and files to the projects that the member is responsible for, and assign members and the appropriate member permissions to the folders and files within each project.	35 – “Many project members need just read-only permission (View or List permission). Note that a member can have read-only permission for one project and edit permission for another project. It is up to you as the Administrator to decide who gets permission for each project, folder, and document.” 36 – “When you join a project, your project Administrator assigns you one of four permission levels: Edit, Review.View or List. An Administrator can change your permission level at any time over the life of the project, according to your role and the needs of the project, and can override the permissions for an individual folder or document. When new folders and project files are added to a project, the permission list is "inherited" according to the permission list and levels of access assigned at the project (top) level. If you find that you need different accessibility to project information, contact your Project Administrator. If you have Edit permission, you can add, edit, update, rename,or delete project folders and files, and share all information with other project members. Project files include drawings, folders, notes, comments, links, and external files, such as Images and Microsoft Word files. If you have Review permission, you can view all the files in a project and can redline drawings and add or edit notes, comments, and links.

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
Edit permission provides the ability to add, edit, update, rename, or delete project folders and files, and share all information with other project members. Review permission provides the ability to view all of the files in a project, redline drawings, and add or edit notes, comments, and links. View permission provides the ability to view project files without the ability to add, edit, or delete them.”	If you have View permission, you can view project files, but cannot add, edit, or delete them. If you have List permission, you can see a list of folders in each project, but cannot view them. Sometimes, you are also assigned a different permission level at a particular folder or file level so you can access and interact with that particular item.” 38 – “Occasionally, permission levels can be changed to allow project members to view and interact with individual folders or project files in an exceptional way. For example, if you want a project member with View permission to be able to edit all files within a particular folder, or edit a single file, just change the project member's permission level to Edit (within the Properties for that folder or file). It is then available to the project member for editing. To modify permission levels at the folder or file level, follow the procedures in the Assigning Permissions at the Folder or File Level section.”

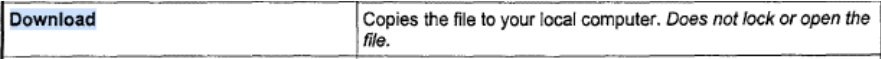
Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
15:23-28: “Thus, if a user has Administrator or Edit permission, the user can lock a file, preventing other project members from editing the file. When the file is locked it remains on server 106 but cannot be edited. However, if the user has Administrator permission, the user can unlock a locked file at any time, as needed. Further, even though a file is locked, it may be downloaded by those with Administrator or Edit permission.”	64 – “Lock and Unlock a File In ProjectPoint, you can lock a file. By doing so, the selected file (stored on the server) is locked and other project members are not able to edit the file. Anyone who tries to edit a locked file will receive a message that the file is locked (the message doesn't contain any extra details about who locked the file, or what is being done with the item). Note: Even though a file is locked, it can be downloaded or viewed by other project members depending on the assigned permission levels. For example, if you are an Administrator or Editor and attempt to download or edit a locked file, a message appears to let you know about the existing lock and asks if you still want to continue the operation. If you continue, a copy of the file is downloaded.”
17:16-19: "When a member saves changes or adds new information to files in the client 104 (must have Administrator or Edit permission; those with Review permission can add and save comments only), the new data is uploaded to the server 106."	61 – “After you complete an editing session, it's a good idea to save the file to your local computer. If the file is ready to be shared with other project members, the updated version can be copied back to the ProjectPoint server using the Update File Wizard (In the Standard Toolbar, click the Update File icon). The current version of the file is replaced by the updated version, and the file is unlocked. Otherwise, you can continue to edit the file until you are ready to share the changes with others.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support		
	104 – “Update: To choose a local document and add it to the project site as a new version of the currently selected document. The local document does not have to have the same name as the document it is updating.”		
17:53-61: “A strong relationship may exist between the files and folders stored on the server 106 and those downloaded to a local computer 104. The first time a user downloads and updates files and folders, a directory path is identified for the selected folder(s), which can include all of its sub-folders and files. Thereafter, each time a user downloads and edits items on the local computer 104, or updates them (which creates a new version on the server 106), the default path is automatically used.”	11 - <table border="1" data-bbox="957 613 1829 690"> <tr> <td data-bbox="957 613 1245 690">Edit</td><td data-bbox="1251 613 1829 690">Opens the <i>Save File to Edit As</i> dialog box. Save the file to a folder of your choice on your local computer. This button is not available if you have only List or View permission.</td></tr> </table>	Edit	Opens the <i>Save File to Edit As</i> dialog box. Save the file to a folder of your choice on your local computer. This button is not available if you have only List or View permission.
Edit	Opens the <i>Save File to Edit As</i> dialog box. Save the file to a folder of your choice on your local computer. This button is not available if you have only List or View permission.		

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
18:4-14: "If a user adds new files to a folder that has been downloaded to the local computer 104, the files are automatically uploaded with the original content of the folder when the user updates to the server 106. Further, when downloading or uploading a folder, only the documents that are changed are moved across the Internet 102. For example, if a user downloads a folder containing 100 documents and during the week, 10 of the documents change, when the user next downloads the same folder, only the 10 changed documents are moved to the client. Uploading documents and folders operates in a similar manner."	61 – "After you complete an editing session, it's a good idea to save the file to your local computer. If the file is ready to be shared with other project members, the updated version can be copied back to the ProjectPoint server using the Update File Wizard (In the Standard Toolbar, click the Update File icon). The current version of the file is replaced by the updated version, and the file is unlocked. Otherwise, you can continue to edit the file until you are ready to share the changes with others."

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
19:15-28: “Notes are text files that are created and stored in folders within a project. When a user creates a note, a stand-alone file is created that can be edited or updated directly within the project site, without requiring an additional application. Additionally, some simple formatting options (font, point size, etc.) may be available when creating a note. Additionally, a user can create a hyperlink in a note. For example, a note may be created about the specifics of a folder or file, or, simply to record a user's thoughts. Then, as with any other file in ProjectPoint, users can email the appropriate project member(s) to notify them that the note has been posted. Each note may be stored in the associated project folder. Unlike Comments (and Discussion threads) notes cannot be chained together.”	55 – “Notes are text files that are created and stored with folders or files in a project. When you create a note, you are creating a stand-alone file (similar to a read me file) that can be edited or updated. There are some simple formatting options (font, point size, etc.) available when you create a note, similar to the way Microsoft Word works. There are lots of useful ways to use Notes. For example, you can create a note about the specifics of a folder or file, or, simply record your thoughts. Then, email the appropriate project member(s), and include the URL address to the corresponding note in the body of the email. Each note is stored with the associated project file. Unlike Comments (and Discussion threads) you cannot chain notes together.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
19:64-20:29: "A user can easily download a project or standard folder and all of its contents to a local computer 104, make changes to some (or all) of the downloaded files, and then upload the modified files by choosing the Update File Wizard. Files may be automatically compressed when they are being transported to the local computer 104 or the server 106. To download a folder, a user selects the folder or file to download from view 206. The user then right-clicks and chooses Download from the menu. In response, a Download dialog box appears. In the To: field, the user enters the destination path for the downloaded item on the local computer 104 (or the user can click browse and navigate to the correct destination location). ... In the Replace locally modified files field, the drop down menu may be utilized to choose one of the following: Ask before replacing - Provides a prompt before downloading a folder or file from the server 106 that may have been modified more recently on the local computer 104.	61 – “When you click Edit, the selected file (stored on the server) is locked so that other project members cannot change the file at the same time you are working with it. Note: A locked file can still be viewed. If you have Administrator or Edit permission, and attempt to download or edit a locked file, a message appears to let you know about the existing lock and asks if you still want to continue the operation. If you continue, the file is downloaded and saved to your computer, but will remain locked by the person who locked the file to start with. After the file is downloaded, the application registered as the file type's editor with Microsoft Windows opens the file on your computer (if a document is a redline, link or a note, it is edited within ProjectPoint). If you want to change the registered application, you can make the change at this point. ... After you complete an editing session, it's a good idea to save the file to your local computer. If the file is ready to be shared with other project members, the updated version can be copied back to the ProjectPoint server using the Update File Wizard (In the Standard Toolbar, click the Update File icon). The current version of the file is replaced by the updated version, and the file is unlocked. Otherwise, you can continue to edit the file until you are ready to share the changes with others.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
Replace - Automatically downloads a folder or file from the server 106 with files that may have been modified more recently on the local computer 104. Don't replace - If a more recently modified folder or file exists on the local computer 104, skips downloading files from the server 106. Once the above steps are complete, the user can click Download to complete the download process."	63 – “In ProjectPoint, you can update a file after you are finished editing it. The updated version is copied from your local computer back to the ProjectPoint server by using the Update File Wizard. The current version of the file is replaced by the updated version, and the file is unlocked.” 9 - 
23:38-41: “Viewing Files and Multiple Views To view project files including drawing files and markup files, a user must install the appropriate viewer to open and view drawing files.”	59 – “After the file is downloaded, the application registered as the file type's editor with Microsoft Windows opens the file on your computer (if a document is a redline, link or a note, it is edited within ProjectPoint). If you want to change the registered application, you can make the change at this point.” 73 – “After the file is downloaded, the application registered as the file type's editor with Microsoft Windows opens the file on your computer (if a document is a redline, link or a note, it is edited within ProjectPoint). If you want to change the registered application, you can make the change at this point.”

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support
24:6-20: "View in Window-This option is used to open a selected file in the Details view (so the user has access to all of the tabbed options) in a separate window. Because the selected file information appears in a separate window, a user can then view any number of details about two files at the same time.	31 - "Project File Versions Tab The Versions tab provides a list of all versions associated with the currently selected file. You can easily open and view any file version from this tab. To open a previous version 1. Select the version you want, right-click, and then choose View. Note: When you choose View in Window, the file will open and be displayed in a separate window outside of ProjectPoint and your browser. To learn more about creating a new file version or editing a version, read the Creating a New Version and Editing a Version sections." 9 - View in Window Opens the item for viewing in a new window.

Portion of Sweat Referenced Relied On	Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support				
27:60-67: "Embodiments of the invention track all files, comments, and notes that are added to the project, and displays them in a What's New List. The list contains files that have been added or changed in the project since the last time the user reviewed them. Once a user have viewed the contents of a file, it is automatically marked as read and is no longer listed in What's New. A user may also manually mark a file as read." As an example, if a user wants to view a drawing, but also needs to view the related discussion thread, the drawing file is selected from the list in the Tree view 206, the user right-clicks, and then chooses View in Window. The selected file appears in a separate window. The user sees the contents of the file (View tab 504) and also has access to all the other tabbed options (General 502, Discussion 506 and Activity Log)."	31 - "Project File Versions Tab The Versions tab provides a list of all versions associated with the currently selected file. You can easily open and view any file version from this tab. To open a previous version 1. Select the version you want, right-click, and then choose View. Note: When you choose View in Window, the file will open and be displayed in a separate window outside of ProjectPoint and your browser. To learn more about creating a new file version or editing a version, read the Creating a New Version and Editing a Version sections." 9 - <table border="1" data-bbox="961 971 1892 1068"> <tr> <td data-bbox="961 971 1318 1008">View in Window</td><td data-bbox="1318 971 1892 1008">Opens the item for viewing in a new window.</td></tr> <tr> <td data-bbox="961 1008 1318 1068">Download</td><td data-bbox="1318 1008 1892 1068">Copies the file to your local computer. <i>Does not lock or open the file.</i></td></tr> </table>	View in Window	Opens the item for viewing in a new window.	Download	Copies the file to your local computer. <i>Does not lock or open the file.</i>
View in Window	Opens the item for viewing in a new window.				
Download	Copies the file to your local computer. <i>Does not lock or open the file.</i>				

Portion of Sweat Referenced Relied On

Fig. 1

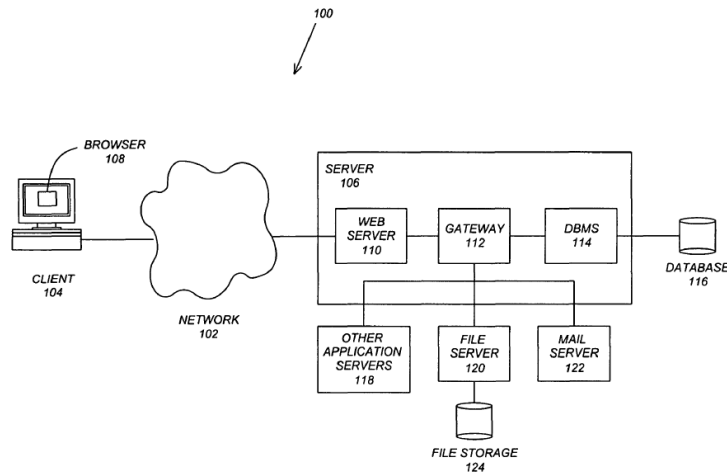


FIG. 1

Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support

3 – “ProjectPoint is a project hosting service that you access with a web browser, so all you need is access to the Web and a Browser, and you're up and running. You can communicate and collaborate with all the members in your design team or construction project. ProjectPoint lets you share CAD drawing files (DWG and DWF), text files, or any other file. You can add comments or reply to discussion threads about a file or any other aspect of the project.

ProjectPoint easily installs on your computer and works inside Microsoft Internet Explorer. It requires Internet access, Microsoft Windows and minimum system requirements. You can view or download files from anywhere in the world to work on them. It's easy to prevent others from overwriting files while you are working on them. All you need to do is lock the file and other project members are notified that you are changing it. When you are done, you can upload and unlock the file, while simultaneously notifying other project members of the change by email.”

Portion of Sweat Referenced Relied On

Fig. 2

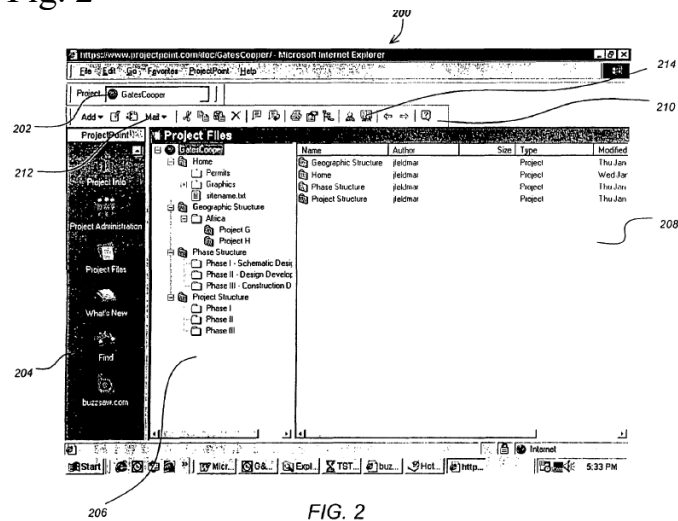
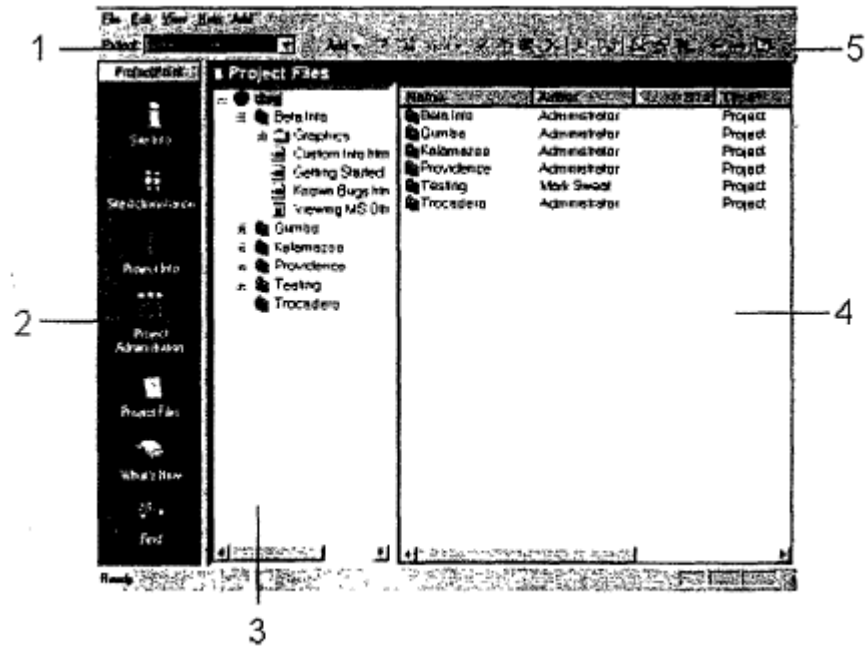


FIG. 2

Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support



7 -

Portion of Sweat Referenced Relied On

Fig. 4

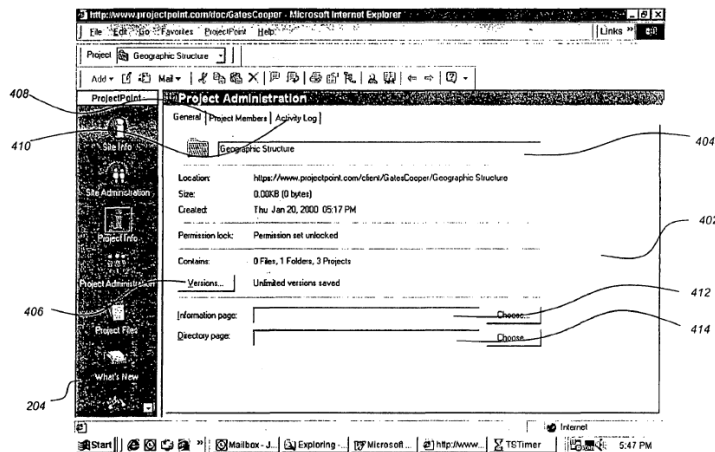


FIG. 4

Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support

25 – “Project Administration Activity Log Tab

The Activity Log tab is used to view and customize data by choosing from a series of "filters" that show or hide information according to what you want to view. So, you can view a broad array of information, or, zero in on a very specific piece of data according to Date, Users, Actions, or Types. For detailed information, see Working with Activity Logs.”

30 – “Project File General Tab

The General Tab is used to:

- View and change the Project name.
- Determine the "not to exceed" limit on file versions that are saved with each file in the project.
- Change a link (URL address) to the Project Information page.
- Change a link (URL address) to the Project Directory.”

Portion of Sweat Referenced Relied On

Fig. 5

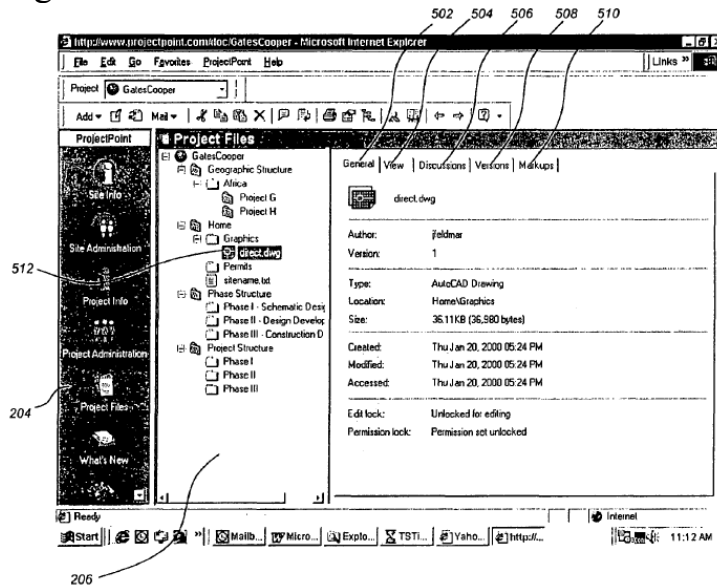
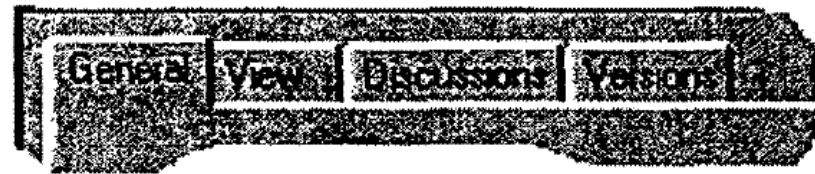


FIG. 5

Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support

68 –



31 – “Project File Versions Tab

The Versions tab provides a list of all versions associated with the currently selected file. You can easily open and view any file version from this tab.”

Portion of Sweat Referenced Relied On

Fig. 10

Activity Log List Options

Include in list

Users:

- ☐ Administrator
- ☒ Joe Editor

Actions:

- ☐ add a group
- ☐ add a group member
- ☒ add a user
- ☐ change password
- ☒ login
- ☐ logout
- ☐ remove a group
- ☐ remove a group member
- ☐ remove a user
- ☐ update a group
- ☐ update a user

Item Types:

- ☐ Group
- ☒ Group member
- ☒ Project member
- ☒ User

Select All Clear Select All Clear Select All Clear

OK Cancel Help

1000

FIG. 10

Exemplary Portion of Sweat Provisional (EX1007) Demonstrating Written Description Support

69 – “Work with Activity Logs

With ProjectPoint, it is possible to customize and view data at the project level by choosing from a series of "filters" that show or hide information according to what you want to view. So, you can view a broad array of information or, zero in on a very specific piece of data according to Date, Users, Actions, or Types.”