Nos. 2022-1600, 2022-1709

United States Court of Appeals for the Federal Circuit

MIRROR WORLDS TECHNOLOGIES, LLC,

Plaintiff-Appellant

v.

META PLATFORMS, INC.,

Defendant-Cross-Appellant

Appeals from the United States District Court for the Southern District of New York in No. 1:17-cv-03473-JGK, Judge John G. Koeltl.

CORRECTED NON-CONFIDENTIAL PRINCIPAL BRIEF OF PLAINTIFF-APPELLANT MIRROR WORLDS TECHNOLOGIES, LLC

October 6, 2022

Marc A. Fenster
mfenster@raklaw.com
Benjamin T. Wang
bwang@raklaw.com
James Tsuei
jtsuei@raklaw.com
Russ, August & Kabat
12424 Wilshire Blvd., 12th Floor
Los Angeles, CA 90025
Tel: (310) 826-7474

Counsel for Plaintiff-Appellant Mirror Worlds Technologies, LLC

CLAIMS AT ISSUE

U.S. Patent No. 6,006,227, Claims 13, 14, and 17 (Appx81):

13. A method which organizes each data unit received by or generated by a computer system, comprising the steps of:

generating a main stream of data units and at least one substream, the main stream for receiving each data unit received by or generated by the computer system, and each substream for containing data units only from the main stream;

receiving data units from other computer systems;

generating data units in the computer system;

selecting a timestamp to identify each data unit;

associating each data unit with at least one chronological indicator having the respective timestamp;

including each data unit according to the timestamp in the respective chronological indicator in at least the main stream; and

maintaining at least the main stream and the substreams as persistent streams.

14. The method of claim 13, wherein each timestamp is selected from the group consisting of: past, present, and future times.

17. The method of claim 13, wherein each data unit includes textual data, video data, audio data and/or multimedia data.

U.S. Patent No. 7,865,538, Claim 1 (Appx112):

1. A method of operating a computer system comprising:

providing the computer system with documents from diverse applications in respective formats unique to the respective applications;

causing the computer system to automatically, without user interaction and without requiring a user to designate directory structures or other pre-imposed document categorizations structures, store the provided documents as a time-ordered main stream of documents associated with respective automatically generated time indicators;

said time-ordered main stream being unbounded to thereby accommodate documents associated with time indicators related to past, present and future times;

said time-ordered main stream requiring no fixed beginning or end and being maintained and being selectively retrievable and searchable by the computer system;

said computer system maintaining the main stream live and responsive to subsequent events by automatically incorporating therein new documents as provided to the computer system while maintaining the thus expanded main stream time-ordered;

providing selected search criteria;

causing said computer system to search said time-ordered main stream according to said search criteria and use search results to create a time-ordered substream of documents from the main time-ordered stream;

further causing said computer system to maintain said substream live and responsive to subsequent events by automatically incorporating therein new document provided to the computer system that meet the search criteria while maintaining the thus expanded substream time-ordered;

displaying at least selected portion of the live main stream or substream on computer display means as a display reflecting the time-ordered nature thereof;

automatically showing on the display means a display of a glance view of a displayed document in response to touching with a cursor a screen area associated with the document;

said glance view being an abbreviated version of the document and indicative of content thereof; and

said showing of the glance view occurring essentially instantaneously in response to said touching with the cursor of the screen area associated with the document.

U.S. Patent No. 8,255,439, Claim 1 (Appx137-138):

1. A method of operating a computer system comprising:

providing the computer system with documents in respective formats according to respective different applications through which the provided documents are generated or modified, which formats differ from one of the document to another for at least some of said provided documents, said provided documents being delivered to the computer system or generated by the computer system;

storing at least some of the documents provided to the computer system in computer storage;

said computer system being configured to automatically generate and store in computer storage respective representations related to the documents provided thereto, thereby forming a main collection of document representations;

said computer automatically generating and storing said main collection of document representations without requiring a user to designate a directory structure, a physical location for storage of document representations or corresponding documents, or another pre-imposed document categorization structure for each of said document representations or documents;

said automatically generated and stored document representations being in a consistent format despite differences in format from one to another of the documents corresponding thereto;

said automatically generated and stored representations of said documents including respective automatically generated time indicators associated with the documents corresponding to said representations;

said automatically generated and stored representations of said documents further including respective automatically generated information relating the document representation to the respective documents provided to and stored in said computer system;

said automatically generated and stored main collection of document representations being unbounded in time and size and being configured to include documents associated with time indicators related to future times as well as to past and present times;

said automatically generated and stored main collection of document representations requiring no fixed beginning or end and being non-transitory and selectively searchable by the computer system;

providing selected search criteria;

causing said computer system to perform a first search of at least said main collection of document representations according to selected first search criteria, to provide first search results, and to utilize said first search results to generate a first sub-collection of document representations related to a respective sub-collection of the documents provided to the computer system;

selectively causing the computer system to display on a computer screen graphical depictions of only a first portion of said first sub-collection of document representations generated by utilizing said first search results, said first portion corresponding to only a portion of the documents provided to and stored in the computer system;

said first portion of said first document sub-collection of document representations corresponding to a multi-document portion of the documents provided to and stored in said computer system;

said computer system being configured to maintain at least one of said the main collection of document representations and said first sub-collection of document automatically responsive to events subsequent to the providing of said first search results such that additional document representations corresponding to additional documents provided to the computer system subsequent to an initial display of said first portion of the first sub-collection of document representations and meeting said

selected search criteria are automatically included in a subsequent display of graphical depictions of one or more portions of said first sub-collection of document representations;

said additional document representations also including automatically generated respective time indicators associated with the documents subsequently provided to the computer system;

automatically showing on the computer screen a display of a glance view of a displayed document depiction while continuing to show on the screen plural displayed graphical depictions of respective plural document representations;

said glance view being an abbreviated version of the document corresponding to the graphical depiction and being indicative of content thereof; and

said showing of the glance view occurring in response to a user designating a displayed document representation by interacting with a screen area currently associated with the graphical depiction, without requiring the user to click on the designated screen area in order to enable such showing of the glance view.

FORM 31. Certificate of Confidential Material **Form 31**
July 2020

UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT

CERTIFICATE OF CONFIDENTIAL MATERIAL

Case Number: 2022-1600, 2022-1709

Short Case Caption: Mirror Worlds Technologies, LLC v. Meta Platforms, Inc.

Instructions: When computing a confidential word count, Fed. Cir. R. 25.1(d)(1)(C) applies the following exclusions:

- Only count each unique word or number once (repeated uses of the same word do not count more than once).
- For a responsive filing, do not count words marked confidential for the first time in the preceding filing.

The limitations of Fed. Cir. R. 25.1(d)(1) do not apply to appendices; attachments; exhibits; and addenda. *See* Fed. Cir. R. 25.1(d)(1)(D).

The foregoing document contains 13 number of unique words (including numbers) marked confidential.

☑ This number does not exceed the maximum of 15 words permitted by Fed. Cir. R. 25.1(d)(1)(A).

☐ This number does not exceed the maximum of 50 words permitted by Fed. Cir. R. 25.1(d)(1)(B) for cases under 19 U.S.C. § 1516a or 28 U.S.C. § 1491(b).

☐ This number exceeds the maximum permitted by Federal Circuit Rule 25.1(d)(1), and the filing is accompanied by a motion to waive the confidentiality requirements.

Date: 10/06/2022

Signature: /s/ Benjamin T. Wang

Name: Benjamin T. Wang

Save for Filing

1. Represented Entities. Fed. Cir. R. 47.4(a)(1).	**2. Real Party in Interest.** Fed. Cir. R. 47.4(a)(2).	**3. Parent Corporations and Stockholders.** Fed. Cir. R. 47.4(a)(3).
Provide the full names of all entities represented by undersigned counsel in this case.	Provide the full names of all real parties in interest for the entities. Do not list the real parties if they are the same as the entities. ☑ None/Not Applicable	Provide the full names of all parent corporations for the entities and all publicly held companies that own 10% or more stock in the entities. ☐ None/Not Applicable
Mirror Worlds Technologies, LLC		Network-1 Technologies, Inc.

☐ Additional pages attached

4. Legal Representatives. List all law firms, partners, and associates that (a) appeared for the entities in the originating court or agency or (b) are expected to appear in this court for the entities. Do not include those who have already entered an appearance in this court. Fed. Cir. R. 47.4(a)(4).

☐ None/Not Applicable ☑ Additional pages attached

Jacob Buczko of Russ August & Kabat	Adam Hoffman of Russ August & Kabat	Steven Udick formerly of Russ August & Kabat
Arka D. Chatterjee formerly of Russ August & Kabat	Theresa M. Troupson formerly of Russ August & Kabat	

5. Related Cases. Provide the case titles and numbers of any case known to be pending in this court or any other court or agency that will directly affect or be directly affected by this court's decision in the pending appeal. Do not include the originating case number(s) for this case. Fed. Cir. R. 47.4(a)(5). See also Fed. Cir. R. 47.5(b).

☑ None/Not Applicable ☐ Additional pages attached

6. Organizational Victims and Bankruptcy Cases. Provide any information required under Fed. R. App. P. 26.1(b) (organizational victims in criminal cases) and 26.1(c) (bankruptcy case debtors and trustees). Fed. Cir. R. 47.4(a)(6).

☑ None/Not Applicable ☐ Additional pages attached

Save for Filing

TABLE OF CONTENTS

CERTIFICATE OF INTEREST i
CONFIDENTIAL MATERIAL OMITTED vi
TABLE OF AUTHORITIES vii
STATEMENT OF RELATED CASES 1
STATEMENT OF JURISDICTION 2
STATEMENT OF THE ISSUES FOR APPEAL 3
I. INTRODUCTION 5
II. STATEMENT OF THE CASE 9
III. STATEMENT OF FACTS 11
A. The Technology Of The Mirror Worlds Patents 11
1. The '227 Patent 15
2. The '538 and '439 Patents 16
B. Facebook's Accused Computer Systems 16
1. The Timeline Backend System 18
a. The TimelineDB 18
b. The Aggregator Of The Timeline Backend System 19
2. The Multifeed System 20
a. The Leaves 20
b. The Aggregator Of The Multifeed System 21
C. The Prior Appeal In *Mirror Worlds I* 22
D. The District Court Proceedings After Remand 23
III. SUMMARY OF THE ARGUMENTS 26
IV. ARGUMENT 29
A. Standards of Review 29
B. Mirror Worlds Met Its Burden By Presenting Evidence That The Accused Timeline Backend System Includes A "Main Stream" 31
1. The TimelineDB Is A "Main Stream" 31

2. The District Court's Reasons For Concluding That The TimelineDB Is Not A "Main Stream" Are Flawed 34

C. Summary Judgment Was Improper Under A Proper Construction Of The Term "Data Unit" 41

1. A POSITA Would Understand That A "Data Unit" Is Not A Query 42

2. The Information Facebook Asserts Are Not Stored By The Accused "Main Streams" Are Queries And, Thus, Not "Data Units" 45

3. A POSITA Would Understand That A "Data Unit" Must Be Information That Is Of "Direct User Interest" 48

4. The Information Facebook Asserts Are Not Stored By The Accused "Main Streams" Are Not Of Direct User Interest And, Thus, Not "Data Units" 55

D. Summary Judgment Of The '439 Patent Was Improper Under A Proper Construction Of The Term "Main Collection" 57

E. Summary Judgment Of The '439 And '538 Patents Based On The "Glance View" Limitations Was Improper 61

VI. CONCLUSION AND STATEMENT OF RELIEF SOUGHT 62

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION

CONFIDENTIAL MATERIAL OMITTED

Material has been redacted in the Non-Confidential Principal Brief of Plaintiff-Appellant Mirror Worlds Technologies, LLC. The material omitted from pages 18-21, 32, and 62 contain information regarding Facebook's confidential system architecture and source code, which is covered by the terms of the governing protective order entered by the district court.

TABLE OF AUTHORITIES

Cases **Page(s)**

Absolute Software, Inc. v. Stealth Signal, Inc.,
659 F.3d 1121 (Fed. Cir. 2011) 37
AFG Indus., Inc. v. Cardinal IG Co.,
375 F.3d 1367 (Fed. Cir. 2004) 35
Anderson v. Liberty Lobby, Inc.,
477 U.S. 252 (1986) 29
Beatie v. City of New York,
123 F.3d 707 (2d Cir. 1997) 36
Bynoe v. Target Corp.,
548 Fed. Appx. 709 (2d Cir. 2013) 62
Charles Mach. Works, Inc. v. Vermeer Mfg. Co.,
723 F.3d 1376 (Fed. Cir. 2013) 29
Convolve, Inc. v. Compaq Computer Corp.,
527 Fed. Appx. 910 (Fed. Cir. 2013) 36
Convolve, Inc. v. Compaq Computer Corp.,
812 F.3d 1313 (Fed. Cir. 2016) 52
E.I. du Pont de Nemours & Co. v. Unifrax I LLC,
921 F.3d 1060 (Fed. Cir. 2019) 45
Epos Techs. Ltd. v. Pegasus Techs. Ltd.,
766 F.3d 1338 (Fed. Cir. 2014) 30
Ethicon Endo-Surgery, Inc. v. Covidien, Inc.,
796 F.3d 1313 (Fed. Cir. 2015) 40
Gart v. Logitech, Inc.,
254 F.3d 1334 (Fed. Cir. 2001) 45
Google LLC v. Network-1 Techs., Inc.,
2018 WL 1468370 (Fed. Cir. Mar. 26, 2018) 52
IEX Corp. v. Blue Pumpkin Software, Inc.,
122 Fed. Appx. 458 (Fed. Cir. 2005) 62
Innovention Toys, LLC v. MGA Entm't, Inc.,
637 F.3d 1314 (Fed. Cir. 2011) 30
InterDigital Communs., LLC v. ITC,
690 F.3d 1318 (Fed. Cir. 2012) 58
Israel Bio-Eng'g Project v. Amgen Inc.,
401 F.3d 1299 (Fed. Cir. 2005) 33
Karlin Tech., Inc. v. Surgical Dynamics, Inc.,
177 F.3d 968 (Fed. Cir. 1999) 57

Kuebel v. Black & Decker Inc.,
643 F.3d 352 (2d Cir. 2011) 30
Liebel-Flarsheim Co. v. Medrad, Inc.,
358 F.3d 898 (Fed. Cir. 2004) 58
Mirror Worlds Technologies, LLC v. Facebook, Inc.,
800 Fed. Appx. 901 (Fed. Cir. 2020) ..... passim
Mirror Worlds, LLC v. Apple Inc.,
692 F.3d 1351 (Fed. Cir. 2012) 13
PPC Broadband, Inc. v. Corning Optical Comm'ns RF, LLC,
815 F.3d 734 (Fed. Cir. 2016) 52
Stratoflex, Inc. v. Aeroquip Corp.,
713 F.2d 1530 (Fed. Cir. 1983) 35
Teleflex, Inc. v. Ficosa N. Am. Corp.,
299 F.3d 1313 (Fed. Cir. 2002) 30
Tolan v. Cotton,
572 U.S. 650 (2014) 36
Toshiba Corp. v. Imation Corp.,
681 F.3d 1358 (Fed. Cir. 2012) 59
Unwired Planet, LLC v. Apple Inc.,
829 F.3d 1353 (Fed. Cir. 2016) 51, 60
Wi-LAN USA, Inc. v. Ericsson, Inc.,
675 Fed. Appx. 984 (Fed. Cir. 2017) 34

Rules

Fed. R. Civ. P. 56(a) 29

STATEMENT OF RELATED CASES

An appeal from the same civil action in the district court was previously before this Court in *Mirror Worlds Technologies, LLC v. Facebook, Inc.*, Appeal No. 2018-2276, and resulted in a decision by Prost, Chief Judge, and Taranto and Stoll, Circuit Judges, on January 23, 2020, as reported in 800 Fed. Appx. 901, reversing the district court's summary judgment of non-infringement and remanding for further proceedings. This Court also previously addressed one of the asserted patents in an earlier filed case involving a predecessor of plaintiff-appellant Mirror Worlds Technologies, LLC ("Mirror Worlds") and a different defendant in *Mirror Worlds, LLC v. Apple Inc.,* Appeal No. 2011-1392, before Newman, Lourie, and Prost, Circuit Judges, resulting in a September 4, 2012 decision reported in 692 F.3d 1351. Counsel for Mirror Worlds is not aware of any case pending in this Court or any other court or agency that will directly affect or be directly affected by this Court's decision in the pending appeal.

STATEMENT OF JURISDICTION

This is a patent infringement case. Mirror Worlds filed a complaint alleging infringement of U.S. Patent Nos. 6,006,227 ("the '227 patent"); 7,865,538 ("the '538 patent"); and 8,255,439 ("the '439 patent") (collectively, the "asserted patents" or "Mirror Worlds patents"). The district court had jurisdiction pursuant to 28 U.S.C. §§1331 and 1338. Mirror Worlds appeals from the district court's summary judgment, dated March 8, 2022, finding that defendant-cross-appellant Facebook, Inc. ("Facebook")[1] does not infringe the asserted patents, and disposing of all claims and parties in the proceeding below. Mirror Worlds timely filed its notice of appeal on April 4, 2022, and this Court has jurisdiction under 28 U.S.C. §§1291 and 1295.

[1] Although Facebook changed its corporate name to Meta Platforms, Inc., this brief uses the name Facebook consistent with the district court proceedings.

STATEMENT OF THE ISSUES FOR APPEAL

1. Did the district court err in granting summary judgment of non-infringement by finding that the TimelineDB (database) component of the accused Timeline backend system did not store information purportedly received from certain separate systems, and thus could not meet the claimed "main stream," which, in relevant part, was construed to require "every data unit" received by the accused system, when the district court overlooked evidence showing, or at minimum establishing a genuine dispute, that the TimelineDB "persisted" "all the data" and thus is a "main stream," and when other evidence, especially when viewed in favor of the non-movant, showed *no* information sent to the accused system from those other systems?

2. Did the district court err in construing the claim term "data unit" too broadly as simply any "item of information," without any limitation, and granting summary judgment based thereon, when it failed to consider evidence from the intrinsic record distinguishing between "search criteria" and "data units," and explaining that the "main stream" is *not* "affected" by a search, which logically must mean that search criteria are not data units or else they *would* affect the main stream, which includes every data unit, and testimony from Facebook's expert and undisputed person of ordinary skill in the art ("POSITA") agreeing that search criteria does not have to be stored in the main stream?

3. Did the district court also err in broadly construing "data unit" as any "item of information" without requiring the information to be of "direct user interest," and granting summary judgment based thereon, when the intrinsic record repeatedly recites that "all the data units ... are of significance to the user," the omission of that requirement in later covered business method ("CBM") proceedings was not a clear and unmistakable disclaimer because those proceedings were in the context of the broadest reasonable interpretation standard, and even then, the patentee repeatedly emphasized that "data units" "are created by a user application for user consumption, *i.e.*, it is an item of information of direct interest to the user class," and the breadth of the district court's construction would undermine the invention's goal of a "new model and system for managing *personal electronic information*," "a diary of a person['s] ... electronic life," as opposed to storing any information received by a computer system, even, for example, executable computer code or system files that have no place in an electronic diary?

4. Did the district court err in construing "main collection," a requirement of the '439 patent, to mean the same thing as "main stream," required by the other asserted patents, when the '439 patent claim language uses the term "main collection" in its plain and ordinary manner, specifies that the "main collection" need only include "some" documents, in contrast to the "main stream" that includes "every" data unit, and where the patentee's deliberate use of those different terms

("main collection" v. "main stream"), in different patents, to mean different things was clear from the record?

5. Did the district court err in granting summary judgment of non-infringement on the '439 and '538 patents based on the claimed "glance view," by considering only a single source of evidence it found inadmissible, while failing to consider other evidence, including expert testimony and source code, showing, or at minimum establishing a genuine dispute, that the accused systems meet the claimed "glance view"?

I. INTRODUCTION

In reversing the district court in the last appeal, this Court found that "Mirror Worlds, in fact, went further" than simply establishing a genuine dispute regarding Facebook's claim that the accused systems do not meet the "main stream" claim requirement because they supposedly do not store "every data unit" received by the systems. Mirror Worlds "in fact" "identified the accused 'systems,' described (with evidentiary support) how those systems work, and established, based on that description, that the accused systems met the 'every data unit' requirement and hence are 'main streams.'" *Mirror Worlds Technologies, LLC v. Facebook, Inc.*, 800 Fed. Appx. 901, 910 (Fed. Cir. 2020) ("*Mirror Worlds I*").

Nonetheless, on remand, the district court permitted Facebook again to raise the same claim that the accused systems supposedly do not store "every data unit"

received by the systems, and the district court again ruled in Facebook's favor. But the district court's analysis was again flawed. It overlooked evidence presented by Mirror Worlds, faulted Mirror Worlds on grounds that are unsupportable when the overlooked evidence is considered, interpreted other evidence in favor of the movant, reasoning that evidence contradicting Facebook's assertions "doesn't mean" Facebook was wrong, credited a Facebook declaration, created after the close of discovery, specifically for summary judgment, that cited no evidence, over Facebook's contrary technical documents presented by Mirror Worlds, and repeated the same error, corrected by this Court in the last appeal, of relying on information obtained by the unaccused frontend systems, as opposed to information received by the accused backend systems.

The full record, with all inferences drawn in Mirror Worlds' favor, as required, established a genuine dispute of whether the accused Timeline backend system includes a "main stream," which the district court construed, in relevant part, as requiring, "every data unit received or generated by" the accused computer system. As one example, evidence that the district court overlooked disclosed that the TimelineDB – that Mirror Worlds asserted was the "main stream" for the accused Timeline backend system – "has **all** the data," and that "Timeline data is persisted in TimelineDB."[2] Other evidence provided further support for a reasonable jury to

[2] Unless otherwise stated, all emphasis (bold or italics) added by Mirror Worlds.

conclude that the TimelineDB includes "every data unit," and contradicted Facebook's assertion that other unaccused systems send data units to the Timeline backend system that are not stored in the TimelineDB.

Summary judgment was also improper because the district court erred in its construction of the terms "data unit" and "main collection." The patentee made clear that a "data unit" does not encompass search criteria, or information that is not of direct user interest. The district court's construction, however, is nearly limitless as any "item of information." But in reaching that overbroad construction, the district court did not consider the intrinsic evidence distinguishing between a "data unit" and "search criteria," or showing that search criteria do not "affect" the "main stream," and hence cannot be "data units," because otherwise the search criteria would be stored in and "affect" the "main stream," which is inclusive of "every data unit." And, contrary to the district court's conclusion, even in later CBM proceedings the patentee repeatedly emphasized that "all the data units ... are of significance to the user." For instance, it explained that "data unit" is "an item of information of direct user interest to the user class." The invention is directed to a new method and system for managing personal electronic information to provide a diary of a person's electronic life. It is not a system for storing any electronic data, no matter the relevance to the user, that may be received by a computer. Under a proper construction of "data unit" that excludes search criteria and information not

of direct user interest, the record established a genuine dispute as to whether the information Facebook argued was missing from the accused "main streams" were even "data units" needing to be stored.

The district court also erred in construing the term "main collection," in the '439 patent, as synonymous with "main stream" in the other asserted patents. The patentee distinguished "main collection" from "main stream" by, for example, reciting in the claim language that the "main collection" need only include "some" documents, whereas a "main stream" includes "every" data unit, and differentiating a "main collection" from a dependent claim that specifically required the "main collection" in the form of a "stream." Under a proper construction of "main collection," summary judgment of the '439 was precluded. Indeed, Facebook raised no arguments under any construction different from the construction for the term "main stream."

Finally, Mirror Worlds established a genuine dispute of whether the accused systems meet the '439 and '538 "glance view" limitations. The district court focused solely on one category of evidence that it found inadmissible, but again overlooked the full record, including expert testimony and source code presented by Mirror Worlds, establishing, especially when viewed in Mirror Worlds' favor, that the accused systems provide "glance views."

In the prior appeal, Mirror Worlds met its burden of producing sufficient evidence to support a jury verdict that the accused systems met the "every data unit" requirement and hence include "main streams." That Facebook may have gathered additional evidence on remand to support its assertions, merely added to the dispute, but did not erase the evidentiary support in favor of Mirror Worlds. Regrettably, however, the district court on remand repeated errors similar to those addressed in the last appeal, interpreting and weighing evidence in favor of Facebook, instead of Mirror Worlds, overlooking other evidence that contradicted Facebook's assertions, and failing to consider the full intrinsic record on claim construction. The district court's summary judgment was erroneous and should again be reversed.

II. STATEMENT OF THE CASE

This is an appeal from the district court's summary judgment that two accused Facebook computer systems, the Timeline backend system and Multifeed system, do not, as a matter of law, infringe the asserted claims of the Mirror Worlds patents.

The district court's judgment is premised on several findings relevant to this appeal. First, it held that both of the accused systems lack the claimed "main stream" that the court construed as "inclusive of every data unit received or generated by the ... computer system." Appx25, Appx36, Appx48. In the court's view, the evidence was unequivocal that the accused Timeline backend system received information from other systems known as TAO, UDB and Coefficient that are not included in

the TimelineDB, what Mirror Worlds asserted was the "main stream." Appx48-59. And, for the accused Multifeed system, the court found the evidence unequivocal that information from other systems known as Coefficient, Ego and Adfinder were received but not included in the Leaves, what Mirror Worlds asserted was the "main stream" in the Multifeed system. Appx36-45, Appx47.

Second, the district court construed the term "data unit" as "an item of information," Appx30, and under that broad construction, held that the information from TAO, UDB, Coefficient, Ego and Adfinder constituted "data units" that must be included in the TimelineDB and Leaves, as applicable, in order for those components of the accused computer systems to qualify as "main streams," Appx40-41, Appx43-44, Appx56-58. As simply an "item of information," the court held that queries (*e.g.*, search strings, or terms used to search a database) constituted "data units." *Id.* Unlike prior courts, it further held that a "data unit" was not limited to information of "direct user interest." Appx27-29; Appx1249. Both of those further qualifications would have rendered the information from TAO, UDB, Coefficient, Ego and Adfinder irrelevant for purposes of infringement under Mirror Worlds' theory of the case because such information would not constitute a "data unit" needing to be stored in the "main stream." Appx12257-12258, Appx12261-12263.

Third, the court construed the term "main collection," appearing only in the claims of the '439 patent, as synonymous with the term "main stream" in the '227

and ’538 patents, the other two asserted patents. Appx30-34. And, as a result, reached the same conclusions with respect to infringement for the ’439 patent as with the other asserted patents. Appx34.

Finally, the court held that Mirror Worlds failed to present evidence establishing a genuine issue of material fact regarding the claimed “glance view” in the ’439 and ’538 patents. Appx59-63. The court’s determination was premised solely on its exclusion of certain screenshots it found inadmissible. Appx63.[3]

III. STATEMENT OF FACTS

A. The Technology Of The Mirror Worlds Patents

The Mirror Worlds patents describe “a new model and system for managing personal electronic information which uses a time-ordered stream as a storage model and stream filters to organize, locate, summarize and monitor incoming information.” Appx75 (3:61-62); *see also* Appx74 (2:49-58) (“object of the present invention is to provide an operating system in which personal data is widely accessible anywhere”; “in accordance with the present invention users can access their personal document streams from any available platform”).

[3] The district court’s summary judgment also addressed two other issues not relevant to Mirror Worlds’ appeal. It denied summary judgment of patent ineligibility under 35 U.S.C. §101, and denied summary judgment of non-infringement of the Multifeed system with respect to information supposedly received from TAO, Recent Interactions, and Readstate. Appx13-22, Appx45-47.

The "stream" identifies data by time instead of by name and maintains the data in chronological order. Appx75 (4:6-30). This "time-ordered sequence of documents ... functions as a diary of a person or an entity's electronic life." *Id.* (4:6-8); *see also id.* (4:26-28) ("like a diary, a stream records evolving work, correspondence and transactions"); Appx1498 ("time-ordered sequence of documents (data units) that functions as a virtual object (diary)").

The streams consist of data that "are of significance to the user," and can include "any type of data," including "traditional text based files, electronic mail files, binary files, audio data, video data, and multimedia data." Appx75 (4:16-18), Appx80 (14:33-37); Appx1497 ("stream of documents (in the broadest sense) that are of significance to the user and which thus determine the events of direct user interest"; "the stream of documents (data units) of significance to the user"); Appx2017 (explaining that data units are information "created by a user application for the user, i.e., it is an item of information of interest to the user class").

The data, or "pointers" to the data, can be stored "in a variety of ways in terms of physical information storage – for example, physical information storage can be centralized or decentralized .... with multiple servers and/or clients ...." Appx110 (11:33-55). This stream-based model can operate as a standalone operating system, in "client/server architecture running over the Internet," "as a document stream

utility for ... other operating system[s]," or "utilize subsystems from other operating systems ...." Appx75 (4:2-5), Appx80 (14:46-51).

Traditional systems for organizing electronic data sought to manage electronic data in a manner similar to paper documents. In those systems, electronic files were assigned names and extensions and then placed within a hierarchical folder structure of folders within folders, like paper documents organized on a desk or file cabinet. *See* Appx74. This system forced users to remember file names and locations within the folder hierarchy in order to retrieve files. *See id.* The patentee viewed this model as suffering from numerous drawbacks which made it a "poor basis for organizing information" in computer systems. *Id.* As the number of files grew, traditional systems became more and more unwieldy. *See id.* Users were also required to name each file, specify a location in which to store the file, and rely on fixed categories, directories, and subdirectories. *Id*. Automatic archiving, reminder functions, and historical context were also deficient in traditional systems. *Id.*

The inventions described in the Mirror Worlds patents have previously been considered by this Court in *Mirror Worlds I*, and *Mirror Worlds, LLC v. Apple Inc.*, 692 F.3d 1351 (Fed. Cir. 2012) ("*Apple*"). In *Mirror Worlds I*, this Court recognized that the inventions "describe and claim systems and methods for presenting and storing data in time-ordered streams on a computer system." *Mirror Worlds I*, 800 Fed. Appx. at 902. The Court noted that "conventional computers used certain kinds

of hierarchical directories to store and organize data," and "a user created a new document by naming the document and choosing a storage location." *Id.* The Court noted the stated disadvantages with conventional systems: "the operations involve needless overhead; file names are often meaningless to a user; and a user of such a system must remember not just the file name but where the document is stored." *Id.* In contrast, the claimed "stream" is "'a time-ordered sequence of documents that functions as a diary of a person or an entity's electronic life. Every document created and every document sen[t] to a person or entity is stored in a main stream.'" *Id.* And, in *Apple*, the Court noted the specification disclosure that "[b]y constantly keeping track of all the documents on the computer in these chronologically ordered streams and making the location and nature of file storage transparent to the user," the invention sought to "improve[] filing operations and enhance[] the quality of the user's experience." *Apple*, 692 F.3d at 1354.

The patentee commercialized the stream-based system in a software product named "Scopeware." Appx2677-2678. The press called Scopeware ground-breaking, and the lead inventor, a professor of computer science at Yale University, Dr. Gelernter, a visionary. One technology analyst remarked that "[t]o try the Gelernter system is to fall in love with it," describing it as "elegant, natural and beautiful." Appx2679. And the *New York Times* described the patented system as

"offer[ing] a way to break out of the numberless files and folders that clutter computer desktops and make information hard to find." Appx2678.

1. The '227 Patent

The '227 patent is the first issued patent in the Mirror Worlds patent family. Filed on June 28, 1996, and originally assigned to Yale University, the patent issued on December 21, 1999. The patent claims particular systems and methods of organizing data in a computer system using a stream-based model. Apx67. Claim 13 recites:

> 13. A method which organizes each data unit received by or generated by a computer system, comprising the steps of:
>
> generating a main stream of data units and at least one substream, the main stream for receiving each data unit received by or generated by the computer system, and each substream for containing data units only from the main stream;
>
> receiving data units from other computer systems;
>
> generating data units in the computer system;
>
> selecting a timestamp to identify each data unit;
>
> associating each data unit with at least one chronological indicator having the respective timestamp;
>
> including each data unit according to the timestamp in the respective chronological indicator in at least the main stream; and

> maintaining at least the main stream and the substreams as persistent streams.

Appx81. Of the asserted claims of the ’227 patent, only the claim terms “main stream” and “data unit” were directly at issue in the district court judgment.

2. The ’538 and ’439 Patents

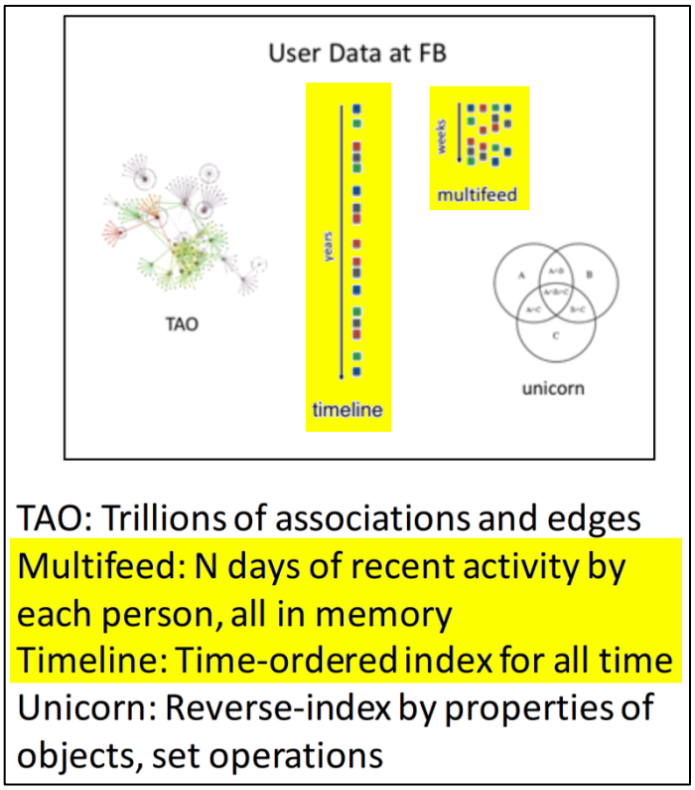
Both the ’538 patent, filed in 2006, and the ’439 patent, filed in 2010, claim priority in-part to the ’227 patent. Appx90; Appx115. Neither of these patents is coextensive in scope with the ’227 patent, and the parties disputed the construction of multiple claim terms unique to specific patents. *E.g.,* Appx1256-1258. For example, the ’439 patent claims recite a “main collection,” whereas the other patents recite a “main stream.” As another example, both the ’538 and ’439 patents recite a “glance view” limitation not present in the ’227 patent. Of the asserted claims in the ’538 and ’439 patents, the terms “main stream,” “main collection,” “data unit,” and “glance view” were directly at issue in the district court judgment. Appx25-34, Appx59-61.

B. Facebook’s Accused Computer Systems

Mirror Worlds accuses Facebook of infringement through two of Facebook’s backend computer systems: (1) the Timeline backend system, which serves Facebook’s Timeline and Activity Log features, and (2) the Multifeed system, which serves Facebook’s News Feed feature. *Mirror Worlds I*, 800 Fed. Appx. at 904. Though users of Facebook generally perceive Facebook as a single system, its

"architecture ... is different from the monolithic service that most ... assume." Appx2694. Facebook is actually "comprised of many different services," and those services are supported by distinct backend systems and infrastructure. *Id.*

Facebook describes and depicts its independent data storage systems as follows, with the accused Timeline backend and Multifeed systems highlighted in yellow:



Appx15245.

As described by Facebook, "the 'News Feed' for a Facebook user displays a variety of items that Facebook has 'deemed to be relevant' to that user. 'Timeline,' ... 'focuse[s] on a particular Facebook user,' showing 'basic information about that user, as well as actions taken on Facebook by or directed toward that user.' And

'Activity Log' provides 'a list of activities that occurred on Facebook that pertain to a particular user.'" *Mirror Worlds I*, 800 Fed. Appx. at 904.

This appeal is only about whether Mirror Worlds presented sufficient evidence to support a jury's finding that the two accused systems (the Timeline backend system and Multifeed) have a "mainstream" for purposes of the '227 and '538 patents, a "main collection" for purposes of the '439 patent, and a "glance view" for purposes of the '538 and '439 patents, and whether genuine issues of fact preclude summary judgment.

1. The Timeline Backend System

The Timeline backend system is comprised of two components: (1) the "TimelineDB," which is a group of networked databases; and (2) the Aggregator, which is software responsible for processing requests to collect data units from the TimelineDB. *See* Appx13613 ("'Timeline backend' is a service and so within that service there's what we call an 'aggregator,' and then there's also a component that is a database, the TimelineDB, and so the two together kind of formed this Timeline backend.").

a. The TimelineDB

The TimelineDB is what Mirror Worlds contends is the "main stream" and "main collection" for the Timeline backend system. As Facebook's technical documents disclose, "Timeline data is persisted in TimelineDB, a fleet of Facebook Architecture

MySQL hosts with master in [redacted] and slaves in [redacted]." Appx14272-14273. The "TimelineDB ... has all the data," Appx14275, "everything relating to [the user]," Appx14532, and stores "all actions by users ... ," Appx14555.

The TimelineDB consists of multiple data tables, in which data units, are recorded and stored. Whenever a user "perform[s] an action (eg, [sic] uploading a photo) [Facebook] ... [w]rite[s] to [the] TimelineDB." Appx14291. Any action taken by a Facebook user causes the "Timeline backend servers to create a row in the Timeline database" that records the action. Appx13661. Facebook calls the primary table within the TimelineDB the "[redacted]" table. Appx13652, Appx13660.

b. The Aggregator Of The Timeline Backend System

The Aggregator of the Timeline backend system retrieves information by querying the TimelineDB. Appx13298. By querying the TimelineDB, the Aggregator "will get access to rows stored in the database. Those rows ... generally represent things a user has done, and that's the result that's ... returned from the db." Appx13628; *see also* Appx14273 ("Aggregator request contains a batch of time-bounded queries and a user summary."); Appx15246 (showing aggregator "get[ting] activities" from the TimelineDB); Appx14281 (showing Aggregator issuing SQL queries to TimelineDB and receiving "Timeline Rows" in response).

The Aggregator aggregates the information collected from the TimelineDB into objects called "[redacted]." Appx14431; Appx13354. The

Source Code are then sent out of the Timeline backend system to a different, unaccused computer system, the frontend system, what Facebook also calls its Web Interface, Webi or PHP layer. Appx14431; Appx13354. Those objects contain information that enable Facebook's frontend system to ultimately render what a user sees in her Timeline or Activity Log. *See* Appx14281, Appx14288-14289. Specifically, they contain "a set of pointers back into content that [the front-end] would then go to TAO to fetch ... the actual data for." Appx13613; *see also* Appx13626; Appx13614.

2. The Multifeed System

The second accused computer system is the Multifeed system, used for Facebook's Newsfeed feature. The "[t]hree main components of multifeed are: the tailer, the leaves, and the aggregator." Appx14303; *see also* Appx13368-13369; Appx14661; Appx13541. The Tailer is used to write all user actions and objects to the Leaves, the Leaves store all the actions and objects, and the Aggregator queries and receives from the Leaves the relevant actions and objects to pass on to a separate frontend system for ultimate rendering as the News Feed feature. *See* Appx14303-14304; Appx13367-13375.

a. The Leaves

The Leaves are what Mirror Worlds contends are the "main stream" in the Multifeed system. The Leaves are "the fast (all-in-RAM), distributed storage layer

that indexes all user actions" that can be displayed in News Feed. Appx14303. "All actions taken in Facebook are recorded in the leaves." *Id.* The "Leaf indexes all users' recent Actions and Objects." Appx14408. "When a user publishes anything on Facebook, we'll go through this pipeline and eventually index them in the multifeed leaf storage system." *Id.*; *see also* Appx14306; Appx14382; Appx13503. The Leaves are the sole source for determining all actions and objects that may appear in News Feed. Appx14665-14666. The "only place [News Feed] can query the multifeed action index [and] the multifeed object index is in the multifeed leaf server." Appx14666.

b. The Aggregator Of The Multifeed System

The Multifeed system's Aggregator "is the query engine that first multi-gets user actions and objects from the storage layer and then applie[s] aggregation to the returned actions and objects to create stories." Appx14303; *see also* Appx13496; Appx14661; Appx14654.

The Aggregator retrieves information from the Leaves, processes them into "Source Code" objects, and sends them to Facebook's separate frontend system to use in generating what is ultimately displayed to the end user. *See* Appx14303; Appx14410-14415. The Source Code objects generated by the Aggregator contain "pointers" which, when sent to the frontend web servers, are used to access

additional relevant data – *e.g.*, images, photographs, and text – for rendering at the user's device. Appx14412-14415; Appx14717.

C. The Prior Appeal In *Mirror Worlds I*

The prior appeal addressed the same principal dispute at issue here: whether there is a genuine dispute that the accused Timeline backend system and Multifeed system include a "main stream" and "main collection." The district court held that there were no genuine disputes and granted summary judgment of non-infringement.

This Court, however, determined that the district court erred in its conclusions. *Mirror Worlds I*, 800 Fed. Appx. at 902, 907-08. The "district court erred in concluding that the record would require a reasonable jury to find that the Aggregators in Multifeed and in the Timeline back-end system receive data from TAO – data that is not included within Leaves or TimelineDB." *Id.* at 907. This Court held that "the record evidence, as presented to the district court and identified to [this Court], does not compel a reasonable jury to find that the Timeline Aggregator receives the asserted data from TAO," or that "the Multifeed Aggregator receives data from TAO that is not in Leaves." *Id.* at 908. This Court noted that not only had Mirror Worlds raised a genuine dispute regarding Facebook's bases for its motion for summary judgment, but "in fact, went further…. [Mirror Worlds] identified the accused 'systems,' it described (with evidentiary support) how those

systems work, and it asserted based on that description that Leaves and TimelineDB meet the 'every data unit' requirement and hence are 'main streams.'" *Id.* at 910.

This Court also rejected Facebook's attempts to raise new non-infringement contentions for the first time on appeal. This Court held that allegations that "certain information ... from Adfinder and Ego for Multifeed ... [and] certain query criteria for the Timeline back-end system," were not only tardy, but also "raise[d] questions about, among other things, whether th[e] information comes within the relevant claim terms, *e.g.*, 'data units,' under a proper construction." *Id.* The district court's judgment was reversed, and the case remanded for further proceedings, including claim construction. *Id.* at 910-11.

D. The District Court Proceedings After Remand

On remand from *Mirror Worlds I*, the district court again permitted Facebook to move for summary judgment on the same issue of whether the accused backend systems include the claimed "main stream" and "main collection" of the asserted patents, and an additional issue of whether the claimed "glance view" of the '538 and '439 patents was met by the accused systems.

In its now second summary judgment motion, Facebook argued: (1) That the Aggregator component of the accused Timeline backend system received certain information from other Facebook systems known as TAO, UDB, and Coefficient, which were not stored in the TimelineDB, and thus the TimelineDB could not meet

the required "main stream" and "main collection" limitations. (2) That the Aggregator component of the accused Multifeed backend system received certain information from Ego, Adfinder, Coefficient, TAO, Recent Interactions, and ReadState, which were not stored in the Leaves, and thus the Leaves could not meet the required "main stream" and "main collection" claim limitations. And (3), that the accused systems do not meet the "glance view" limitations because the relevant features of the systems do not supposedly show "an abbreviated version of the [underlying] document" or are not "indicative of the content thereof." Appx7465-7466.

In response, Mirror Worlds argued that, at minimum, the record established disputed issues of material fact that precluded summary judgment on all asserted bases. Mirror Worlds presented evidence that the TimelineDB and Leaves store all "data units" sent to the Timeline backend and Multifeed systems, respectively. Appx12253-12261. Mirror Worlds also asserted the information Facebook argued was received but not stored in TimelineDB and Leaves were not "data units" because the information was either query information or not of direct user interest, characteristics that fell outside the proper construction of the term "data unit." *Id*. Mirror Worlds also argued that Facebook's non-infringement contention for the '439 patent depended on an improper interpretation of the "main collection" as synonymous with a "main stream," which was inconsistent with the intrinsic record

and the plain and ordinary use of the term "main collection" in the '439 patent. Appx12263. And Mirror Worlds argued that certain source code, expert testimony, and screenshots precluded summary judgment of the "glance view" limitations. Appx12263-12265.

The district court, however, granted summary judgment of non-infringement. Appx65. Notwithstanding this Court's prior finding that Mirror Worlds had "identified the accused 'systems,' ... described (with evidentiary support) how those systems work, and ... asserted based on that description that Leaves and TimelineDB meet the 'every data unit' requirement and hence are 'main streams,'" *Mirror Worlds I*, 800 Fed. Appx. at 910, the district court considered Facebook's evidence "unequivocal," and Mirror Worlds' as either irrelevant or nonexistent. Appx36-45, Appx48-59.

Underlying the district court's findings, were various claim construction rulings. It held that the "main stream" was "inclusive of every data unit received or generated by the alleged computer system." Appx25. That "main collection" was synonymous with a "main stream." Appx34. And that "data unit" means "an item of information." Appx30.

As explained in this appeal, the district court erred in failing to consider all of the evidence presented by Mirror Worlds, weighing the evidence and failing to draw

all reasonable inferences in Mirror Worlds' favor, and for applying erroneous constructions of the terms "data unit" and "main collection."

III. SUMMARY OF THE ARGUMENTS

1. Mirror Worlds presented evidence establishing a genuine dispute of whether the TimelineDB is a "main stream" because it is inclusive of every data unit received or generated by the Timeline backend system. The district court was incorrect in concluding that Mirror Worlds' evidence did not specifically address the TimelineDB. Indeed, the district court overlooked evidence presented by Mirror Worlds that expressly referenced the TimelineDB, and disclosed that "Timeline data is persisted in TimelineDB," and "TimelineDB eventually has all the data." Appx14273, Appx14275. And the district court failed to properly view other evidence in Mirror Worlds' favor; instead, improperly weighing and interpreting the evidence to conclude that "just because" the evidence did not support Facebook, "doesn't mean [what Facebook's asserts] doesn't exist ...." Appx52.

2. The district court's claim construction of "data unit" was flawed in two ways, both of which undermine its summary judgment conclusions. First, it erred in holding that a "data unit" encompasses information in a search query, and failed to even consider the intrinsic record on this issue. *See* Appx40. The patent claims specifically distinguish between "search criteria" and "data unit." Appx89 (2:24-35); *see also* Appx112 (16:20-63); Appx137-138 (16:20-17:47). And the specifications

explain that the "main stream," which includes every data unit, is not affected by the creation or destruction of the results of a search, which necessarily means a search query is not a data unit. Appx75 (4:47-61). If queries were data units, then a query would affect the main stream since it would be included in the main stream. Indeed, Facebook's own claim construction expert agreed that queries did not have to be stored in the "main stream." Appx14215. Because Facebook did not dispute that the relevant information it asserted was received but not stored in the TimelineDB and Leaves were queries, summary judgment under a proper construction of "data unit" was precluded.

3. The district court's construction of "data unit" was also incorrect because all data units must be of "direct user interest." The district court's rejection of this condition was primarily based on an erroneous finding of prosecution disclaimer in post-issuance CBM proceedings. Appx27-29. But nothing in those proceedings gave rise to a clear and unmistakable disclaimer. Those proceedings were under the different, broadest reasonable interpretation standard, and even then, the patentee explained that "data units" are "for the user," and are "created by a user application for the user, *i.e.*, it is an item of information of interest to the user class." Appx2012, Appx2017-2018. In other words, rather than a clear and unmistakable disclaimer, the patentee was instead explaining that a person of ordinary skill in the art would understand that a "data unit" inherently is information of interest to the

user. And that characterization was consistent not only in the CBM proceedings, but throughout the prosecution history. As the patentee explained during the original prosecution of the ’227 patent, “all the data units ... are of significance to the user.” Appx1498. Without that qualification, the district court’s overly broad construction, extending to any and all information a computer may receive or generate, no matter the relevance to the user, undermines the goal of the invention to manage personal data, and act as a diary of a person’s electronic life. Appx74 (2:49-58), Appx75 (3:61-62, 4:6-8, 4:26-28); Appx1498. Moreover, Mirror Worlds presented evidence establishing at least a genuine dispute of material fact as to whether the information Facebook alleged were not stored in the accused “main streams” were “data units” under a proper construction of the term that excluded information not of “direct user interest.” Appx12258, Appx12262-12263.

4. The district court’s construction of the term “main collection” in the ’439 patent as the same as “main stream” in the ’227 and ’538 patents was incorrect. The ’439 patent claims use the term “main collection” in its plain and ordinary manner, and without the conditions required of the “main stream” in the ’227 and ’538 patents. Appx137-138 (16:20-17:47). Critically, the ’439 patent claims specify that the “main collection” need only include “some” documents, in contrast to the “every data unit” requirement of a “main stream.” Appx137 (16:29-30). In addition, the district court failed to consider that only in a dependent claim of the ‘439 patent

does the "main collection" take the form of a "stream." Appx30-34; Appx139 (20:43-46).

5. Mirror Worlds presented evidence establishing a genuine dispute of whether the accused systems meet the "glance view" limitations of the '538 and '439 patents. The district court's conclusion was based on one source of evidence it deemed inadmissible, but failed to consider additional evidence presented by Mirror Worlds, including expert testimony and Facebook source code. Appx63; Appx12265. Thus, the district court erroneously failed to recognize a genuine issue of material fact precluding summary judgment.

IV. ARGUMENT

A. Standards of Review

A district court's grant of summary judgment can only be affirmed when the moving party has shown "that there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law." Fed. R. Civ. P. 56(a). "[S]ummary judgment will not lie if the dispute about a material fact is 'genuine,' that is, if the evidence is such that a reasonable jury could return a verdict for the nonmoving party." *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 252, 249 (1986). In ruling on a motion for summary judgment, "[t]he evidence of the nonmovant is to be believed, and all justifiable inferences are to be drawn in his favor." *Id.* at 255.

This Court "review[s] a grant of summary judgment under the law of the regional circuit." *Charles Mach. Works, Inc. v. Vermeer Mfg. Co.*, 723 F.3d 1376, 1378 (Fed. Cir. 2013). The Second Circuit reviews a grant of summary judgment *de novo* and without deference, "construing the evidence in the light most favorable to the nonmovant and drawing all reasonable inferences in that party's favor." *Kuebel v. Black & Decker Inc.*, 643 F.3d 352, 358 (2d Cir. 2011).

Determining infringement is a two-step process: first, performing claim construction and, second, applying the claim construction to the facts of the case. *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1323 (Fed. Cir. 2002). The "first step, claim construction, is a question of law, reviewed de novo." *Innovention Toys, LLC v. MGA Entm't, Inc.*, 637 F.3d 1314, 1319 (Fed. Cir. 2011). The second step, "[i]nfringement[,] is a question of fact." *Epos Techs. Ltd. v. Pegasus Techs. Ltd.*, 766 F.3d 1338, 1342 (Fed. Cir. 2014).

On appeal from a summary judgment of non-infringement, this Court "must determine whether, after resolving reasonable factual inferences in favor of the patentee, the district court correctly concluded that no reasonable jury could find infringement." *Id.*

B. Mirror Worlds Met Its Burden By Presenting Evidence That The Accused Timeline Backend System Includes A "Main Stream"

The district court erred in concluding that Facebook presented "unequivocal evidence" that the Timeline backend system receives data units that are not stored in the TimelineDB. The district court improperly weighed the evidence, crediting declarations from a Facebook engineer and a Facebook expert, while discrediting contrary evidence and expert testimony presented by Mirror Worlds. The district also improperly interpreted the evidence in favor of the movant, as opposed to the non-movant. But contrary to the district court's characterizations, and especially when all inferences are drawn in favor of the non-movant, the evidence presented by Mirror Worlds supports a finding that the TimelineDB stores every data unit received or generated by the Timeline backend system, and contradicts Facebook's assertion that other information is received but not stored. Under the proper standard of review for summary judgment, the evidence presented by Mirror Worlds raised a genuine dispute of material fact that precluded summary judgment.

1. The TimelineDB Is A "Main Stream"

The evidence presented by Mirror Worlds supports a finding that all data units received or generated by the Timeline backend system are stored in the TimelineDB. Facebook's own internal documents specifically refer to the TimelineDB as persisting all data for Timeline. An internal Facebook document, for instance,

specifically describes the "architecture" of the Timeline backend system, including the "storage – TimelineDB" of the "Timeline backend," and states that "**Timeline data is persisted in TimelineDB**, a fleet of [redacted] MySQL hosts with master in [redacted] and slaves in [redacted]." Appx14273. The same document also describes the process of storing data "**to make sure TimelineDB eventually has all the data** ...." Appx14275.

Similarly, in a Facebook presentation entitled "Timeline In A Nutshell," Facebook explains that "when you **perform an action** (eg, uploading a photo) **we**: ... **Write to TimelineDB** ...." Appx14291.

Another Facebook document describes the "backend" "Timeline Infrastructure" as a "time-ordered index for all time" of "user data." Appx15241-15245. The document explains that "building timeline involved backfilling .... minutiae, new friendships, photos, life events," and that the TimelineDB is used to "persist activities," and involves [redacted] updates per minute and [redacted] inserts per minute. Appx15244-15246.

Likewise, a Facebook "guide [that] explains the architecture of Timeline Backend," states that "Timeline is a backend system that persists all actions by users ...." Appx14554-14555.

And another Facebook document explains that "[t]he centrality of timeline is important ... – **everything relating to you is here** and you can review it." Appx14532.

In addition, expert testimony from Dr. Eric Koskinen, submitted by Mirror Worlds, asserted that the "TimelineDB of the Timeline infrastructure is a 'main stream,'" and that this "mainstream of the Timeline Backend System receives each data unit received by or generated by the computer system." Appx13304, Appx13306; *see also* Appx13309-13310. Dr. Koskinen reviewed Facebook's documents, testimony, and source code to explain that "all actions and objects that are used to generate the visually-rendered information in the Timeline service ... are located in the TimelineDB." Appx13291-13292. Dr. Koskinen explained that a single user's actions "are written to one machine and one ... instance of the" TimelineDB, and that "[i]n technical terms, the TimelineDB is a 'sharded' database, and **any given Facebook user's Timeline-related data is stored on a specific shard of that database**." Appx13294. Dr. Koskinen further explained that whenever "a Facebook user performs some action," specific source code is executed with "the goal of the process ... to write data reflecting [the] user's action into storage in the backend." Appx13296.

The evidence presented by Mirror Worlds, at minimum, raises a genuine dispute of material fact as to whether the TimelineDB contains all data units received

or generated by the Timeline backend system. Especially when all reasonable inferences are drawn in favor of Mirror Worlds, the evidence supported a finding that **all** Timeline data is persisted, in other words, stored, in the TimelineDB. The district court's failure to properly credit Mirror Worlds' evidence requires reversal. *See, e.g., Israel Bio-Eng'g Project v. Amgen Inc.*, 401 F.3d 1299, 1304 (Fed. Cir. 2005) (reversing summary judgment after finding genuine issues of material fact because "the district court failed to analyze evidence supporting [patentee's] claim"); *Wi-LAN USA, Inc. v. Ericsson, Inc.*, 675 Fed. Appx. 984, 995 (Fed. Cir. 2017) (reversing summary judgment where "District Court ... ignored conflicting evidence in the record and placed greater weight on [the movant's] evidence").

2. The District Court's Reasons For Concluding That The TimelineDB Is Not A "Main Stream" Are Flawed

The district court rejected Mirror Worlds' evidence, but did so on grounds that should be reversed on appeal. First, the district court held that the evidence "actually refer[s] to 'Timeline' (the backend system writ large), not the TimelineDB. Counsel for Mirror Worlds and Mirror Worlds' expert, Dr. Eric Koskinen, simply changed the subjects of the ... sentences." Appx50; *see also* Appx53 ("Dr. Koskinen has simply changed the subject of the quoted sentences from 'Timeline' to 'the TimelineDB'"). The district court, however, was incorrect. Some of the evidence did, in fact, refer to the TimelineDB specifically. And, while some other documents did not expressly refer to the TimelineDB, the context of those documents and the

full record, combined with the standard for summary judgment, should have led to an interpretation that favored Mirror Worlds.

For instance, the district court ignored Appx14272-14277 and Appx14279-14299, both of which expressly reference and describe the TimelineDB. Appx14272-14277 identifies the "storage – TimelineDB" of the "Timeline backend" and explains that "**Timeline data is persisted in TimelineDB,**" "**to make sure TimelineDB eventually has all the data** ...." Appx14273, Appx14275. Likewise, Appx14279-14299 expressly refers to the TimelineDB and explains that "when you **perform an action** ... **we** ... **Write to TimelineDB** ...." Appx14291. The district court's rationale is inconsistent with these Facebook documents that specifically identify the TimelineDB as having "all the data" for Timeline. And it is "jurisprudentially inappropriate to disregard any relevant evidence on any issue in any case, patent cases included." *Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530, 1538 (Fed. Cir. 1983).

Moreover, even though other documents, or even just portions of those documents, may not have expressly referred to the TimelineDB, the district court should have interpreted those documents in Mirror Worlds' favor. The documents describe Timeline as having "**everything relating to you,**" Appx14532, and more specifically, to the backend as "persist[ing] all actions by users," Appx14554-14555, and a "time-ordered index for all time" of "user data," that includes, for example,

"minutiae, new friendships, photos, life events," etc., Appx15241-15245. That some of these statements did not specifically identify the TimelineDB was not cause for the district court to disregard them, especially where the documents refer to the "backend" and the the TimelineDB is undeniably the DB (database) for the backend. *See AFG Indus., Inc. v. Cardinal IG Co.*, 375 F.3d 1367, 1373 (Fed. Cir. 2004) (reversing summary judgment based on internal document from accused infringer even though "not conclusive as to the structure of the accused products" because "a fact-finder may consider it relevant in determining whether … [the accused] products meet the limitations of the … claim"); *Convolve, Inc. v. Compaq Computer Corp.*, 527 Fed. Appx. 910, 928 (Fed. Cir. 2013) (disagreeing with district court's rejection of evidence as supposedly "too generic" as failing to "draw[] all reasonable inferences in the non-movant's favor"). And Mirror Worlds presented an expert interpretation of the import of those documents. Appx13304, Appx13310, Appx13291-13296. The district court was not free to simply substitute its own interpretation of them (which was not in the light most favorable to the non-movant).

Second, the district court dismissed Mirror Worlds' evidence showing an absence of interaction between TAO, UDB or Coefficient and the Timeline Aggregator on the grounds that "just because it's not there [(*i.e.*, shown in the document)] doesn't mean it doesn't exist ...." Appx52. But that conclusion is based on reasoning this Court has already rejected, an impermissible weighing of the

evidence, and a failure to draw all reasonable inferences in Mirror Worlds' favor. *See Mirror Worlds I*, 800 Fed. Appx. at 908-909; *Tolan v. Cotton,* 572 U.S. 650, 656 (2014) ("'judge's function' at summary judgment is not 'to weigh the evidence and determine the truth of the matter but to determine whether there is a genuine issue for trial'"); *Beatie v. City of New York*, 123 F.3d 707, 710 (2d Cir. 1997) ("[w]hen deciding a summary judgment motion, a trial court's function is not to weigh the evidence, make credibility determinations or resolve issues of fact"); *Absolute Software, Inc. v. Stealth Signal, Inc.*, 659 F.3d 1121, 1133 (Fed. Cir. 2011) (reversing summary judgment where district court "fail[ed] to draw a reasonable inference in favor of [patentee]").

For instance, the mistake addressed by this Court's prior decision resurfaced when the district court held that "[i]t is undisputed that TAO exists, and the record is clear that the Timeline ... feature[] could not function without TAO." Appx52. In reaching that conclusion, the district court relied on testimony from Facebook's director for Timeline, Jeffrey Huang, that "in any form of **rendering** Timeline ... you have to have TAO." *Id.* (quoting Appx13638 (128:15-17)). But nearly identical testimony was considered and previously rejected by this Court as "not show[ing] that the Timeline Aggregator is the element of Facebook's infrastructure that receives the referred-to content from TAO. Mr. Huang's testimony is consistent with the theory that front-end systems receive that content without the content having first

gone through the accused (back-end) system. Indeed, other testimony by Mr. Huang suggests that the **front-end web layer retrieves that content**" "'to actually **show** anything on Timeline' ...." *Mirror Worlds I*, 800 Fed. Appx. at 909.

And the district court impermissibly weighed the evidence, and failed to draw all reasonable inferences in favor of the non-movant. Documents and testimony presented by Mirror Worlds supported a reasonable inference that information from TAO, UDB and Coefficient are *not* received by the Timeline backend system at all, and thus need not be stored in the TimelineDB for it to be a "main stream."

For example, the Facebook "Architecture" document described *supra*, depicts Facebook's "frontend running Timeline queries." Appx14273. But nowhere in that depiction of the Timeline architecture is there any indication that information from TAO, UDB or Coefficient is sent to the Aggregator as part of the query from the frontend (also referred to as "webi"). *Id.*

In another Facebook presentation on the "Timeline Infrastructure," the Timeline "Architecture" is again depicted without any reference to the frontend (Webi) request to the Aggregator containing any information from TAO, UDB or Coefficient. Appx14542; *see also* Appx14301.

Indeed, the "Timeline In A Nutshell" document described *supra* depicts the "request flow" in a Timeline request from the frontend to the Aggregator as including only "User, date, [and] params" information, without any reference to

information from TAO, UDB or Coefficient. Appx14281. In further describing the contents of the request from the frontend to the Aggregator, the document refers only to the "user, start date, end date," or "start time[,] end time[,] user id[,] type of units ... [, and] other parameters for special cases," but never to any information from TAO, UDB or Coefficient. Appx14282, Appx14285.

Facebook's director for Timeline similarly identified only the "user id, a start time, and end time" as inputs in the frontend request to the Aggregator. Appx13642-13642 (140:13-143:20). Mr. Huang's description of the source code similarly made no reference to any fields from TAO, UDB or Coefficient as part of the request to the Aggregator. Appx13664-13665 (230:6-234:9). And in reviewing a Facebook architecture document, he acknowledged that there was no indication that the Aggregator received any information from UDB. Appx13626 (78:18-22). Individually, and collectively, these documents supported a reasonable inference contradicting Facebook's assertion and the district court's conclusion that information from TAO, UDB or Coefficient is sent to the Aggregator but not stored in the TimelineDB.

Finally, although unnecessary on a motion for summary judgment, the supposedly "unequivocal" evidence favored by the district court hardly justified dismissing the evidence presented by Mirror Worlds, which included Facebook's own documents and employee and expert testimony. In total, the district court relied

on three paragraphs from a Facebook employee's (Yifei Tang) declaration submitted in support of Facebook's motion, and a dozen paragraphs from a report from Facebook's expert. Appx49, Appx57-58 (*citing* Appx8156-8158 (¶¶10-12); Appx8254, Appx8258-8260, Appx8286-8287, Appx8290-8293, Appx8300 (¶¶246, 254, 256-57, 300-301, 307-309, 313, 317, 337)). But the Tang declaration is conclusory, refers to no documents (or other corroboration) to support the assertions, addresses none of the evidence presented by Mirror Worlds, and is an after-the-fact, self-serving characterization of Timeline, created after discovery closed and specifically for litigation by an interested witness. Appx8156-8158 (¶¶10-12); *see also Ethicon Endo-Surgery, Inc. v. Covidien, Inc.*, 796 F.3d 1313, 1326 (Fed. Cir. 2015) (reversing summary judgment as "based on an incomplete view of the record," and because "self-serving testimony from [movant's employee] ... witnesses ... does not negate the evidence in the record ... supporting" possible infringement).

The paragraphs from Facebook's expert also are by no means dispositive. Paragraph 246 simply relapses to the irrelevant and rejected theory of data necessary to "render" a Timeline that this Court addressed in reversing the district court's prior summary judgment order. Appx8254; *Mirror Worlds I*, 800 Fed. Appx. at 909 (considering same testimony from Mr. Huang, which is similar to testimony from Mr. Tang, concerning data for "rendering" Timeline). Paragraphs 254, 256-257, 300-301, and 317 simply quote Mr. Tang's unsupported testimony. Appx8258-8260,

Appx8286-8287, Appx8293. Paragraphs 307-309, and 337 just refer to certain data fields in a query to, and response from, the Timeline backend, but do not provide any evidence showing any data coming from TAO, UDB or Coefficient. Appx8290-8292, Appx8300. And paragraph 313 addresses how Coefficient data is used, but says nothing about the Timeline backend receiving or using that data. Appx8292. The record, at minimum, demonstrated a disputed issue of material fact that should have precluded summary judgment.[4]

C. Summary Judgment Was Improper Under A Proper Construction Of The Term "Data Unit"

While Mirror Worlds presented evidence disputing that information was received by the accused backend systems and not stored in the accused "main streams," summary judgment was improper for the independent reason that the district court's construction of the term "data unit" was incorrect. Under a proper construction of the term "data unit," the district court should have denied summary judgment because, even assuming that Facebook established that certain information was received by the accused backend systems but not stored in the asserted "main streams," such information fell outside of the proper scope of the term "data unit."

[4] The district court also faulted Mirror Worlds for presenting evidence that TAO is not a data store, finding it immaterial to infringement. Appx51. But Mirror Worlds did not rely on that evidence to directly address infringement. Rather, it showed Facebook's own employees were equivocal and contradictory about TAO and what may or may not come from TAO, and demonstrated an additional reason to doubt Facebook's assertions. Appx12256.

The district court's claim construction resulted in an erroneous finding of summary judgment in two ways. First, the district court's claim construction encompassed information in a query or request even though the intrinsic record reveals that the patentee did not consider queries a "data unit." The district court, however, conducted no analysis of the intrinsic record on this issue. Under a proper construction of the term, the district court should have found that even if information from TAO, UDB and Coefficient was received by the accused Timeline backend system, and information from Coefficient and Ego was received by the Multifeed backend system, such information was only received in the form of a search query and thus not a "data unit."

Second, the district court's construction also improperly included information not of direct user interest. The intrinsic record, however, makes clear that the term "data unit" is limited to information that is of direct user interest, and not information from TAO, UDB, Coefficient, Ego, or advertisements, and nothing in the record amounts to a clear and unmistakable disclaimer of the user interest requirement.

1. A POSITA Would Understand That A "Data Unit" Is Not A Query

The meaning of the term "data unit" does not extend to information received in a query. In construing the term simply as "an item of information," the district court's construction was overly broad and inconsistent with the scope of the term in light of the intrinsic record. Indeed, the district court failed to consider evidence from

the patent claims and specifications distinguishing search criteria from data units, and demonstrating that a search cannot be a data unit because a main stream, which includes every data unit, is not "affect[ed]" by a search. Appx75 (4:47-61). The district court asked only whether a query is "an item of information," and not whether more was required for information to constitute a "data unit." Appx40. As a result, its holding that "[a] query is plainly 'an item of information,'" *id.*, and thus within the construction of the term "data unit," was flawed and should be reversed.[5]

The intrinsic record confirms that the term "data unit" does not encompass a query. The patents distinguish between a search query and data unit. A search query specifies criteria used to identify relevant data units from the main stream, but is not itself a data unit. Claim 55 of the '227 patent, for example, differentiates between "data units" and "search criteria" in describing the step of "identifying a set of *data units* from said main stream or at least one substream based on *search criteria* provided by a user ...." Appx89 (2:24-35); *see also* Appx112 (16:20-63); Appx137-138 (16:20-17:47). The '227 patent specification further explains this difference,

[5] The district court should have considered whether the intrinsic record supports including queries in the scope of "data unit." Mirror Worlds raised the issue and referred to the intrinsic record and Facebook's claim construction expert's acknowledgement that queries are not data units. Appx12257-12258. And this Court acknowledged that "query criteria" and advertising information "raise questions about, among other things, whether [such] ... information comes within the relevant claim terms, *e.g.*, 'data units,' under a proper construction." *Mirror Worlds I*, 800 Fed. Appx. at 910.

describing the use of a "search query" to search the main stream, resulting in the creation of a substream of "documents that are relevant to the search query." Appx75 (4:49-56); *see also* Appx76 (6:61-64). In other words, a search query is not a data unit, but, rather, is a means for identifying data units.

Importantly, the specification explains that the resulting substream "can be created and destroyed on the fly without affecting the main stream or other substreams." Appx75 (4:59-61). If a query constituted a data unit, however, a substream created by a search query could not be created or destroyed "without affecting the main stream" since the query necessarily would be included in the main stream, which the parties and district court agreed, must include all data units received or generated by the computer system. In other words, every query would itself constitute a new data unit that would necessarily become part of the main stream, and thus "affect" the main stream, contrary to the specification's explanation that the main stream is *not* affected.

Facebook's own claim construction expert, an undisputed POSITA, agreed that the term "data unit" does not encompass queries. Facebook's expert, Stephen Gray, admitted that "[s]toring search queries is not required to have a main stream." Appx14215 (129:14-16); *id.* (128:4-25) ("Q. ... In other words, storing search queries are not required to have a main stream; agreed? A. ... I don't think so."). His

response only makes sense if search queries are not data units because, otherwise, a search query, as a data unit, would need to be included in the main stream.

The district court, however, did not consider the intrinsic record, or Mr. Gray's opinion, on the question of whether a query constitutes a data unit. Appx40, Appx56-58. Instead, having construed the term "data unit" as simply "an item of information," the district court conducted no further consideration of the issue, and summarily concluded that the information identified by Facebook constituted "data units." This was error. *See E.I. du Pont de Nemours & Co. v. Unifrax I LLC*, 921 F.3d 1060, 1069 (Fed. Cir. 2019) ("We must consider whether the context of the entire patents and other intrinsic evidence supports such an interpretation."); *Gart v. Logitech, Inc.,* 254 F.3d 1334, 1343–44 (Fed. Cir. 2001) (holding that district court "summarily conclud[ing]" accused products did not infringe was improper as "premised on an erroneous construction"). The district court should have looked beyond simply a "data unit" as an "item of information," and considered whether the patentee had further delineated the scope of the term.

2. The Information Facebook Asserts Are Not Stored By The Accused "Main Streams" Are Queries And, Thus, Not "Data Units"

Under a proper construction of "data unit" that excludes queries, summary judgment should have been denied. It was undisputed that the information Facebook asserted was not stored in the accused "main streams" were queries, and thus not

"data units." Indeed, Facebook did not dispute that the information was only included in queries, arguing instead only that queries were "data units." Appx18002. Nor did the district court fairly address this issue, ignoring the issue completely for Timeline, *see* Appx56-58, and for Multifeed, ignoring the issue in the context of Ego, *see* Appx44, and overlooking the evidence presented by Mirror Worlds in the context of Coefficient, *see* Appx40. Instead, the district court simply concluded that the information identified by Facebook were items of information and thus "data units," as opposed to considering whether the information were queries.

For the Timeline backend system, Mirror Worlds identified evidence establishing that the information from TAO, UDB and Coefficient that Facebook argued were not stored in the TimelineDB are part of a query. For instance, Facebook's fact declarant for Timeline, Mr. Tang, asserted that "the 'front end' ... constructs a **request** to the Timeline Aggregator that supplies the Aggregator with a number of pieces of information ... including [the background user information] .... obtained from 'TAO' and 'UDB' ...." Appx8156-8157 (¶10). He further explained that "the web tier will ... grab the necessary data and construct **a Timeline request**, which includes .... all ... **major life events** we get from TAO, the **coefficient** ... and a bunch of other necessary data .... [a]nd this request will be sent to Timeline Aggregator ... [to] be used as input to **query** Timeline DB in some cases, when necessary." Appx14161 (77:6-25); *see also* Appx14162 (81:18-24) (indicating that

"ranking information," *i.e.*, coefficient data, is "got from the Timeline request"; "major life events ... also passed to aggregator through the Timeline request"). Similarly, Facebook's expert, Ravin Balakrishnan, stated in his report that the information from TAO and Coefficient "are provided to the Timeline Aggregator to construct requests ...." Appx8286 (¶300).

Likewise, for the Multifeed system, Mirror Worlds identified evidence establishing that the information from Coefficient and Ego that Facebook argued is not stored in the Leaves is part of a query. For instance, although the district court faulted Mirror Worlds for presenting "no evidence that coefficient scores should be considered 'queries,'" Appx40, Mirror Worlds pointed to the same Coefficient-related evidence addressed in the context of the Timeline backend system. Appx12262 (cross-referencing Section IV.B.1 [6] of Mirror Worlds' brief in opposition to summary judgment). And for Ego, Mirror Worlds identified testimony from Facebook's News Feed engineer, Xingdi Li, asserting that, in the front-end web layer, the "multifeed inquiry cause[s] the ego services to get ... ego objects, and those objects ... mov[e] ... through the multifeed **request** into aggregator ...." Appx14001 (37:10-15). Similarly, Facebook's expert, Dr. Balakrishnan, acknowledged in his

[6] Although Mirror Worlds inadvertently cited Section IV.B.1 instead of the correct Section IV.B.2 in its brief, the typographical error was obvious because Section B.1 addressed the substance of whether information was received but not stored in the Leaves, while Section B.2 expressly addressed the issue of what information constitutes a data unit. Appx12254-12258.

report that the information from Coefficient and Ego are part of requests to the Aggregator. Appx8230 (¶205) (depicting the front-end web layer receiving information from Coefficient and sending a "Read Request" to the Aggregators), Appx8248-8249 (¶236) (referring to the same testimony from Mr. Li regarding Ego as part of the "multifeed request into [the] aggregator").

Because it was undisputed that the TAO, UDB, Coefficient and Ego information is only received as part of a query, summary judgment should have been denied under a proper construction of "data unit" that excluded queries.

3. A POSITA Would Understand That A "Data Unit" Must Be Information That Is Of "Direct User Interest"

The district court's construction of the term "data unit" was also incorrect because it was not limited to information that is of "direct user interest." The district court rejected that further limitation, reasoning that it was not supported by the intrinsic record and was disclaimed during post-issuance CBM proceedings. Appx27-30. However, the district court's analysis failed to consider all of the relevant intrinsic evidence, was inconsistent with claim construction principles, was based on an erroneous finding of prosecution disclaimer, and resulted in an erroneous construction that conflicts with the purpose of the invention.

A review of the intrinsic record demonstrates that, while the meaning of "data unit" is broad, it is not limitless, and the information must be of direct user interest. The patentee described the invention as limited to data units "of direct user interest."

In response to the patent examiner's "concern about the breadth of coverage of the claims," they explained that "all the data units ... are of significance to the user." Appx1497-1498. In the same response, they repeatedly emphasized that data units are focused on user interest: "stream of documents (in the broadest sense) that are of significance to the user and which thus determine the events of direct user interest"; "the stream of documents (data units) of significance to the user"; and "substreams allow a user to determine the events of direct user interest from the stream of data units of significance to the user (main stream)." *Id*. They further explained, as memorialized in the patent examiner's interview summary, that data units are of significance to the user if they are "of direct user interest[.]" Appx18008.

The patentee reiterated again during reexamination of the '227 patent that data units were those of direct user interest. They explained that the "main stream includes each data unit that is of significance to the user," Appx18025, and referred to the "stream of each data unit, or document, received by or generated by the computer system that may be of interest to the user," Appx18037-18038.

Indeed, the requirement of "direct user interest" flows from one of the principal goals of the invention. The patentee sought to provide a "new model and system for managing personal electronic information." Appx75 (3:61-62); *see also* Appx74 (2:49-58) ("an operating system in which personal data is widely accessible anywhere"; "users can access their personal document streams from any available

platform"). They described the "present invention [a]s a time-ordered sequence of documents that functions as a diary of a person or an entity's electronic life." Appx75 (4:6-8); *see also id.* (4:26-28) ("like a diary, a stream records evolving work, correspondence and transactions"); Appx1498 ("time-ordered sequence of documents (data units) that functions as a virtual object (diary)"). And they explained that, while "a data unit 'can contain any type of data,'" it still must be "related to 'an entity's electronic life.'" Appx1507. The streams of the invention reflect this focus on users by limiting data units to those of direct user interest.

Although the district court acknowledged the focus on "information that is interesting to the user," it considered it a "goal of the invention writ large" that "cannot be crammed into the term 'data unit.'" Appx27. But the district court looked too narrowly at just the display aspects of the invention and failed to take into account the repeated and express statements by the patentee that data units, irrespective of the display of data units, "are of significance to the user." *E.g.,* Appx1497-1498; *see also* Appx18008 ("of direct user interest"); Appx18025 ("of significance to the user"); Appx18037-18038 ("of interest to the user"). Nor is there any logical basis, on this record, to conclude that the scope of the term "data unit" and the display aspects of the patents are mutually exclusive with respect to what the district court considered the "goal of the invention."

Moreover, the district court failed to appreciate the conflict between its construction and the claimed system of managing personal electronic information in the asserted patents. Appx75 (3:62-65). By not limiting the scope of "data unit," the district court relegated the invention to a storage system of any and all information, regardless of its relevance to the user. But the patents make clear that the streams of data units amount to a diary of a person's electronic life, and by helping users to "organize, locate, summarize and monitor incoming information," *id.* (3:64-65), necessarily distinguishes between information of interest to the user, and other information that may be received or generated by a computer system. Indeed, Mirror Worlds presented expert testimony, from Dr. Koskinen, explaining that POSITA differentiate between "system data [that] exists in support of the overall [computer] operating system, in support of applications, etc.," and data meant for a user, such as "files created by the user ... , files that result from user action, ... [and] files created by a group of users or ... resulting of actions by a group of users." Appx1676-1677 (¶¶13-18). Dr. Koskinen further explained that "[c]omputer systems ... distinguish data that is of direct user interest by making them accessible to said user." Appx1677 (¶17). The district court's construction, however, does not account for the differences between information of interest to the user and system data, for instance, and as a result, yields a system that would inundate the user with information meaningless to the user, negating the benefit of claimed "document stream operating

system," Appx67, and rendering impossible the objective to provide "a diary of a person['s] ... electronic life." Additional restriction is clearly required to identify which "items of information" generated or received should be in the main stream.

Nor did the patentee clearly and unmistakably disclaim the user interest requirement during later CBM proceedings, as found by the district court. *E.g., Unwired Planet, LLC v. Apple Inc.*, 829 F.3d 1353, 1358 (Fed. Cir. 2016) ("disclaimer or disavowal of claim scope must be clear and unmistakable, requiring 'words or expressions of manifest exclusion or restriction' in the intrinsic record"). Rather than a disclaimer, the patentee's statements show that a POSITA would understand "data unit" to mean that the information be of interest to the user.

The district court's finding of disclaimer was flawed for several reasons. First, the CBM proceedings applied the broadest reasonable interpretation standard, and not the *Phillips* standard applied by Federal courts. *See PPC Broadband, Inc. v. Corning Optical Comm'ns RF, LLC*, 815 F.3d 734, 740-42 (Fed. Cir. 2016) ("[t]his case hinges on the claim construction standard applied"; "the claim construction standard is outcome determinative"); *Google LLC v. Network-1 Techs., Inc.*, 2018 WL 1468370, *5 (Fed. Cir. Mar. 26, 2018) (acknowledging that a BRI construction might not even be the "correct construction under the framework laid out in *Phillips*"); *Convolve, Inc. v. Compaq Computer Corp.*, 812 F.3d 1313, 1325 (Fed.

Cir. 2016) ("err[or]" to fail to "account[] for the differences between the [BRI] standard and Phillips").

Second, although the district court disagreed, the patentee in the CBM proceedings made clear that its arguments were in the context of the BRI standard. The patentee, for instance, asserted that the CBM petition "fail[ed] to address the broadest reasonable construction of 'data unit' ...." Appx1946. And throughout the entirety of its claim construction response, it repeated that it was applying the "broadest reasonable construction" of the term "data unit" at least fifteen times. *Id.*, Appx2004-2006, Appx2010, Appx2014-2015, Appx2023. Even the Board acknowledged that the patentee was "urg[ing] us to apply the broadest reasonable interpretation of 'data unit,'" ultimately agreeing with the patentee's broadest reasonable interpretation. Appx2037-2038. Interpreting a term under the BRI standard, while expressly distinguishing that interpretation from a district court's "different legal standard," *e.g.,* Appx2014-2015, cannot be a "clear and unmistakable" disclaimer under the *Phillips* standard.

Finally, even in the context of the broadest reasonable interpretation, the patentee explained that "data units" are nevertheless "for the user." Appx2012. The patentee explained that a data unit is information that "is created by a user application for the user, *i.e.*, it is an item of information of interest to the user class." Appx2017. The patentee explained that "whether something is a 'data unit,' or

'document' ... turns on whether it is an item of information for users generally." Appx2017-2018; *see also* Appx2023 ("unifying property [between 'data unit' and 'document' is] that they are created by a user application for user consumption, i.e., it is an item of information of direct interest to the user class"). And the patentee explained that these "item[s] of information that [are] ... *for users* stands in contrast with, for example, data for the internal operations of the computer itself, e.g., the executable code responsible for converting keyboard strokes into the words on this page." Appx2018. Rather than a clear and unmistakable disclaimer of data units being of interest to a user, the patentee instead emphasized that this characteristic of a "data unit" was in fact inherent to the term, and sought to show this by, *inter alia*, referring to dictionary and treatise references of the term "document" as "information structured for human consumption ...." Appx2013.[7]

Because there was no clear and unmistakable disclaimer, and the intrinsic record, including statements during the CBM proceedings, repeatedly emphasized

[7] The district court's finding of prosecution disclaimer is also inconsistent with its own construction of "data unit." The same statements from the CBM proceedings that the district court considered disclaimers, also referred to data units as "an item of information." *See* Appx28. And yet the district court's construction for "data unit" was "an item of information." Appx30.

that a "data unit" must be of direct user interest, the district court's construction of "data unit" as simply "an item of information" was in error and should be reversed.[8]

4. The Information Facebook Asserts Are Not Stored By The Accused "Main Streams" Are Not Of Direct User Interest And, Thus, Not "Data Units"

Under a proper construction of "data unit" that excludes information that is not of direct user interest, summary judgment should have been denied because there is at least a genuine dispute of material fact as to whether the information Facebook asserted were not stored in the accused "main streams" are "data units."

For the Timeline backend system, Mirror Worlds showed that Facebook describes the information from TAO and UDB as being used for "sorting and ranking." Appx7460. Likewise, Facebook documents describe the Coefficient information as a "score," which is used in the process of "ranking ... actions." Appx7459-7460; Appx8158 (¶12). While sorting and ranking may be useful to select

[8] In a footnote the district court also rejected the user interest requirement as "indefinite because different users have different opinions about what information is of direct interest ...." Appx29. But that view was cursory, conclusory, and unsupported by credible evidence. The only support provided by the court was a citation to Facebook's claim construction brief, which itself cites only the testimony of its expert, Mr. Gray. But his testimony is conclusory in the same way – indeed, the district court appears to have copied the language in Mr. Gray's declaration and Facebook's brief. In any event, the "direct user interest" requirement is not indefinite. As the intrinsic record explains, the "user" is not a "particular user," but rather a "'user' in the generic sense, *i.e.*, a member of the class of users or end users rather than to a particular user that happens to be using a computer at that moment." Appx2016; *see also* Appx2017-2020; Appx1650; Appx1676-1678.

information that may be of direct user interest, that sorting and ranking information is not, itself, information of direct user interest. The patents, for instance, explain that similar information used to organize user's data, such as "owner, date, access permission and keywords," are not themselves "data units." Appx75 (3:61-67) ("metadata" used to organize user's data, such as "owner, date, access permission and keywords" are not themselves "documents"); *see also id*. (4:1-5) (distinguishing between "document" and "document's metadata").

For the Multifeed system, Coefficient information is used to sort or rank and is not of "direct user interest" for the same reasons. Nor is information from Ego and ads from AdFinder of direct user interest. Facebook witnesses described such information as "non-organic" content: in other words, information a user has not requested. Appx13977 (107:25-108:6); Appx14740 (41:15-23); Appx14561 ("non organic stories are feed objects like PYMK (people you may know) [(which is an example of Ego information)]" or advertisements); Appx14073-14074 (197:24-200:15) ("ads would not be organic content").

Especially when viewed in Mirror Worlds' favor, the evidence presented by Mirror Worlds established at least a genuine dispute of material fact as to whether the information relied upon by Facebook are of direct user interest and thus "data units." Summary judgment should have been denied on this alternative basis.

D. Summary Judgment Of The '439 Patent Was Improper Under A Proper Construction Of The Term "Main Collection"

Unlike the '227 and '538 patents, the asserted claim of the '439 patent refers to a "main collection of document representations" as opposed to a "main stream of data units." The district court, however, construed the '439 patent's "main collection" as synonymous as "main stream." Appx30-34. As a result, the court granted summary judgment on the same grounds as the other patents, namely that the accused systems lack a "main stream" that stores "every data unit received or generated by" the computer system. The district court's claim construction, however, is inconsistent with the intrinsic record. Its claim construction and summary judgment as to the '439 patent should, therefore, be reversed.

The intrinsic record confirms that "main collection" is different from a "main stream." The claim language expressly refers to a main collection, and not a main stream. Claim 1 generally describes a computer system that receives or generates documents in various formats, automatically generating and storing representations of those documents to "form[] a main collection of document representations," where the "main collection" is unbounded in time and size, non-transitory, without a fixed beginning or end, maintained by the system, and searchable with the results being displayed as a sub-collection of document representations. Appx137-138 (16:20-17:47). The claim uses the term in its ordinary manner, and explains the

content, function, and characteristics of the main collection. Construing the term "main collection" to mean the same thing as the "main stream" term from the two other asserted patents impermissibly rewrites the '439 patent claims to mean something different than intended. *See Karlin Tech., Inc. v. Surgical Dynamics, Inc.*, 177 F.3d 968, 971-72 (Fed. Cir. 1999) ("common sense notion that different words or phrases used in separate claims are presumed to indicate that the claims have different meanings and scope").

Moreover, nothing in the claim requires the "main collection" to have the same characteristics of a "main stream," such as (i) "time-ordered," (ii) functioning "as a diary of a person or entity's electronic life" that has "three main portions: past, present, and future," or (iii) including "every data unit received by or generated by the computer system." Claim 1 recites none of those characteristics, and includes no requirement of "every" document received or generated by the system. Rather, the main collection need only include "at least some" documents, as opposed to "every" data unit as required for a "main stream." Appx137 (16:29-30).

Conflating "main collection" with "main stream" also violates the doctrine of claim differentiation. Claim 19, and all independent claims, recite "a main collection of the collected documents," or "document representations," without further limitation on the form of the collection. *E.g.*, Appx139. Dependent claim 20 then narrows the main collection to "the form of a stream comprising a time-ordered

collection of documents that functions as a diary of a person or an entity's electronic li[fe]." *Id*. If "main collection" were synonymous with "main stream," claim 19 would have the same scope as dependent claim 20, even though claim 19 makes no reference to the form of the main collection. *See Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 910 (Fed. Cir. 2004) ("where the limitation that is sought to be 'read into' an independent claim already appears in a dependent claim, the doctrine of claim differentiation is at its strongest").

The district court's reasons for equating "main collection" with "main stream" are flawed. First, contrary to the court's conclusion, "main collection" is not a "coined" term. *See* Appx30. Nowhere in the intrinsic record did the patentee define or use the term in a manner inconsistent with its plain meaning. Indeed, the only support cited by the district court for its construction was testimony from Facebook's claim construction expert. *Id*. But the expert's testimony is conclusory and relies on nothing from the intrinsic record. Appx1452 (¶133). His assertion that there is no "established meaning in the field," ignores that "main" and "collection" are ordinary terms used in the claims in a manner consistent with their ordinary meaning. "Absent disclaimer or lexicography, the plain meaning of the claim controls." *Toshiba Corp. v. Imation Corp.*, 681 F.3d 1358, 1369 (Fed. Cir. 2012).

Second, the district court erred in concluding the patentee used the "terms 'stream' and 'collection' interchangeably. Appx31. The specification states "top-

down streams are more permanent, generally more administrative streams or collections of information" is not definitional. Appx135 (12:39-40). But that is no concession that every collection is a stream, or even that a "main collection" has the same meaning as a "main stream." Nor is the reference in the different '227 patent to a substream as a "'subset' of the main stream document collection" definitional. There, the patentee simply described a stream as a "document collection" in one embodiment. Appx76 (6:11-13). That also is not an admission that a collection is always a stream.

Third, the district court's determination that the "main collection" could not be interpreted more broadly than the specification's description of "streams" impermissibly elevated the specification above the actual claims. *See Unwired Planet, LLC,* 829 F.3d at 1359 ("'not enough that the only embodiments, or all of the embodiments, contain a particular limitation" to limit claims beyond their plain meaning"). Indeed, the district court did not address that the claims themselves differentiate between a "main collection" in the independent claims, and a "main collection ... in the form of a stream" in a dependent claim. Appx139 (20:43-46).

Under the plain and ordinary meaning of "main collection," summary judgment should have been denied. Indeed, the only ground for summary judgment asserted by Facebook turned on equating "main collection" with "main stream."

Appx18003. Facebook raised no argument that the accused systems did not infringe under the plain and ordinary meaning of the term.

E. Summary Judgment Of The '439 And '538 Patents Based On The "Glance View" Limitations Was Improper

The district court granted summary judgment of the '439 and '538 patents based on the "glance view" limitations without addressing all the evidence presented by Mirror Worlds, and its decision should therefore be reversed.

Mirror Worlds presented testimony from its expert, Dr. Koskinen, demonstrating at least a genuine dispute of material fact as to whether the accused system satisfied the "glance view" limitations. Appx12265 (*citing* Appx13326-13334, Appx13400-13402, Appx13403-13405). Dr. Koskinen's testimony was based both on screenshots of Timeline and Newsfeed, and on the source code for those features. Appx13326-13334, Appx13400-13402, Appx13403-13405. The district court, however, only considered the screenshots, finding them inadmissible, and faulting Mirror Worlds for "present[ing] no admissible evidence to create a dispute of material fact ...." Appx63.

But regardless of the screenshots, the district court overlooked Dr. Koskinen's testimony and source code relevant to the "glance view" limitation. As Dr. Koskinen explained, "Facebook renders their web site through a set of [source code] components where each component renders an area on the screen and may implement some functionality such as a hover action." Appx13326 (¶178); *see also*

Appx13329-13330 (¶183), Appx13332 (¶188), Appx13400 (¶365), Appx13403 (¶370). Referring to specific source code files, Dr. Koskinen then explained that "a profile-link component ... renders within a profile info and implements hovercard logic." *E.g.,* Appx13326 (¶178). He then identified the specific code that rendered the hovercard logic, and explained that it "create[s] a contextual-dialog component containing a Source Code component and displays it ... when hovering over a[n] ... image." *E.g.*, *id.*

Especially when viewed in Mirror Worlds' favor, Dr. Koskinen's testimony, and source code relevant to the "glance view" limitation raised genuine issues of material fact that should have precluded summary judgment. *Bynoe v. Target Corp.*, 548 Fed. Appx. 709, 711–12 (2d Cir. 2013) (vacating summary judgment where "District Court ... appears to have either missed the point of [the expert's] findings and conclusions, or failed to view them in the light most favorable to the plaintiff"); *IEX Corp. v. Blue Pumpkin Software, Inc.*, 122 Fed. Appx. 458, 467-70 (Fed. Cir. 2005) (reversing summary judgment where district court did not articulate reasons for rejecting expert's analysis).

VI. CONCLUSION AND STATEMENT OF RELIEF SOUGHT

For the forgoing reasons, this Court should vacate the district court's judgment of non-infringement, and remand.

Respectfully submitted,

October 6, 2022

/s/ Benjamin T. Wang

Marc A. Fenster
mfenster@raklaw.com
Benjamin T. Wang
bwang@raklaw.com
James Tsuei
jtsuei@raklaw.com
Russ, August & Kabat
12424 Wilshire Blvd., 12th Floor
Los Angeles, CA 90025
Tel: (310) 826-7474

Counsel for Plaintiff-Appellant
Mirror Worlds Technologies, LLC

ADDENDUM

Addendum Table of Contents

Page No.	Document
Appx1	3/8/22 Order Granting in Part and Denying in Part Facebook's Motion for Summary Judgement (Doc312)
Appx66	3/8/22 Judgement (Doc313)
Appx67	U.S. Patent No. 6,006,227
Appx90	U.S. Patent No. 7,865,538
Appx115	U.S. Patent No. 8,255,439

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

MIRROR WORLDS TECHNOLOGIES, LLC,

Plaintiff,

- against -

FACEBOOK, INC.,

Defendant.

17-cv-3473 (JGK)

OPINION AND ORDER

JOHN G. KOELTL, District Judge:

The plaintiff, Mirror Worlds Technologies, LLC ("Mirror Worlds") brought this patent infringement suit against Facebook, Inc., alleging that three systems within Facebook's social media platform infringe three patents owned by Mirror Worlds.

Mirror Worlds owns U.S. Patent Nos. 6,006,227, 7,865,538, and 8,255,439, "which describe and claim systems and methods for presenting and storing data in time-ordered streams on a computer system." Mirror Worlds Techs., LLC v. Facebook, Inc., 800 F. App'x 901, 902 (Fed. Cir. 2020).[1] The '227 patent issued from an application filed in 1996. The '538 and '439 patents claim priority to the '227 patent.

The '227 patent states that, as of its priority date, conventional computers used hierarchical directories to store

[1] Unless otherwise noted, this Opinion and Order omits all alterations, citations, footnotes, and internal quotation marks in quoted text.

and organize data. '227 patent, col. 1, lines 21-30.[2] When creating a new document under that conventional system, users were required to name the document and choose a storage location within a pre-existing directory structure. Id. The patent states that this system had several disadvantages, including: needless overhead, file names that were often meaningless to the user, and requiring the user to remember a given document's name as well as where that document was stored. See id., col. 1, lines 40-59. As an alternative system, the '227 patent describes storing documents in a chronologically ordered "stream." Id., col. 1, lines 4-6.

The '227 patent defines a "stream" as "a time-ordered sequence of documents that functions as a diary of a person or an entity's electronic life. Every document created and every document sen[t] to a person or entity is stored in a main stream." Id., col. 4, lines 6-10. A stream contains documents from the past and present, and also could contain "documents allotted to future times and events, such as[] reminders, calendar items, and to-do lists." Id., col. 4, lines 18-21. The patent explains that "[a] document can contain any type of data," including "pictures, correspondence, bills, movies, voice mail and software programs." Id., col. 4, lines 16-18.

[2] As the parties did in their briefs, the Court will refer primarily to the specification of the '227 patent. Unless otherwise noted, citations to the '227 patent apply to all three patents at issue. See ECF No. 241, at 3 n.2; ECF No. 288, at 5 n.3.

In addition to a "main stream," which contains every document in a given computer system, the patent describes "substreams." A substream is a "subset" of the main stream. Id., col. 5, lines 16-17. A user may create a substream by applying a filter to the documents in the main stream, for example "all emails I've sent to Schwartz." See id., col. 4, lines 50-51. Substreams are "dynamic" and "persistent" in that they will, at the user's request, collect applicable information as it is added to the main stream, and substreams will continue to exist "until destroyed by the user." See id., col. 5, lines 1-13. Each substream document is in the main stream, and the same document can exist in multiple substreams. Id., col. 5, lines 14-19.

Both the '538 and '439 patents incorporate by reference the patent application that issued as the '227 patent. '538 patent, col. 1, lines 14-16; '439 patent, col. 1, lines 15-17. The specifications of the '538 patent and the '439 patent are "nearly identical." Compl., ECF No. 1 ¶ 9.

Each of the claims asserted by Mirror Worlds contains a "main stream" or "main collection" limitation and a "substream" or "subcollection" limitation. Mirror Worlds Techs., 800 F. App'x at 903.[3] "The parties agree that the 'main stream' has two properties: first, it includes every data unit received or

[3] The '439 patent uses the term "main collection" instead of the term "main stream." E.g., '439 patent, claim 1. The parties dispute whether these terms are synonymous. The term "main collection" does not appear in the '227 patent or the '538 patent.

generated by the 'computer system'; second, it is a time-ordered sequence of data units." Id.

Mirror Worlds contends that three Facebook systems infringe the asserted claims: "News Feed," "Timeline," and "Activity Log." "News Feed provides a scrolling display (or 'feed') that provides stories that might be of interest to a viewing user, for example, if friends of the user posted comments or photos, uploaded videos, or performed other actions." ECF No. 241, at 13. Timeline "allows a user to share information such as text, images, photos, videos, and other types of data, with other users on Facebook." Tang Decl., ECF No. 250-17 ¶ 4. "'Activity Log' is similar to Timeline in that it can provide a list of actions that occurred on Facebook that pertain to a particular user." Id. ¶ 6.

Mirror Worlds contends that the following Facebook components satisfy the main stream/main collection and computer system limitations: in News Feed, Mirror Worlds alleges that the Multifeed System (the backend infrastructure for News Feed) meets the computer system limitation, and that the Multifeed Leaves (a storage system for keeping track of recent user actions) meets the main stream/main collection limitation. In both Timeline and Activity Log, Mirror Worlds alleges that the Timeline backend system meets the computer system limitation, and that the TimelineDB (Timeline database) meets the main

stream/main collection limitation. See Mirror Worlds Techs., 800 F. App'x at 905-06; ECF No. 288, at 4; Mirror Worlds' Response to Facebook's Statement of Material Facts ("Mirror Worlds' RTSMF"), ECF No. 288-1, at 12-13.

Mirror Worlds asserts claims 13, 14, and 17 of the '227 patent. These claims recite:

> 13. A method which organizes each data unit received by or generated by a computer system, comprising the steps of:
>
> generating a main stream of data units and at least one substream, the main stream for receiving each data unit received by or generated by the computer system, and each substream for containing data units only from the main stream;
>
> receiving data units from other computer systems;
>
> generating data units in the computer system;
>
> selecting a timestamp to identify each data unit;
>
> associating each data unit with at least one chronological indicator having the respective timestamp;
>
> including each data unit according to the timestamp in the respective chronological indicator in at least the main stream; and
>
> maintaining at least the main stream and the substreams as persistent streams.
>
> 14. The method of claim 13, wherein each timestamp is selected from the group consisting of: past, present, and future times.
>
> 17. The method of claim 13, wherein each data unit includes textual data, video data, audio data and/or multimedia data.

'227 patent, col. 16.

Mirror Worlds asserts claim 1 of the '538 patent and claim 1 of the '439 patent. Claim 1 of the '538 patent recites:

> 1. A method of operating a computer system comprising:
>
> providing the computer system with documents from diverse applications in respective formats unique to the respective applications;
>
> causing the computer system to automatically, without user interaction and without requiring a user to designate directory structures or other pre-imposed document categorizations structures, store the provided documents as a time-ordered main stream of documents associated with respective automatically generated time indicators;
>
> said time-ordered main stream being unbounded to thereby accommodate documents associated with time indicators related to past, present and future times;
>
> said time-ordered main stream requiring no fixed beginning or end and being maintained and being selectively retrievable and searchable by the computer system;
>
> said computer system maintaining the main stream live and responsive to subsequent events by automatically incorporating therein new documents as provided to the computer system while maintaining the thus expanded main stream time-ordered;
>
> providing selected search criteria;
>
> causing said computer system to search said time-ordered main stream according to said search criteria and use search results to create a time-ordered substream of documents from the main time-ordered stream;
>
> further causing said computer system to maintain said substream live and responsive to subsequent events by automatically incorporating therein new document provided to the computer system that meet the search

criteria while maintaining the thus expanded substream time-ordered;

displaying at least selected portion of the live main stream or substream on computer display means as a display reflecting the time-ordered nature thereof;

automatically showing on the display means a display of a glance view of a displayed document in response to touching with a cursor a screen area associated with the document;

said glance view being an abbreviated version of the document and indicative of content thereof; and

said showing of the glance view occurring essentially instantaneously in response to said touching with the cursor of the screen area associated with the document.

Claim 1 of the '439 patent recites:

1. A method of operating a computer system comprising:

providing the computer system with documents in respective formats according to respective different applications through which the provided documents are generated or modified, which formats differ from one of the document to another for at least some of said provided documents, said provided documents being delivered to the computer system or generated by the computer system;

storing at least some of the documents provided to the computer system in computer storage;

said computer system being configured to automatically generate and store in computer storage respective representations related to the documents provided thereto, thereby forming a main collection of document representations;

said computer automatically generating and storing said main collection of document representations without requiring a user to designate a directory structure, a physical location for storage of document representations or corresponding documents, or another

pre-imposed document categorization structure for each of said document representations or documents;

said automatically generated and stored document representations being in a consistent format despite differences in format from one to another of the documents corresponding thereto;

said automatically generated and stored representations of said documents including respective automatically generated time indicators associated with the documents corresponding to said representations;

said automatically generated and stored representations of said documents further including respective automatically generated information relating the document representation to the respective documents provided to and stored in said computer system;

said automatically generated and stored main collection of document representations being unbounded in time and size and being configured to include documents associated with time indicators related to future times as well as to past and present times;

said automatically generated and stored main collection of document representations requiring no fixed beginning or end and being non-transitory and selectively searchable by the computer system;

providing selected search criteria;

causing said computer system to perform a first search of at least said main collection of document representations according to selected first search criteria, to provide first search results, and to utilize said first search results to generate a first sub-collection of document representations related to a respective sub-collection of the documents provided to the computer system;

selectively causing the computer system to display on a computer screen graphical depictions of only a first portion of said first subcollection of document representations generated by utilizing said first

search results, said first portion corresponding to only a portion of the documents provided to and stored in the computer system;

said first portion of said first document sub-collection of document representations corresponding to a multi-document portion of the documents provided to and stored in said computer system;

said computer system being configured to maintain at least one of said the main collection of document representations and said first sub-collection of document automatically responsive to events subsequent to the providing of said first search results such that additional document representations corresponding to additional documents provided to the computer system subsequent to an initial display of said first portion of the first sub-collection of document representations and meeting said selected search criteria are automatically included in a subsequent display of graphical depictions of one or more portions of said first sub-collection of document representations;

said additional document representations also including automatically generated respective time indicators associated with the documents subsequently provided to the computer system;

automatically showing on the computer screen a display of a glance view of a displayed document depiction while continuing to show on the screen plural displayed graphical depictions of respective plural document representations;

said glance view being an abbreviated version of the document corresponding to the graphical depiction and being indicative of content thereof; and

said showing of the glance view occurring in response to a user designating a displayed document representation by interacting with a screen area currently associated with the graphical depiction, without requiring the user to click on the designated screen area in order to enable such showing of the glance view.

In addition to the main stream/main collection limitation, the asserted claims of the '538 and '439 patents require the display of a "glance view," or a preview, of a document that the user interacts with. '538 patent, col. 16, lines 55-60; '439 patent, col. 17, lines 34-41.

I.

Facebook previously moved for summary judgment on the issue of non-infringement before discovery was completed, and the Court granted that motion in a Memorandum Opinion and Order dated August 11, 2018. 320 F. Supp. 3d 538 (S.D.N.Y. 2018). The Court concluded that the Multifeed Leaves and the TimelineDB (the accused main streams) could not be main streams because the record established that the accused computer systems receive data from TAO (a storage system which stands for "The Associations and Objects") that does not enter the Multifeed Leaves or the TimelineDB. See id. at 547-49. The Federal Circuit Court of Appeals held that conclusion was erroneous and declined to affirm the grant of summary judgment on alternative grounds urged by Facebook. See 800 F. App'x at 909-11. Accordingly, the court of appeals reversed and remanded, noting that its ruling was "without prejudice to otherwise-appropriate consideration of non-infringement contentions on remand, especially once the record is fully developed." Id. at 910. Discovery in this case has now closed, and Facebook has filed a new motion for summary

judgment relying, in part, on additional grounds of non-infringement.

There are several motions before the Court. Facebook has brought another motion for summary judgment, making three arguments in particular: (1) the asserted claims are ineligible for patent protection under 35 U.S.C. § 101; (2) Facebook does not infringe any of the asserted claims; and (3) there was no willful infringement in this case. See ECF No. 241. Mirror Worlds has brought a motion for partial summary judgment of no invalidity based on Facebook's prior art defenses. See ECF No. 233. Both parties have brought motions to exclude certain opinions of the opposing party's experts. See ECF Nos. 223, 228, 236, 246. Finally, there are fully-briefed claim construction disputes before the Court, which the court of appeals clarified are open on remand. See 800 F. App'x at 911; ECF Nos. 103-1, 108, 110.

For the reasons explained below, Facebook's motion for summary judgment is **granted in part and denied in part**. The Court will focus primarily on Facebook's motion for summary judgment, and will discuss the other outstanding motions to the extent they are relevant in resolving Facebook's motion for summary judgment.

II.

"The court shall grant summary judgment if the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law." Fed. R. Civ. P. 56(a); see also Celotex Corp. v. Catrett, 477 U.S. 317, 322-23 (1986). "[T]he trial court's task at the summary judgment motion stage of the litigation is carefully limited to discerning whether there are any genuine issues of material fact to be tried, not to deciding them. Its duty, in short, is confined at this point to issue-finding; it does not extend to issue-resolution." Gallo v. Prudential Residential Servs., Ltd. P'ship, 22 F.3d 1219, 1224 (2d Cir. 1994). The moving party bears the initial burden of "informing the district court of the basis for its motion" and identifying the matter that "it believes demonstrate[s] the absence of a genuine issue of material fact." Celotex, 477 U.S. at 323. The substantive law governing the case will identify which facts are material, and "[o]nly disputes over facts that might affect the outcome of the suit under the governing law will properly preclude the entry of summary judgment." Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 248 (1986).

In determining whether summary judgment is appropriate, the court must resolve all ambiguities and draw all reasonable inferences against the moving party. See Matsushita Elec. Indus.

Co., Ltd. v. Zenith Radio Corp., 475 U.S. 574, 587 (1986). With respect to the issues on which summary judgment is sought, if there is any evidence in the record from any source from which a "reasonable inference" could be drawn "in favor of the non-movant," summary judgment is improper. Roche Palo Alto LLC v. Apotex, Inc., 531 F.3d 1372, 1377 (Fed. Cir. 2008). The nonmoving party must "point to an evidentiary conflict created on the record," and may not rely only on "mere denials or conclusory statements." Armco, Inc. v. Cyclops Corp., 791 F.2d 147, 149 (Fed. Cir. 1986); see also Fed. R. Civ. P. 56(c).

III.

Facebook argues that all the asserted claims claim ineligible subject matter under 35 U.S.C. § 101.

Section 101 of the Patent Act, which defines the subject matter eligible for patent protection, provides: "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." 35 U.S.C. § 101. The Supreme Court has held that "this provision contains an important implicit exception: Laws of nature, natural phenomena, and abstract ideas are not patentable." Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 573 U.S. 208, 216 (2014) (quoting Ass'n for

Molecular Pathology v. Myriad Genetics, Inc., 569 U.S. 576, 589 (2013)).

The "concern that drives this exclusionary principle [is] one of pre-emption." Id. "Laws of nature, natural phenomena, and abstract ideas are 'the basic tools of scientific and technological work.' 'Monopolization of those tools through the grant of a patent might tend to impede innovation more than it would tend to promote it,' thereby thwarting the primary object of the patent laws." Id. (quoting Myriad, 569 U.S. at 589, and Mayo Collaborative Servs. v. Prometheus Lab'ys, Inc., 566 U.S. 66, 71 (2012)). However, courts must "tread carefully in construing this exclusionary principle" because, "[a]t some level, 'all inventions embody, use, reflect, rest upon, or apply laws of nature, natural phenomena, or abstract ideas.'" Id. at 217 (quoting Mayo, 566 U.S. at 71).

In Alice, the Supreme Court reaffirmed the two-step framework set out in Mayo for analyzing patent eligibility under § 101. At the first step, the court asks whether the claims at issue are directed to a patent-ineligible concept. If so, the court proceeds to step two and asks whether the claims provide an "inventive concept"—that is, "an element or combination of elements that is 'sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the

ineligible concept itself.'" Id. at 217-18 (quoting Mayo, 566 U.S. at 73).

A.

Facebook argues that the asserted claims "are directed to the abstract idea of organizing information in a time-ordered manner." ECF No. 241, at 8. Facebook cites the specification of the '227 patent, which states that the claimed invention is "a new model and system for managing personal electronic information which uses a time-ordered stream as a storage model and stream filters to organize, locate, summarize and monitor incoming information." '227 patent, col. 3, lines 62-65. Facebook also cites a line of Federal Circuit cases holding that collecting, analyzing, and displaying information is an abstract idea. See, e.g., Elec. Commc'n Techs., LLC v. ShoppersChoice.com, LLC, 958 F.3d 1178, 1182 (Fed. Cir. 2020) (abstract idea of "gathering, storing, and transmitting information"); Intellectual Ventures I LLC v. Cap. One Fin. Corp., 850 F.3d 1332, 1340 (Fed. Cir. 2017) (abstract idea of "collecting, displaying, and manipulating data"); Content Extraction and Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n, 776 F.3d 1343, 1347 (Fed. Cir. 2014) ("The concept of data collection, recognition, and storage is undisputedly well-known. Indeed, humans have always performed these functions.").

Mirror Worlds argues that the asserted claims are "directed to the way in which computers name, organize, and retrieve electronic documents and distinguish the way in which conventional computers had done so." ECF No. 288, at 6 (quoting Apple, Inc. v. Mirror World Techs., LLC, No. CBM2016-00019 (P.T.A.B. May 26, 2016) ("PTAB Decision"), ECF No. 289-4, at 15). Mirror Worlds argues that the asserted claims are patent-eligible and not directed to an abstract idea because they are directed to improving computer functionality. See Alice, 573 U.S. at 225; Enfish, LLC v. Microsoft Corp., 822 F.3d 1327, 1335-36 (Fed. Cir. 2016). Mirror Worlds also notes that a federal district court and the Patent Trial and Appeal Board ("PTAB") previously concluded that the claims of the '227 patent are patent-eligible under § 101. See Mirror Worlds Techs., LLC v. Apple Inc., No. 6:13-cv-419, 2015 WL 6750306, at *10 (E.D. Tex. July 7, 2015); PTAB Decision, at 17-18.

"The Supreme Court has suggested that claims purporting to improve the functioning of the computer itself, or improving an existing technological process might not succumb to the abstract idea exception." Enfish, 822 F.3d at 1335. The cases construing the "abstract idea" exception in the computer context therefore draw a distinction between (1) claims that are directed to improving computer functionality, and (2) claims that are

directed to "a process that qualifies as an 'abstract idea' for which computers are invoked merely as a tool." Id. at 1335-36.

Enfish made clear that the court may ask whether the asserted claims are directed to improving computer functionality at Alice step one. See id. Enfish also made clear that, while improvements to computer-related technology "such as a chip architecture, an LED display, and the like" are "undoubtedly not abstract," "[s]oftware can make non-abstract improvements to computer technology just as hardware improvements can." Id. at 1335. The court in Enfish also stated that, because there is no "definitive rule to determine what constitutes an 'abstract idea' sufficient to satisfy" Alice step one, "both [the Federal Circuit] and the Supreme Court have found it sufficient to compare claims at issue to those claims already found to be directed to an abstract idea in previous cases." Id. at 1334.

In Enfish, the asserted claims related to a self-referential computer database structure. The self-referential database improved upon the prior art "relational" model by (1) including all data entities in a single table rather than providing each type of data in a separate table, and (2) defining a table's columns by rows in that same table. See id. at 1330-32. The patents in Enfish taught that these design improvements allowed for faster searching of data, more effective storage of certain types of data, and more flexibility

in configuring a database. See id. at 1333. The court concluded that the asserted patents were not directed to an abstract idea at Alice step one; rather, they were "directed to a specific improvement to the way computers operate, embodied in the self-referential table." Id. at 1336. The court emphasized the patents' disparagement of the prior art relational model. See id. at 1337. The court also found that the invention's "ability to run on a general-purpose computer" did not "doom[] the claims" because the claims were directed to improving computer functionality, unlike the claims in cases such as Alice that "can readily be understood as simply adding conventional computer components to well-known business practices." Id. at 1338.

The claims asserted in this case are similar, even though their focus is broader than the claims at issue in Enfish. The claims asserted here are also aimed at improving the storage and retrieval of data on a computer. The patents teach that conventional operating systems were cumbersome, difficult to navigate, and carried several disadvantages: users had to store new information in fixed categories, archiving was not automatic, and "the historical context of a document [was] lost because no tracking of where, why and how a document evolves [was] performed." '227 patent, col. 1, lines 42-52. The claimed invention attempts to solve these problems by storing documents

in a computer system in time-ordered streams. See id., col. 2, lines 13-16. Because the claimed invention stores all data units in a time-ordered main stream, the user is not required to store documents in a pre-imposed directory structure, documents are automatically archived, and the user can see how a document has evolved over time. The claimed operating system also allows the user to avoid the "unneeded overhead" of naming a file and choosing a storage location whenever a file is created. See id., col. 1, lines 42-44; id., col. 2, lines 20-24 (under the claimed operating system, "the storage of the files is handled automatically and file names are only used if a user chooses to invent such names"). This is similar to the way in which the patents in Enfish improved upon the functionality of prior art computer systems. See 822 F.3d at 1333 ("[W]hereas deployment of a relational database often involves extensive modeling and configuration of the various tables and relationships in advance of launching the database, Enfish argues that the self-referential database can be launched without such tasks and instead configured on-the-fly."). The asserted claims seek to improve upon traditional computer operating systems by storing and retrieving data in a more efficient way. Accordingly, the claims are directed to improving computer functionality.[4]

[4] The two prior decisions that considered the patent eligibility of the '227 patent reached the same conclusion. The PTAB Decision, relying on Enfish and DDR Holdings, LLC v. Hotels.com, L.P., 773 F.3d 1245 (Fed. Cir. 2014),

Facebook argues that "[t]he asserted claims fall squarely within the oft-cited line of Federal Circuit cases repeatedly invalidating claims directed to collecting, analyzing, and displaying information." ECF No. 241, at 9. This is not so. The claims in the cases cited by Facebook used the computer as a tool to collect and analyze data for some purpose other than improving computer functionality. See, e.g., ShoppersChoice.com, 958 F.3d at 1181 ("providing advance notification of the pickup or delivery of a mobile thing"); SAP Am., Inc. v. InvestPic, LLC, 898 F.3d 1161, 1164-67 (Fed. Cir. 2018) (statistical method for analyzing financial data); Intellectual Ventures I, 850 F.3d at 1339-40 (method for editing specific type of computer documents); Electric Power Grp., LLC v. Alstom S.A., 830 F.3d 1350, 1351-53 (Fed. Cir. 2016) ("systems and methods for performing real-time performance monitoring of an electric power grid"). Those claims, like the claims in Alice, used computers to perform a process or task that itself qualified as an

---

emphasized that the '227 patent sought to solve problems "that arose specifically in the realm of computer operating systems" and "that did not exist in the pre-computer world." PTAB Decision, at 15-17. The district court in Mirror Worlds Techs. v. Apple, 2015 WL 6750306, concluded that the '227 patent was directed to an abstract idea but that the asserted claims contained a sufficient "inventive concept" at Alice step two. See id. at *8-10. However, the district court also concluded that the '227 patent was "directed to improving computer technology," id. at *8, and the district court's decision came before the Federal Circuit Court of Appeals clarified in Enfish that claims may be deemed patent-eligible at step one because they are directed to improving computer functionality. Accordingly, both decisions that have considered the patentability of the '227 patent under § 101 support the conclusion that the asserted claims are patent-eligible because they are directed to improving computer functionality.

abstract idea. The asserted claims in this case, by contrast, are directed to improving how computers "carry out one of their basic functions of storage and retrieval of data." Id. at 1354.

Facebook also argues that the claims are "directed at using computers to solve a human problem of storing and organizing information." ECF No. 241, at 10. But "describing the claims at such a high level of abstraction and untethered from the language of the claims all but ensures that the exceptions to § 101 swallow the rule." Enfish, 822 F.3d at 1337; see also Alice, 573 U.S. at 217 ("[W]e tread carefully in construing this exclusionary principle [of laws of nature, natural phenomena, and abstract ideas] lest it swallow all of patent law."). It is true that storing and organizing information is a human problem that exists independent of computers. But the asserted claims are directed to a specific improvement in the way that computers store and retrieve data; the patents articulate, and attempt to overcome, challenges created by traditional prior art operating systems that are specific to the computer context. See DDR Holdings, LLC v. Hotels.com, L.P., 773 F.3d 1245, 1257–59 (Fed. Cir. 2014) (concluding that the claims at issue were patent-eligible under § 101 because "the claimed solution is necessarily rooted in computer technology in order to overcome a problem specifically arising in the realm of computer networks").

Accordingly, the asserted claims are directed to improving computer functionality. Facebook fails to distinguish Enfish in a meaningful way or to explain why the prior decisions holding the '227 patent patent-eligible under § 101 are mistaken. And while Facebook emphasizes the breadth of the claim language and its lack of technological details, this is due—at least in part—to the breadth of the problem that the claims aim to solve and the fact that data storage and retrieval are fundamental aspects of computer functionality. Moreover, Facebook points to no cases where claims directed to improving computer functionality were found ineligible under § 101 due to a lack of implementation details or for any other reason.[5]

Because the claims are directed to improving computer functionality, the claims are not directed to an abstract idea at Alice step one and the Court need not reach step two. See Enfish, 822 F.3d at 1339.[6]

---

[5] At argument, Facebook relied on Berkheimer v. HP Inc., 881 F.3d 1360 (Fed. Cir. 2018). In Berkheimer, the court of appeals explained that the use of a "parser" to transform a data structure from source code to object code was an abstract idea without evidence that this transformation improved computer functionality in some way. See id. at 1367. The asserted claims in this case, however, describe a stream-based operating system that is directed to improving computer functionality.

[6] The Court concludes that all the asserted claims are directed to improving the way in which computers store and retrieve data. Claims 14 and 17 of the '227 patent, which depend on independent claim 13, are substantially similar to claim 13. Claim 14 simply clarifies that past, present, and future timestamps can be selected, and claim 17 clarifies that all types of data may be included in the time-ordered streams. See '227 patent, col. 16, lines 26-28, 38-40. Claim 1 of the '538 patent and claim 1 of the '439 patent are also substantially similar to claim 13 of the '227 patent. While these claims recite an additional "glance view" limitation that provides an "abbreviated version of the document corresponding to the graphical depiction and being

IV.

The Court turns next to the issue of infringement and the parties' related claim construction disputes.

Infringement analysis is a two-step process: "The first step is determining the meaning and scope of the patent claims asserted to be infringed. The second step is comparing the properly construed claims to the device accused of infringing." Markman v. Westview Instruments, Inc., 52 F.3d 967, 976 (Fed. Cir. 1995) (en banc), aff'd, 517 U.S. 370 (1996); see also N. Am. Container, Inc. v. Plastipak Packaging, Inc., 415 F.3d 1335, 1344 (Fed. Cir. 2005).

A.

Claim construction, the first step in infringement analysis, is a matter of law. See Markman, 52 F.3d at 979. "It is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude." Phillips v. AWH Corp., 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc). "A court construing a patent claim seeks to accord a claim the meaning it would have to a person of ordinary skill in the art at the time of the invention."

---

indicative of content thereof," '538 patent, col. 16, lines 55-60; '439 patent, col. 17, lines 34-41, both claims focus on the stream-based, time-ordered storage and retrieval of data on a computer. "[B]ecause all the claims are substantially similar," addressing each claim separately for the purposes of § 101 is unnecessary. Content Extraction and Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n, 776 F.3d 1343, 1348 (Fed. Cir. 2014). Accordingly, claim 13 of the '227 patent is representative of all the asserted claims for purposes of the § 101 analysis, and all the asserted claims are directed to improving computer functionality.

Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc., 381 F.3d 1111, 1116 (Fed. Cir. 2004) (collecting cases). "The inquiry into how a person of ordinary skill in the art understands a claim term provides an objective baseline from which to begin claim interpretation." Phillips, 415 F.3d at 1313. "[T]he words of a claim are generally given their ordinary and customary meaning." Id. at 1312.

"[T]he 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 disputed term appears, but in the context of the entire patent, including the specification." Id. at 1313. In deciding issues of claim construction, "the court looks to those sources available to the public that show what a person of skill in the art would have understood disputed claim language to mean. Those sources include the words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art." Id. at 1314. "[T]he context of the surrounding words of the claim also must be considered in determining the ordinary and customary meaning of those terms." Id. "[W]hile extrinsic evidence can shed useful light on the relevant art, . . . it is less significant than the intrinsic record in determining the legally operative meaning of claim language." Id. at 1317.

"Only those terms need be construed that are in controversy, and only to the extent necessary to resolve the controversy." Eon Corp. IP Holdings v. Silver Spring Networks, 815 F.3d 1314, 1318-19 (Fed. Cir. 2016). The parties agree that all the asserted claims recite a main stream or main collection limitation, which requires Mirror Worlds to identify a stream that is inclusive of every data unit received or generated by the alleged computer system. There are two disputed terms, however, that must be decided in order to resolve Facebook's motion for summary judgment on non-infringement: "data unit" and "main collection."

"Data unit" is in issue because the parties' infringement dispute hinges on whether the Multifeed Leaves and the TimelineDB, the alleged main streams/main collections, include every data unit received or generated by the alleged computer systems (the Multifeed System and the Timeline backend system, respectively). To determine whether the alleged main streams/main collections include every data unit received or generated by the alleged computer systems, the Court must first clarify what is and what is not a "data unit."

Facebook contends that "data unit" means "document." First, Facebook points to the prosecution history of the '227 patent, which is "designated as part of the 'intrinsic evidence,'" Phillips, 415 F.3d at 1317. Facebook cites a 1999 amendment

submitted by the patentees in which the patentees sought to "clarify key terms" with express definitions. See ECF No. 109-1, at FBMW_00097563.[7] In this amendment, the patentees state: "A 'data unit' is a 'document' because a 'document can contain any type of data.'" Id.[8] The amendment went on to use the terms "documents" and "data units" interchangeably. See, e.g., id. ("a time-ordered sequence of documents (data units)"). Facebook notes that "The patentee is held to what he declares during the prosecution of his patent." Gillespie v. Dywidag Sys. Int'l, USA, 501 F.3d 1285, 1291 (Fed. Cir. 2007). Facebook also notes that, in connection with a covered business method ("CBM") review of the '227 patent, Mirror Worlds stated that "The broadest reasonable construction of 'data unit' is 'document.'" ECF No. 103-3, at 61; see also Aylus Networks, Inc. v. Apple, Inc., 856 F.3d 1353, 1359-60 (Fed. Cir. 2017) (doctrine of prosecution disclaimer applies to statements made in post-issuance proceedings).

Mirror Worlds contends that "data unit" means "an item of information that is of direct user interest." Mirror Worlds

[7] By its text, the 1999 amendment sought to clarify "key terms in the amended claims." ECF No. 109-1, at FBMW_00097563 (emphasis added). The asserted independent claim 13 is one of the claims that was amended, see id. at FBMW_00097555, and, in any event, the '227 patent provides no basis for defining "data unit" differently across different claims. See Phillips, 415 F.3d at 1314 ("[C]laim terms are normally used consistently throughout the patent[.]").

[8] The patentees were quoting the then-current version of the specification. This language appears in the final specification. '227 patent, col. 4, lines 16-18 ("A document can contain any type of data including but not limited to pictures, correspondence, bills, movies, voice mail and software programs.").

argues that "document," Facebook's proposed construction, has two deficiencies: (1) it fails to account for the fact that data unit only encompasses information that is of direct user interest; and (2) it fails to clarify that a data unit can include video, audio, and multimedia data as opposed to only textual data.

Mirror Worlds' first contention is without merit. First, it finds no support in the claim language or the specification. While the specification does make clear that the invention seeks to provide a system for organizing and managing "personal electronic information," '227 patent, col. 3, lines 62-63, the patent's overarching aim of displaying information that is interesting to the user cannot be crammed into the term "data unit." Displaying information that is of interest to the user is a goal of the invention writ large; it is not captured in these two words. A person of ordinary skill in the art would not understand "data unit" to refer only to those units of data that happen to interest a particular user at a particular time.[9]

Second, the doctrine of prosecution disclaimer precludes Mirror Worlds' construction. In their preliminary response filed in connection with a CBM review, the patentees disputed the petitioner's indefiniteness challenge by disclaiming the very

[9] Nor would a person of ordinary skill in the art interpret "data unit" to refer only to data that is of direct interest to a "user" in the generic sense, as the patentees once argued. See ECF No. 103-3, at 72.

construction that they urge this Court to adopt. See ECF No. 103-3, at 65-66 ("The definitions of 'data unit' in both the specification and prosecution history do not include the narrowing limitation that forms the basis of Petitioner's indefiniteness challenge—i.e., 'an item of information that is of direct user interest in the user's timeline.'"). This is a "clear and unmistakable" disavowal of the "direct user interest" limitation, and Aylus makes clear that the patentees cannot argue claims "one way in order to maintain their patentability and in a different way against accused infringers." 856 F.3d at 1360.[10] Even apart from the doctrine of disclaimer, the patentees persuasively argued in their preliminary response that the '227

[10] While Aylus specifically held that the doctrine of prosecution disclaimer applies to statements made in inter partes review ("IPR") proceedings, the logic of Aylus and the policy behind prosecution disclaimer apply equally to CBM proceedings. Applying prosecution disclaimer in this case will "promote the public notice function of the intrinsic evidence and protect the public's reliance on definitive statements made during" CBM proceedings. Aylus, 856 F.3d at 1360. Further in support of applying the doctrine in this case is the fact that the patentees argued to the PTAB that "document" is the best interpretation of "data unit," not merely the broadest reasonable interpretation ("BRI"). See, e.g., ECF No. 103-3, at 64 ("Consistent with the implicit definition of 'data unit' in the '227 specification, the patent applicant explicitly defined 'data unit' as a 'document' during prosecution."). Mirror Worlds appears to argue that the Court should not consider Mirror Worlds' statement in its CBM preliminary response because the BRI standard governed claim construction in the CBM review. This argument is unpersuasive because the BRI standard governed claim construction in IPR proceedings when Aylus was decided in 2017. See Celgene Corp. v. Peter, 931 F.3d 1342, 1349 n.8 (Fed. Cir. 2019) (noting that the revised claim construction standard for IPR proceedings applies only to petitions filed on or after November 13, 2018). Mirror Worlds points to no cases undermining the holding of Aylus. And while counsel for Mirror Worlds states that "the patent owner's statements were always expressly made under the BRI standard, and were expressly distinguished from the Phillips standard," ECF No. 310, at 2, that is simply not true. As the quoted text from the preliminary response demonstrates, Mirror Worlds repeatedly argued in the preliminary response that the patentees—explicitly during prosecution and implicitly in the language of the '227 patent—defined "data unit" as "document" without any narrowing limitation.

patent defines the term "data unit" as "document" by implication because "the patentee uses 'data unit' throughout the entire specification in a manner consistent with, and as a synonym for, 'document.'" ECF No. 103-3, at 63 (citing Irdeto Access, Inc. v. Echostar Satellite Corp., 383 F.3d 1295, 1301 (Fed. Cir. 2004)). Accordingly, the doctrine of prosecution disclaimer applies and, furthermore, the intrinsic evidence does not support Mirror Worlds' "of direct user interest" limitation.[11]

[11] The Court also agrees with Facebook that Mirror Worlds' proposed construction is "confusing," "subjective," and would render the term "data unit" indefinite because different users have different opinions about what information is of direct interest, and even the same user may be interested in different information at different times. See ECF No. 108, at 3. While the court in Mirror Worlds Techs., LLC v. Apple Inc., No. 6:13-cv-419, 2015 WL 179072, at *3 (E.D. Tex. Jan. 14, 2015) accepted the "of direct user interest" limitation, this Court disagrees with that construction. The Court does not find a basis for this limitation in the language of the patent. The district court in Texas concluded that the "of direct user interest" limitation "arose during prosecution in an examiner interview and is thus expressly supported." Id. (citing a January 19, 1999 interview summary by the patent examiner). But the 1999 amendment, which was made in direct response to the January 19, 1999 interview, is more important to the construction of the claim term than the interview summary prepared by the examiner. In the amendment, the patentees stated that the "primar[y]" amendment in light of the January 19, 1999 interview was to clarify that "'each data unit received by or generated by the computer system' is received by the 'main stream'. In other words, all the data units, without regard to whether a data unit was generated internally or externally, are of significance to the user." ECF No. 109-1, at FBMW_00097562-63. This language appears to emphasize that all the information contained in the computer system is included in the main stream, regardless of its source. This language does not appear to add a new "of direct user interest" limitation to the term "data unit." The amendment goes on: "[S]ubstreams allow a user to determine the events of direct user interest from the stream of data units of significance to the user (main stream)." Id. at FBMW_00097563 (emphasis added). This contrast between substreams and the main stream makes clear that the substream is the layer in which the user has the opportunity to filter information that is of direct interest. The main stream, on the other hand, contains all information that is received or generated by the computer system, without any selection criteria from the user. The language of the patent provides no basis for concluding that only data "of direct user interest" is included in the computer system, and a person of ordinary skill in the art would not understand "data unit" to be so limited. And, given the opportunity to amend

Mirror Worlds' other contention—that "document" fails to clarify that a data unit can include video, audio, and multimedia data—has more force. The specification makes clear that "A document can contain any type of data including but not limited to pictures, correspondence, bills, movies, voice mail and software programs." '227 patent, col. 4, lines 16-18. Indeed, Facebook does not dispute that "data unit" encompasses any type of data, not merely textual data. The Court agrees with Mirror Worlds that "an item of information" is clearer in this respect than "document." Accordingly, the Court construes "data unit" as "an item of information."[12]

The second disputed term is "main collection." The term "main collection" appears only in the '439 patent. Facebook argues that "main collection" means "main stream." Mirror Worlds argues that, in claim 1 of the '439 patent, "main collection" means "main collection of document representations."[13]

The Court agrees with Facebook. "Main collection" is a coined term with no established meaning in the field. See Gray Decl., ECF No. 108-1 ¶ 133. The specification of the '439 patent

---

the claim language in light of the January 19, 1999 interview, the patentees expressly defined "data unit" as "document," without any limiting language.

[12] The Court "need not accept the constructions proposed by either party." Bancorp Servs., L.L.C. v. Sun Life Assurance Co. of Canada (U.S.), 687 F.3d 1266, 1274 (Fed. Cir. 2012).

[13] In its claim construction brief, Mirror Worlds urges a different construction of "main collection" for claims 9, 19, and 20 of the '439 patent. However, only claim 1 of the '439 patent is at issue. Facebook's Statement of Material Facts, ECF No. 241-1 ¶ 2; Mirror Worlds' RTSMF ¶ 2. Accordingly, the Court will only consider Mirror Worlds' proposed construction for claim 1.

does not use the term "main collection." But, like the '227 patent, the '439 specification discusses "time-ordered streams" and a "stream-based" operating system. See '439 patent, title; id., col. 1, line 67-col. 2, line 1.[14] The '439 patent also explicitly incorporates the '227 patent application in its entirety, which discusses main streams at length. See id., col. 1, lines 15-17. And as Facebook notes, "claim drafters can [] use different terms to define the exact same subject matter." Curtiss-Wright Flow Control Corp. v. Velan, Inc., 438 F.3d 1374, 1380 (Fed. Cir. 2006). The '439 patent uses the terms "stream" and "collection" interchangeably. See '439 patent, col. 12, lines 39-42 (emphasis added) ("Top-Down streams are more permanent, generally more administrative streams or collections of information . . . ."). The '227 patent also uses these terms interchangeably. See '227 patent, col. 5, lines 16-17 ("A substream, in other words, is a 'subset' of the main stream document collection."); id., col. 6, lines 11-13 ("Each view of a stream is implemented as a client of the server and provides the user with a 'viewpoint' interface to document collections, that is, streams."). Because the patents use the terms "stream" and "collection" interchangeably, "main stream" and "main

[14] See also '439 patent, abstract (emphasis added) ("A st[r]eam-based document storage and retrieval system accepts documents that are in diverse formats and come from diverse application, automatically creates document model objects describing these documents in a consistent format and associating time stamps with the documents to automatically create a main stream in chronological order.").

collection" are synonymous. Neither the specification of the '439 patent (which does not use the term "main collection" at all) nor the claim language provides a basis for concluding otherwise.

Mirror Worlds argues that "The patentees expressly used 'main collection' when referring to documents or document representations, and a 'main stream' when referring to data units." ECF No. 103-1, at 11. But this is plainly untrue. See, e.g., '227 patent, col. 5, lines 14-19 (emphasis added) ("Although a document may belong to any number of substreams, the document also enters and remains on the main stream. A substream, in other words, is a 'subset' of the main stream document collection. In other words, a way of looking at the main stream so as to exclude certain documents temporarily."). The '439 patent's use of the word "document" instead of "data unit" is of no moment because the patentees, in the prosecution history of the '227 patent, expressly stated that these terms are synonymous. ECF No. 109-1, at FBMW_00097563.

Mirror Worlds also argues that "main collection" cannot mean "main stream" because claim 1 of the '439 patent does not require characteristics that the parties agree describe a stream, namely: (1) being time-ordered; (2) functioning "as a diary of a person or entity's electronic life" that has "three main portions: past, present, and future," and (3) including

"every data unit received by or generated by the computer system." ECF No. 103-1, at 11. But any construction of "main collection" that lacked these characteristics would be impermissibly untethered to the specification. Because "main collection" is a coined term with no "established meaning to one of ordinary skill in the art," it must be interpreted in light of the specification and "cannot be construed broader than the disclosure in the specification." Indacon, Inc. v. Facebook, Inc., 824 F.3d 1352, 1357 (Fed. Cir. 2016). The '439 specification clearly states that the invention incorporates the characteristics of a stream. See, e.g., '439 patent, col. 1, line 67-col. 2, line 3 ("The system is stream-based in that it creates time-ordered streams of information items or assets, beginning with the oldest and continuing through current and on to future items."); id., col. 2, line 67-col. 3, line 2 (emphasis added) ("One can search on any word or phrase, as every word in every document is indexed, on document types and metadata, and on time-related data[.]"). The patentees' use of an undefined, coined term in the claim language cannot broaden the invention described in the specification. See Retractable Techs., Inc. v. Becton, Dickinson and Co., 653 F.3d 1296, 1305 (Fed. Cir. 2011) (construction must "tether the claims to what the specifications indicate the inventor actually invented").

"Main collection" also cannot lack the characteristics of a stream because those characteristics are what set the invention apart from the prior art that the specification disavows. "Where the general summary or description of the invention describes a feature of the invention and criticizes other products that lack that same feature, this operates as a clear disavowal of these other products." Edwards Lifesciences LLC v. Cook Inc., 582 F.3d 1322, 1333 (Fed. Cir. 2009). The specification of the '439 patent, like the '227 specification, specifically disavows the "traditional storage and retrieval systems" that were not time-ordered. '439 patent, col. 1, lines 49-67. Mirror Worlds' construction fails because it seeks to capture, through claim construction, the non-time-ordered systems the specification specifically disavows.[15]

Accordingly, the Court construes "main collection" in claim 1 of the '439 patent as "main stream." Because the Court construes "main collection" as "main stream," the analysis below concerning the main stream limitation applies with equal force to all the patents at issue.

[15] Similarly, the fact that claim 1 of the '439 patent contains vague language leaving open the possibility that the "main collection" includes "at least some," but not necessarily all, of the documents in the computer system, '439 patent, col. 16, line 29, does not save Mirror Worlds' construction because "the specifications tell us otherwise." Retractable Techs., 653 F.3d at 1305; see '439 patent, col. 3, line 1 ("every word in every document is indexed"); id., abstract (emphasis added) (the system "automatically create[s] a main stream in chronological order").

B.

The second step of infringement analysis "is comparing the properly construed claims to the device accused of infringing." Markman, 52 F.3d at 976. Mirror Worlds alleges that three Facebook features infringe the asserted patents: News Feed, Timeline, and Activity Log.

Mirror Worlds pursues a theory of "literal infringement." "To establish literal infringement, every limitation set forth in a claim must be found in an accused product, exactly." Advanced Steel Recovery, LLC v. X-Body Equipment, Inc., 808 F.3d 1313, 1319 (Fed. Cir. 2015). "Summary judgment of noninfringement is proper when no reasonable jury could find that every limitation recited in a properly construed claim is found in the accused device either literally or under the doctrine of equivalents." Id. at 1317. "The absence of even a single limitation of [the asserted claims] from the accused device precludes a finding of literal infringement." Kahn v. Gen. Motors Corp., 135 F.3d 1472, 1477 (Fed. Cir. 1998).

1.

Within News Feed, Mirror Worlds alleges that the Multifeed System (the backend infrastructure for News Feed) is the "computer system," and that the Multifeed Leaves (a storage system for keeping track of recent user actions) is the "main stream." Accordingly, to prove infringement, Mirror Worlds must

show that the Multifeed Leaves includes every data unit received or generated by the Multifeed System. Facebook argues that it is entitled to summary judgment of non-infringement because the record conclusively establishes that the Multifeed System contains data units that are not contained in the Multifeed Leaves. Facebook bears the burden at this stage because it is the party moving for summary judgment.

Facebook's theory of non-infringement with respect to News Feed focuses primarily on the Multifeed Aggregator. The Multifeed Aggregator is part of the Multifeed System. Mao Decl., ECF No. 250-16 ¶¶ 6-7. Facebook argues that the Multifeed Aggregator receives several categories of information that are not stored in the Multifeed Leaves. The Court will address these categories of information in turn.

Coefficient

Facebook argues that the Multifeed Aggregator receives information known as "coefficient scores" from the "Coefficient service." ECF No. 241, at 15. "A coefficient score provides a numerical weight that describes the strength of the relationship between the user and a particular friend." Mao Decl. ¶ 12. "The [Multifeed] Aggregator uses these coefficient scores to potentially boost candidate stories relating to friends with whom the user has a closer relationship." Id. Yun Mao, an

Engineering Director at Facebook who leads the Feed and Stories Infrastructure Team, unequivocally testified that coefficient scores are received by the Multifeed Aggregator, and that coefficient scores "are not stored in or received from the [Multifeed] Leaves." Id.

Mirror Worlds responds that "record evidence shows that the 'coefficient score' Facebook alleges is 'received' by the Multifeed Aggregator is not received as part of the write pathway for writing data to the Leaves." Mirror Worlds' RTSMF ¶ 44. In support of this proposition Mirror Worlds cites a broad statement from Gregory Marra, Facebook's director of product management: "I understand that leaves are where information from News Feed are stored." Id. (citing Tsuei Decl., Ex. 7, ECF No. 290-2, at 83). Because this general statement does not address coefficient scores or how data is written to the Multifeed Leaves or the Multifeed System, it is insufficient to create a dispute of material fact.

Mirror Worlds also cites Exhibits 27, 28, and 30 to the Tsuei Declaration. Id. Together, Exhibits 27 and 30 demonstrate that the "Tailer" is used to write data to the Multifeed Leaves. See Tsuei Decl., Ex. 27, ECF No. 290-19, at FBMW_00118300 ("[The Tailer] is the input data pipeline that gets user actions to the storage layer [the Leaves] in real time."); id., Ex. 30, ECF No. 290-22, at FBMW_00136009 (diagram showing "Tailers" writing into

"Feed Leaf"). Exhibit 28, ECF No. 290-20, at 1, does not discuss the Tailer in particular, but states that "When a user takes any action on Facebook, there are two independent pipelines that record this action (i.e. a UserAction) to the Leaves."

Mirror Worlds' argument appears to be that information cannot be in the Multifeed System (the alleged computer system) unless it is "received through the Tailer[]." ECF No. 288, at 18. This argument finds no support in the record. Facebook has offered specific evidence that the Multifeed Aggregator (part of the Multifeed System, the alleged computer system) receives data (coefficient scores) that is not contained in the Multifeed Leaves (the alleged main stream). Even if the record established that there is only one path for information to get into the Multifeed Leaves (which it does not, see Ex. 28; Balakrishnan Rpt., ECF No. 250-19 ¶ 204), it would not necessarily follow that there is only one path for information to get into the Multifeed System. Mirror Worlds points to no evidence demonstrating that the Tailer is the only pathway into the Multifeed System or the Multifeed Aggregator in particular. Accordingly, Mirror Worlds' attempt to show that there is only one data pipeline into the Multifeed Leaves, even if successful, would not create a dispute of material fact.

Mirror Worlds likewise misses the mark by arguing that "Facebook has presented no evidence that the information

identified in its motion . . . is user data received by the Tailer[]." ECF No. 288, at 19. Facebook does not need to show that the data it points to is received by the Tailer to prove non-infringement. Facebook need only prove that there is data in the Multifeed System that is not contained in the Multifeed Leaves. Moreover, Mirror Worlds' insistence that Facebook point specifically to "user data" that is contained in the alleged computer system but not the alleged main stream is misleading. The patents are clear that the main stream must contain every data unit (of whatever type) that is received or generated by the computer system. See, e.g., '227 patent, col. 16, lines 9-10 ("A method which organizes each data unit received by or generated by a computer system"); id., col. 4, lines 8-10 ("Every document created and every document sen[t] to a person or entity is stored in a main stream."); ECF No. 109-1, at FBMW_00097563 ("A 'data unit' is a 'document' because a 'document can contain any type of data.'").[16] As Facebook correctly notes, "That the Multifeed Leaves store user actions (e.g., provided by the Tailer[]) is in no way inconsistent with the Multifeed Aggregator receiving additional information beyond

[16] See also '538 patent, claim 1 (reciting a "main stream"); '439 patent, claim 1 (reciting a "main collection," which the Court has construed as "main stream"); id., abstract ("main stream"); Omega Eng'g, Inc. v. Raytek Corp., 334 F.3d 1314, 1334 (Fed. Cir. 2003) ("[W]e presume, unless otherwise compelled, that the same claim term in the same patent or related patents carries the same construed meaning.").

user actions that is not stored in the Multifeed Leaves." ECF No. 300, at 4-5.[17]

Finally, Mirror Worlds argues that coefficient scores being included in the Multifeed Aggregator but not the Multifeed Leaves does not prove non-infringement because coefficient scores are "queries," and therefore not "data units" within the meaning of the asserted patents. See ECF No. 288, at 15-16, 19-20. First, Mirror Worlds has presented no evidence that coefficient scores should be considered "queries." Even if a query is involved in getting coefficient scores from the Coefficient service to the Multifeed Aggregator, the record is clear that a coefficient score is an item of information that the Multifeed Aggregator "receives and relies on." Mao Decl. ¶ 12. The coefficient score is not simply a request made to or from the Multifeed Aggregator. Moreover, Mirror Worlds' argument fails under the Court's construction of "data unit." A query is plainly "an item of information," which is also consistent with Mirror Worlds' argument that "data unit" includes any type of data.

[17] Mirror Worlds' "user data" argument also fails to the extent it is based on Mirror Worlds' construction of the term "data unit." First, for the reasons explained above, "data unit" does not contain an "of direct user interest" limitation. Second, Mirror Worlds explicitly argues in its claim construction brief that "data unit" should be construed to clarify that "data unit" can refer to data in any format (that is, textual or otherwise). The Court accepted that argument in construing "data unit" to mean "an item of information." Accordingly, there is no subject matter or format limitation built into the term "data unit."

Facebook has established that the Multifeed Aggregator (part of the Multifeed System, the alleged computer system) receives coefficient scores, which are items of information that are not contained in the Multifeed Leaves. Mirror Worlds has offered no evidence to the contrary. Accordingly, the News Feed system does not infringe the asserted patents, and Facebook is entitled to summary judgment of non-infringement with respect to News Feed.

AdFinder

Facebook also argues that the Multifeed Aggregator receives information from a system called "AdFinder," and that this information is not stored in the Multifeed Leaves. ECF No. 241, at 15-16; Mao Decl. ¶¶ 9-10, 16; Facebook's Statement of Material Facts ("Facebook's SMF"), ECF No. 241-1 ¶¶ 47-48; Balakrishnan Rpt. ¶¶ 226, 229, 544-45, 563-64.

AdFinder generates advertisements relevant to a particular user and provides those ads to the Multifeed Aggregator. Mao Decl. ¶ 16. Mr. Mao declared:

> The interactions between AdFinder and the Multifeed Aggregator involve the exchange of hundreds of pieces of information that did not come from (and were not stored in) the Multifeed Leaves, but other separate and distinct storage systems within Facebook. The candidate stories generated by the Multifeed Aggregator corresponding to these advertisements thus incorporate information that came from AdFinder and that was not stored in the Multifeed Leaves.

Id. Accordingly, there is unequivocal evidence in the record that the Multifeed Aggregator receives information from AdFinder that is not contained in the Multifeed Leaves.

Mirror Worlds cites no evidence to the contrary. Mirror Worlds cites deposition testimony from Facebook engineer Mainak Mandal, Mirror Worlds' RTSMF ¶ 47[18] (citing Tsuei Decl., Ex. 15, ECF No. 290-9, at 107-08), but this testimony supports Facebook. Mr. Mandal testified that there are two types of stories in a user's feed—"ad stories" and "organic stories." Tsuei Decl., Ex. 15, at 107-08. He further testified that the organic stories ("any story that is not an ad") "come from Multifeed Leaves," while the ad stories are "generated on AdFinder." Id. at 108. This testimony supports Facebook's argument that the Multifeed Aggregator contains information from AdFinder that is not stored in the Multifeed Leaves.[19]

The remainder of Mirror Worlds' response to Facebook's Statement of Material Fact regarding AdFinder is identical to Mirror Worlds' response concerning Coefficient service. Compare

[18] Mirror Worlds' RTSMF ¶¶ 47 and 48, which both address AdFinder, are identical in substance.

[19] Mirror Worlds next cites deposition testimony from Mr. Mao. Mirror Worlds' RTSMF ¶ 47 (citing Tsuei Decl., Ex. 48, ECF No. 290-35, at 41). This testimony supports the distinction Mr. Mandal drew between ad stories and organic stories. See Tsuei Decl., Ex. 48, at 41 ("[organic content is] actually not—not from the ads bracket"). Mirror Worlds also cites the deposition testimony of Facebook's expert, Dr. Ravin Balakrishnan. RTSMF ¶ 47 (citing Tsuei Decl., Ex. 17, ECF No. 290-11, at 197-200). This testimony does not help Mirror Worlds because Dr. Balakrishnan simply reiterates that ads are not "organic content." Tsuei Decl., Ex. 17, at 199-200.

Mirror Worlds' RTSMF ¶ 47, with id. ¶ 44. The sources that are duplicated from ¶ 44 fail to create a dispute of material fact for the same reasons explained with respect to Coefficient service: the sources do not address AdFinder in particular, and Facebook does not need to show that the information from AdFinder is "received as part of the write pathway for writing data to the Leaves," id. ¶ 47, to prove non-infringement. The thrust of Facebook's argument is that the information from AdFinder is not contained in the Leaves.

Also for the reasons explained with respect to Coefficient service, Mirror Worlds' argument that the information the Multifeed Aggregator receives from AdFinder does not consist of "data units" fails. Under the Court's construction of "data unit," there is no "of direct user interest" limitation, and it is plain that the Multifeed Aggregator receives items of information from AdFinder.[20] Accordingly, Facebook has established that the Multifeed Aggregator receives information from AdFinder that is not stored in the Multifeed Leaves.

Ego

Ego is a Facebook system that "generates recommendations, for example, identifying other Facebook users whom the current

[20] Mirror Worlds does not argue, as it did with respect to coefficient scores, that information from AdFinder is merely part of a query, and therefore not a data unit. See ECF No. 288, at 19.

user may wish to become friends ('People You May Know'), or Groups or Pages that might be of interest to the current user." Mao Decl. ¶ 17. As with information from AdFinder, Mr. Mao declares: "The recommendation information provided by Ego to the Multifeed Aggregator did not come from (and are not stored in) the Multifeed Leaves, and thus, the candidate stories generated by the Multifeed Aggregator to reflect those recommendations incorporate information that was not stored in the Multifeed Leaves." Id.; see also Facebook's SMF ¶ 49; Balakrishnan Rpt. ¶¶ 236, 540, 563, 565.

Mirror Worlds cites no evidence to the contrary. Indeed, Mirror Worlds' RTSMF ¶ 49 is identical to ¶ 47 (the response concerning AdFinder), except the words "information from Ego" replace the words "information from AdFinder." Mirror Worlds cites no record evidence to dispute that the Multifeed Aggregator receives information from Ego that is not stored in the Multifeed Leaves. For the reasons explained above, Mirror Worlds' arguments that (1) Facebook has not shown that information from Ego is received "as part of the write pathway for writing data to the Leaves," Mirror Worlds' RTSMF ¶ 49, and (2) the information from Ego is not a "data unit" (either because it is part of a query or because it is not "of direct user interest") also fail. Accordingly, Facebook has established

that the Multifeed Aggregator receives information from Ego that is not stored in the Multifeed Leaves.

TAO, Recent Interactions, and ReadState

Facebook also argues that user information from TAO, as well as recent interaction information and information called the "ReadState," is received by the Multifeed Aggregator but not contained in the Multifeed Leaves. However, there are issues of fact that preclude these categories of data from being a basis for summary judgment of non-infringement.

TAO is a "data store that provides access to objects and their associations with other objects." Bronson Decl., ECF No. 123 ¶ 20.[21] Facebook argues that three categories of user information that are not stored in the Multifeed Leaves are received by the Multifeed Aggregator from TAO: "(1) a list of 'Friends' of the user for whom the News Feed is being prepared; (2) the list of 'Pages' on Facebook that the user has liked, and (3) the list of 'Groups' the user has joined." ECF No. 241, at 14. Mirror Worlds, however, points to a Facebook document concerning the Multifeed System that states: "A user's friends, pages, liked, and other actions are also stored in the leaves."

[21] "Generally, the content Facebook users see is an amalgamation of 'objects' and 'associations,' which are two classes of data. Users, pictures, and comments are types of objects, while associations describe the relationship between objects. For example, if user 'Alice' posts a comment on Facebook, an 'authorship' association would connect Alice and the comment." Mirror Worlds Techs., 800 F. App'x at 904.

Tsuei Decl., Ex. 27, at FBMW_00118300. The same document states that the Multifeed Leaves "indexes all user actions." Id. (emphasis added); see also id. ("all of a user's recent Actions and Objects will be indexed"). This record evidence is sufficient to create a dispute of material fact as to whether lists of the user's friends, pages liked, and groups joined are stored in the Multifeed Leaves.

Facebook also argues that the Multifeed Aggregator receives "recent interaction information" ("such as posts the user recently liked or videos the user recently watched") that is not stored in the Multifeed Leaves. ECF No. 241, at 15; Mao Decl. ¶ 15. But, as explained in the previous paragraph, Mirror Worlds cites record evidence to the effect that the Multifeed Leaves indexes "all user actions," which is sufficient to raise a dispute of material fact as to whether recent interaction information is stored in the Multifeed Leaves.

Finally, Facebook argues that the Multifeed Aggregator receives information called the "ReadState," "which informs the Aggregator which stories were already recently presented to the user." ECF No. 241, at 15; see also Mao Decl. ¶ 14. There is record evidence that ReadState indexes all stories that a user has seen along with those stories' "respective timestamps and view duration." Tsuei Decl., Ex. 30, at FBMW_00136001. What stories a user views, and for how long a user views those

stories, could plausibly be considered "user actions," which record evidence suggests are indexed in the Multifeed Leaves. Any ambiguity regarding whether "all user actions" includes the Readstate information must be resolved in favor of Mirror Worlds at this stage.

Accordingly, summary judgment of non-infringement cannot be granted based on (1) the data that Facebook alleges is sent to the Multifeed Aggregator by TAO, (2) recent interaction information, or (3) ReadState.

In sum, with respect to News Feed, while Facebook is not entitled to summary judgment of non-infringement based on data received from TAO, recent interaction information, or ReadState, Facebook is entitled to summary judgment of non-infringement on three independent bases: the record conclusively establishes that the Multifeed Aggregator (part of the Multifeed System, the alleged computer system) receives information from (1) Coefficient service, (2) AdFinder, and (3) Ego that is not stored in the Multifeed Leaves (the alleged main stream).

2.

Mirror Worlds alleges that two Facebook features in addition to News Feed infringe the asserted patents: Timeline and Activity Log. The parties treat Timeline and Activity Log together because they are supported by the same backend infrastructure. See ECF No. 241, at 16-17; ECF No. 288, at 12

n.6; Tang Decl. ¶ 2. The Court will likewise discuss these two features together. For purposes of Timeline and Activity Log, Mirror Worlds alleges that the Timeline backend system is the computer system, and that the TimelineDB is the alleged main stream. See Facebook's SMF ¶¶ 36-37; Mirror Worlds' RTSMF ¶¶ 36-37. Accordingly, to prove infringement, Mirror Worlds must show that the TimelineDB includes every data unit received or generated by the Timeline backend system.

The TimelineDB is "a database used within Facebook to keep track of certain types of actions taken by users on Facebook." Tang Decl. ¶ 9. The "Timeline Aggregator" is part of the Timeline backend system. Id. ¶ 7. The Timeline Aggregator, which is used by both the Timeline and Activity Log features, "deliver[s] a list of user engagements (or user actions) that pertain to a particular user." Id. ¶ 8. Similar to its argument regarding News Feed, Facebook argues that the Timeline Aggregator (part of the alleged computer system) receives information that is not contained in the TimelineDB (the alleged main stream). The Court will address each disputed set of information in turn.

Background User Information

Facebook argues that the Timeline Aggregator receives "extensive information about the user for whom the Timeline or

Activity Log pertains including the user's birthday, list of family members, list of children, list of places previously lived, education history, among other information," from "at least two different storage systems"—TAO and UDB (user database). ECF No. 241, at 16. The record contains unequivocal evidence that this background user information is received by the Timeline Aggregator and is not stored in the TimelineDB. Yifei Tang, an engineering manager at Facebook on the Timeline team, declared:

> The Timeline Aggregator component implements a complex process that requires the Aggregator to receive and rely on an extensive amount of information that is not stored in the TimelineDB. For example, the "front end" . . . constructs a request to the Timeline Aggregator that supplies the Aggregator with a number of pieces of information relating to the user in question, including the user's birthday, list of family members, list of children, list of places previously lived, education history, along with other information about that user. This information received by the Aggregator is not stored in the TimelineDB but was instead obtained from "TAO" and "UDB" (user database), storage systems that are separate and distinct from the TimelineDB.

Tang Decl. ¶ 10; see also Balakrishnan Rpt. ¶¶ 307-09.

Mirror Worlds attempts to create an issue of fact with respect to this clear evidence, but none of its attempts are sufficient. Mirror Worlds cites various exhibits that purportedly stand for the propositions that the TimelineDB "is a backend system that persists all actions by users and pages and indexes them chronologically" and that "everything relating to

[the user]" is contained in the TimelineDB. See ECF No. 288, at 4, 12-13; Koskinen Decl., ECF No. 145 ¶ 74. But the documents cited for these propositions actually refer to "Timeline" (the backend system writ large), not the TimelineDB. See Tsuei Decl., Ex. 37, ECF No. 290-29, at FBMW_00007832; id., Ex. 35, ECF No. 290-27, at FBMW_00098325. Counsel for Mirror Worlds and Mirror Worlds' expert, Dr. Eric Koskinen, simply changed the subjects of the quoted sentences.[22] Because these documents do not describe the TimelineDB, they do not stand for the proposition that the TimelineDB contains all background user information contained in the Timeline backend system.

Mirror Worlds also cites evidence that there is a path from TAO to the TimelineDB. See id., Ex. 24, ECF No. 290-16, at FBMW_00000211. But this does not establish that all data from TAO is contained in the TimelineDB; nor does it establish that background user information in particular (birthday, list of family, places lived, etc.) is stored in the TimelineDB. And while Mirror Worlds points to snippets of the testimony of

[22] In other portions of their papers, counsel for Mirror Worlds and Dr. Koskinen demonstrate an understanding of this distinction. See Mirror Worlds' RTSMF ¶ 36 ("Mirror Worlds contends that the 'computer system' for its theory of infringement by Facebook's Timeline and Activity Log features is the 'Timeline Backend' system."); id. ¶ 37 ("Mirror Worlds contends the TimelineDB is a 'main stream[.]'"); ECF No. 288, at 4 (arguing that "the Timeline Backend system stores in the TimelineDB all user data that may appear in Timeline or Activity Log in time-order"); Koskinen Decl. ¶ 26 ("The Timeline backend system includes the "TimelineDB"). This distinction was also made clear by the court of appeals. See Mirror Worlds Techs., 800 F. App'x at 904 ("The Timeline back-end system includes the TimelineDB database and an Aggregator.").

Jeffrey Huang, Director of Engineering at Facebook, none of those snippets contradict the clear testimony of Mr. Tang that the Timeline Aggregator receives background user information that is not contained in the TimelineDB. Tang Decl. ¶ 10.

For example, Mirror Worlds cites testimony from Mr. Huang that casts some doubt as to whether TAO is properly referred to as a storage system. E.g., Huang Depo., ECF No. 290-4, at 37 (TAO is "an interface to fetch data, but I wouldn't necessarily say that it is a way to store the data."). However, whether TAO is properly referred to as a storage system is not a material fact. "Only disputes over facts that might affect the outcome of the suit under the governing law will properly preclude the entry of summary judgment." Anderson, 477 U.S. at 248. Facebook has adduced clear evidence that the Timeline Aggregator receives background user information that is not stored in the TimelineDB. Mirror Worlds points to no evidence to the contrary. It is not material whether TAO (1) stores the background user information and delivers that information to the Timeline Aggregator, or (2) is an interface that is responsible for sending the background user information to the Timeline Aggregator.[23] Accordingly, this dispute is one of terminology and does not affect the outcome of the non-infringement dispute.

[23] Nathan Bronson, a Facebook engineer who helped implement TAO, explained that, "when News Feed, Timeline, Activity Log [and other features] seek to retrieve objects such as photos and comments from Facebook's data store, it

Mirror Worlds also cites a diagram illustrating a portion of the accused system, and emphasizes that the diagram "show[s] no interaction between UDB, TAO or Coefficient with the Aggregator." ECF No. 288, at 14 (citing Tsuei Decl., Ex. 26, ECF No. 290-18, at FBMW_00014972). But this diagram does not appear to depict TAO at all. It is undisputed that TAO exists, and the record is clear that the Timeline and Activity Log features could not function without TAO. See, e.g., Huang Depo. at 128 ("[S]o in any form of rendering Timeline or Activity Log you have to have TAO."). Moreover, when counsel for Mirror Worlds confronted Mr. Huang at his deposition with diagrams such as this one that do not explicitly reference TAO, Mr. Huang explained (1) that references to data sources that TAO can fetch from, such as UDB, could be understood as references to TAO; and (2) that the diagrams alluded to focus on specific aspects of the Timeline service that do not include the role of TAO, see id. at 128 ("[TAO is] not part of this diagram because these are very infra, very backend heavy documents and diagrams. . . . just because it's not there doesn't mean it doesn't exist."); id. at 130 (A: "The diagrams that we've been looking at, again,

---

is TAO that retrieves and delivers an up-to-date version of those objects to the features." "In cases where the data associated with an object is very large, such as for photos and videos, an object is stored in TAO that contains enough information to fetch the full contents from a separate system specially optimized for that data type. For most objects, TAO stores all of the data directly . . . ." Bronson Decl. ¶¶ 4, 21-22. This description confirms that TAO acts both as a storage system and as a go-between that fetches data for the accused features from elsewhere.

are very zoomed in on a specific piece of --" Q: "Mr. Huang, it's a 'yes' or 'no' question."). Accordingly, the lack of interaction between the Timeline Aggregator and UDB or TAO in any particular diagram does not create a dispute of material fact.

Mirror Worlds also relies on statements by its expert, Dr. Koskinen. But those conclusions are not supported by the evidence in the record. See, e.g., Koskinen Rpt., ECF No. 279-1 ¶ 114 (stating without citing any evidence that the TimelineDB is a "main stream"). Moreover, Dr. Koskinen's statements rely on documents where Dr. Koskinen has simply changed the subject of the quoted sentence from "Timeline" to "the TimelineDB." Compare, e.g., id. ¶ 115 ("TimelineDB is a 'time-ordered index for all time[.]'"), with Tsuei Decl., Ex. 54, ECF No. 290-37, at slide 5 ("Timeline: Time-ordered index for all time"). Dr. Koskinen's conclusory statements are not supported by the underlying documents and are therefore not entitled to any weight.

Mirror Worlds appears to argue that data cannot enter the Timeline backend system (the accused computer system) unless that data goes through what Mirror Worlds asserts is the designated path for writing data into the TimelineDB (the

accused main stream).[24] This argument, like Mirror Worlds' argument concerning the Tailer and the Multifeed System, finds no support in the record. Facebook has presented evidence that data is received by the Timeline Aggregator that is not contained in the TimelineDB. E.g., Tang Decl. ¶ 10. The testimony of Mr. Huang (and other sources relied on by Mirror Worlds) only speak to how data is written into the TimelineDB. But even if Mirror Worlds could demonstrate that there is only one pathway for writing data into the TimelineDB, Mirror Worlds has presented no evidence that there is only one pathway for writing data into the Timeline backend system or the Timeline Aggregator in particular. To prove non-infringement, Facebook does not need to show that the data that is included in the Timeline Aggregator is received as part of the pathway for writing data into the TimelineDB. Rather, Facebook need only show that data received by the Timeline Aggregator is not included in the TimelineDB, and Facebook has done so. Accordingly, Mirror Worlds' argument concerning the write pathway to the TimelineDB fails to create a dispute of material fact.

[24] See ECF No. 288, at 14 ("Timeline utilizes a write path to store 'user data'—to 'persist activities'—in the TimelineDB, and a separate read path to 'get activities.' Writing user data to the backend requires specific function calls. Facebook has presented no evidence that any of its newly-identified information is subject to any such function call . . . ."); Mirror Worlds' RTSMF ¶ 51 (emphasis added) ("[R]ecord evidence shows that the information . . . Facebook alleges is 'received' by the Timeline Aggregator is not received at all, or received as part of the write pathway for writing data to the TimelineDB.").

The court of appeals found that summary judgment was not warranted on the record of the first summary judgment motion with respect to TAO and the Timeline and Activity Log features. See 800 F. App'x at 908-09. Most of the court of appeals' discussion of this issue focused on Facebook's failure to provide definitive evidence that the Timeline Aggregator receives the asserted evidence from TAO. That defect in Facebook's argument has been cured because there is now unequivocal evidence that the Timeline Aggregator receives background user information from TAO. See Tang Decl. ¶ 10; Balakrishnan Rpt. ¶¶ 307-09.

The court of appeals also highlighted evidence suggesting that the frontend, rather than the Timeline Aggregator, receives the data from TAO referenced by Facebook. See 800 F. App'x at 909. The court of appeals cited Mr. Huang's testimony that "it's really the web tier that I consider the frontend that then, you know, needs to go to TAO to actually fetch any of the content." Id. (quoting Huang Depo., at 29-30). In light of the fully-developed record, this testimony does not create a dispute of material fact. That the frontend queries TAO, and that the frontend receives data from TAO, is not inconsistent with the Timeline Aggregator receiving background user information from TAO. Mr. Huang did not testify that the frontend is the only destination to which TAO sends data. Moreover, there is record

evidence specifically establishing that TAO sends background user information to the Timeline Aggregator. See Tang Decl. ¶ 10; Balakrishnan Rpt. ¶¶ 307-09. Finally, this testimony does not reflect at all on the relationship between UDB and the Timeline Aggregator. Because Facebook has provided evidence that background user information that is not stored in the TimelineDB is received by the Timeline Aggregator from both TAO and UDB, this testimony does not create a dispute of material fact as to whether the Timeline Aggregator receives background user information that is not contained in the TimelineDB.

As it argued with respect to the News Feed feature, Mirror Worlds argues that, even if the Timeline Aggregator receives background user information that is not contained in the TimelineDB, that would not establish non-infringement because background user information does not consist of "data units." Mirror Worlds' RTSMF ¶ 51. This argument fails. The background user information consists of items of information.

Accordingly, the record establishes that the Timeline Aggregator receives background user information that is not contained in the TimelineDB. Facebook is entitled to summary judgment of non-infringement with respect to the Timeline and Activity Log features on this basis.

Major Life Events

Facebook also argues that the Timeline Aggregator receives a list of a user's major life events, such as graduations, marriages, or moving to a new city. See ECF No. 241, at 17; Facebook's SMF ¶ 52. Facebook points to unequivocal evidence that this major life event information is received by the Timeline Aggregator from TAO or UDB, and that this data is not stored in the TimelineDB. Tang Decl. ¶ 11; see also Balakrishnan Rpt. ¶¶ 254, 256-57, 307, 317, 337.

Mirror Worlds' RTSMF ¶ 52 is identical to its RTSMF ¶ 51, other than replacing background user information with "major life events." Accordingly, for all the reasons explained in connection with background user information, Mirror Worlds has failed to create a dispute of material fact as to whether the Timeline Aggregator receives major life event information that is not contained in the TimelineDB. None of the sources cited by Mirror Worlds in its RTSMF ¶ 52, on their own or taken together, establish either that the TimelineDB contains major life event information, or that the Timeline Aggregator does not receive major life event information. Mirror Worlds' argument that this information does not consist of "data units" also fails because this information meets the Court's construction of "data unit."

Therefore, Facebook is entitled to summary judgment of non-infringement with respect to Timeline and Activity Log for the

independent reason that it has proven beyond reasonable dispute that the Timeline Aggregator receives major life event information that is not contained in the TimelineDB.

Coefficient

Finally, Facebook argues that the Timeline Aggregator receives information from the Coefficient service (discussed above with respect to News Feed) that is not contained in the TimelineDB. See ECF No. 241, at 16-17; Facebook's SMF ¶ 53. Facebook again points to unequivocal testimony that this information, which is used by the Timeline Aggregator to identify a user's degree of engagement with other users, is not stored in the TimelineDB. Tang Decl. ¶ 12; see also Balakrishnan Rpt. ¶¶ 246, 300-01, 307-09, 313.

Mirror Worlds' RTSMF ¶ 53 is identical to its RTSMF ¶¶ 51-52, except for inserting the phrase "coefficient score" in place of background user information or major life events. Accordingly, for all the reasons explained in connection with background user information, Mirror Worlds has failed to create a dispute of material fact as to whether the Timeline Aggregator receives data from the Coefficient service that is not contained in the TimelineDB. Mirror Worlds' argument that this information does not consist of "data units" fails here as well.

Therefore, Facebook is entitled to summary judgment of non-infringement with respect to Timeline and Activity Log on the independent basis that the Timeline Aggregator receives data from the Coefficient service that is not contained in the TimelineDB.

To sum up regarding Timeline and Activity Log: Facebook is entitled to summary judgment of non-infringement with respect to these two features because Facebook has proven that the Timeline Aggregator receives the following data that is not contained in the TimelineDB: (1) background user information; (2) major life event information; and (3) coefficient data. Each of these sets of information constitutes an independent basis entitling Facebook to summary judgment of non-infringement with respect to Timeline and Activity Log.

3.

The absence of a "main stream" and "main collection"[25] in the accused features means that Facebook is entitled to summary judgment of non-infringement with respect to all the asserted claims. Facebook also urges an additional ground of non-infringement with respect to the asserted claims of the '538 and '439 patents: namely, that none of the accused Facebook features contain a "glance view" limitation.

[25] The '227 patent and the '538 patent explicitly recite a main stream limitation, see '227 patent, claim 13; '538 patent, claim 1. The '439 patent contains a "main collection" limitation, which the Court has construed to be equivalent to a main stream limitation. See '439 patent, claim 1.

Claim 1 of the '538 patent and claim 1 of the '439 patent each contain a "glance view" limitation. See '538 patent, col. 16, lines 55-60; '439 patent, col. 17, lines 34-41. The patents explain that the glance view is intended to "make searching for and working with a document more intuitive." '538 patent, col. 3, lines 44-45; see also '439 patent, col. 3, lines 46-47. The glance view provides a preview of a document to a user who is scrolling through a stream of documents, "such as a thumbnail image of the first page of the document, a WAV or MP3 preview of an audio file, [] an animated GIF preview of a video file," or "text summaries." '538 patent, col. 7, lines 20-26; see also id., col. 2, lines 62-63; '439 patent, col. 7, lines 23-29. Claim 1 of both patents requires the glance view to be "an abbreviated version of the document" being depicted and "indicative of content thereof." '538 patent, col. 16, lines 59-60; '439 patent, col. 17, lines 39-41.[26] The patents also explain that the glance view appears in response to "touching with a cursor a screen area associated with the document." '538 patent, col. 16, lines 57-58; see also '439 patent, col. 17, lines 42-47.

Mirror Worlds contends that the glance view limitation of the asserted claims of the '538 and '439 patents is satisfied by

[26] Mirror Worlds says this "summary" limitation only appears in the specification, ECF No. 288, at 22, but that is plainly not true. Mirror Worlds simply ignores the claim language that says the glance view must be "an abbreviated version of the document" and "indicative of content thereof."

a Facebook functionality in which the user can "hover" the user's cursor over a profile link (such as a link associated with a Page on Facebook). Facebook's SMF ¶ 61; Mirror Worlds' RTSMF ¶ 61. Facebook has offered evidence demonstrating that this hover functionality does not meet the glance view limitation. See Tang Decl. ¶¶ 15-19. Specifically, Facebook has shown that its hover functionality (or the "contextual dialog box," id. ¶ 14) provides only information about the source or author of the selected link, not information about the content of any underlying document. See id. ¶ 16. Facebook illustrates that two stories from the same author relating to completely different topics will yield the same contextual dialog box. See id. ¶¶ 17-18. Moreover, Mr. Tang declares that the content of the underlying story or document is not even provided to the source code responsible for creating the contextual dialog box. Id. ¶ 19. Accordingly, the contextual dialog box does not display—or even take into account—the content of the underlying document, and so it is not "an abbreviated version of the document" or "indicative of content thereof," '538 patent, col. 16, lines 59-60; '439 patent, col. 17, lines 39-41.

Mirror Worlds points to screenshots that allegedly illustrate that the contextual dialog box does provide information about the content of the underlying document. See ECF No. 288, at 23; Koskinen Rpt. ¶¶ 179, 184, 189, 194, 199,

361, 366, 371, 378, 384 (reproducing same three screenshots). Dr. Koskinen copied these screenshots from the Complaint. Id. ¶ 179. The Complaint alleges that these three screenshots came from third-party websites. Compl., at 30-32.

Facebook argues that all of Dr. Koskinen's testimony relying on these third-party materials should be excluded. The Court agrees. One of the screenshots was taken from a YouTube video that no longer exists and was never produced. See Balakrishnan Rpt. ¶ 443. Dr. Koskinen admits that he was not involved in creating any of the screenshots, and he testified that he could not tell when the images were taken, when they were archived, whether they had been cropped, or even what user action prompted the creation of the asserted glance view. See Koskinen Depo., ECF No. 227-4, at 116, 119-20, 127, 129, 131.

These screenshots are not independently admissible because they cannot be authenticated. See Fed. R. Evid. 901. Mirror Worlds offers no evidence to support a finding that these screenshots are what they purport to be. Instead, Mirror Worlds invokes Federal Rule of Evidence 703, which allows an expert to rely on inadmissible facts or data "[i]f experts in the particular field would reasonably rely on those kinds of facts or data in forming an opinion on the subject." It is plainly unreasonable for a technical expert to rely on unauthenticated, undated screenshots in forming an opinion. See Bd. of Trustees

of AFTRA Ret. Fund v. JPMorgan Chase Bank, N.A., No. 09-cv-686, 2011 WL 6288415, at *9-10 (S.D.N.Y. Dec. 15, 2011) (screenshot unreliable basis for expert opinion "where the expert does not know the source of the documents and cannot guarantee its reliability"). Relying on the screenshots in this case is especially unreasonable because Dr. Koskinen cannot verify what user action produced the asserted glance view, and Dr. Koskinen had access to the source code for the accused feature.[27]

Accordingly, Facebook's motion to exclude the opinions of Dr. Koskinen that rely on unauthenticated screenshots purporting to demonstrate the accused glance view functionality is **granted**. Mirror Worlds presents no admissible evidence to create a dispute of material fact as to whether Facebook's contextual dialog box indicates the content of the underlying story or document. Facebook's evidence establishes that its hover functionality does not meet the glance view limitation. Therefore, summary judgment of non-infringement as to the asserted claims of the '538 and '439 patents is appropriate on this basis as well.

---

[27] The cases cited by Mirror Worlds are not to the contrary. RMail Ltd. v. Amazon.com, Inc., No. 2:10-cv-258, 2019 WL 10375642, at *2 (E.D. Tex. June 12, 2019) involved a screenshot offered to prove notice of infringement, not a screenshot relied on by an expert to show infringement. The screenshots in ICON Internet Competence Network B.V. v. Travelocity.com LP, No. 3:11-cv-1131, 2013 WL 655024, *4 (N.D. Tex. Feb. 22, 2013) were offered to show prior disclosure of an expert's theory. The screenshots in DE Techs., Inc. v. Dell, Inc., No. 7:04-cv-628, 2007 WL 1112406, at *29 (W.D. Va. Apr. 9, 2007) were offered to illustrate the functionality of a prior art system. None of these cases involve a challenge to an expert's use of unauthenticated screenshots to demonstrate infringement.

V.

The Court need not reach the remainder of the parties' disputes. Because the Court concludes that there was no infringement of the asserted patents as a matter of law, this was necessarily not an "egregious" case of willful infringement warranting enhanced damages. See Halo Elecs., Inc. v. Pulse Elecs., Inc., 579 U.S. 93, 104 (2016). The Court need not reach the remainder of Facebook's motion to exclude certain opinions of Dr. Koskinen, ECF No. 223 (that is, all of the motion except the section regarding "glance view") because the remaining opinions are not relevant to the Court's summary judgment ruling. The Court does not need to reach Facebook's motion to exclude the opinions of Mr. Bergman, ECF No. 228, because those opinions relate only to damages. The Court need not reach Mirror Worlds' motion for partial summary judgment of no invalidity based on Facebook's prior art defenses, ECF No. 233, because the Court grants Facebook's motion for summary judgment of non-infringement. The Court need not reach Mirror Worlds' motion to exclude certain opinions of Facebook's expert witnesses, ECF No. 236, because the Court did not need to consider (1) Facebook's invalidity theories based on prior art, or (2) Dr. Balakrishnan's opinions regarding the summary of Mirror Worlds' prior infringement accusations in deciding Facebook's motion for summary judgment. Finally, the Court need not reach Mirror

Worlds' motion to strike certain opinions of Mr. Bokhart, ECF No. 246, because those opinions relate only to damages.

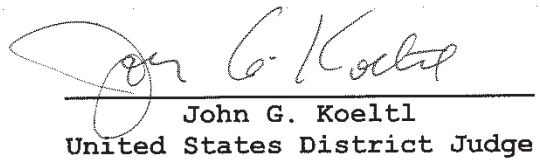
CONCLUSION

The Court has considered all of the arguments raised by the parties. To the extent not specifically addressed above, the arguments are either moot or without merit. For the reasons explained above, Facebook's motion for summary judgment is **granted in part and denied in part.** The Court concludes that the asserted claims are patent-eligible under § 101, and that there was no infringement in this case as a matter of law. The evidence establishes that no reasonable juror could find that Facebook infringed the patents at issue.

The Clerk is directed to enter judgment dismissing this case. The Clerk is also directed to close all pending motions and to close this case.

SO ORDERED.

Dated: New York, New York
March 2, 2022



John G. Koeltl
United States District Judge

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

----------X

MIRROR WORLDS TECHNOLOGIES, LLC,

Plaintiff,

-against-

17 **CIVIL** 3473 (JGK)

JUDGMENT

FACEBOOK, INC.,

Defendant.

----------X

It is hereby **ORDERED, ADJUDGED AND DECREED:** That for the reasons stated in the Court's Opinion and Order dated March 8, 2022, the Court has considered all of the arguments raised by the parties. To the extent not specifically addressed, the arguments are either moot or without merit. Facebook's motion for summary judgment is granted in part and denied in part. The Court has concluded that the asserted claims are patent-eligible under § 101, and that there was no infringement in this case as a matter of law. The evidence establishes that no reasonable juror could find that Facebook infringed the patents at issue. Judgment is entered dismissing this case; accordingly, the case is closed.

Dated: New York, New York

March 8, 2022

RUBY J. KRAJICK

Clerk of Court

BY: K. Mango

Deputy Clerk

US006006227A

United States Patent [19]

Freeman et al.

[11] **Patent Number: 6,006,227**

[45] **Date of Patent: *Dec. 21, 1999**

[54] **DOCUMENT STREAM OPERATING SYSTEM**

[75] Inventors: **Eric Freeman**, Branford; **David H. Gelernter**, Woodbridge, both of Conn.

[73] Assignee: **Yale University**, New Haven, Conn.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **08/673,255**

[22] Filed: **Jun. 28, 1996**

[51] **Int. Cl.**[6] **G06F 17/30**

[52] **U.S. Cl.** **707/7**; 707/2; 707/102

[58] **Field of Search** 395/611, 616; 707/100, 102, 200, 2, 7

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,063,495	11/1991	MacPhail	395/761
5,140,676	8/1992	Langelaan	395/777
5,159,669	10/1992	Trigg et al.	395/357
5,402,526	3/1995	Bauman et al.	706/49
5,448,729	9/1995	Murdock	395/615
5,530,859	6/1996	Tobias, II et al.	395/551
5,535,063	7/1996	Lamming	360/4
5,625,818	4/1997	Zarmer et al.	395/615
5,649,182	7/1997	Reitz	707/7
5,835,129	11/1998	Kumar	348/15
5,890,177	3/1999	Moody et al.	707/511

OTHER PUBLICATIONS

Cowart, Mastering Windows 3.1, 1992, chapter 12, pp. 396–417, Jan. 1, 1992.

Gelernter, The Cyber–Road Not Taken, The Washington Post, Apr. 3, 1994, 9 pages.

Freeman, Lifestreams Project Home Page, 3 pages, 1994–1996, Jan. 1, 1994.

Freeman, Lifestreams for the Newton, Steve Mann's Developers Corner, Oct. 31, 1995, 5 pages.

Steinberg, Lifestreams, Wired 5.02, Feb. 28, 1997, 11 pages.

Freeman et al, Lifestreams: Organizing your Electronic Life, AAAI Fall Symposium, Al Applications in Knowledge Navigation and Retrieval, Nov. 30, 1995, Cambridge MA, 6 pages.

Getting Results with Microsoft Outlook, copyright Microsoft Corporation 1995–1996, pp. 28–29, Jan. 1, 1995.

Nelson, Theodor H., The Right Way to Think About Software Design, In The Art of Human–Computer Interface Design, Brenda Laurel, (Ed.), (1990): pp. 235–243.

Landsdale, M., The Psychology of Personal Information Management, Applied Ergonomics, (Mar. 1988): pp. 55–66.

Malone, Thomas W., How Do People Organize Their Desks? Implications For The Design Of Office Information Systems., ACM Transactions On Office Systems, vol. 1, No. 1 (Jan. 1983): pp. 99–112.

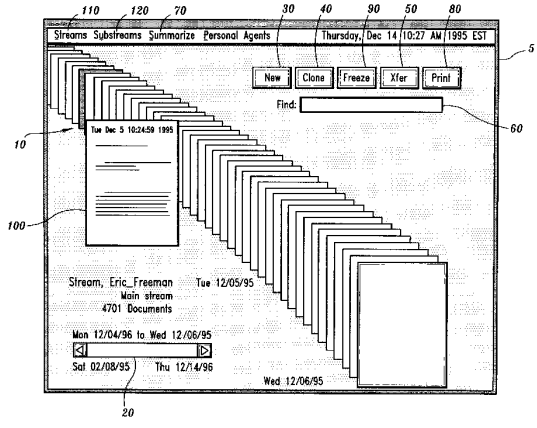
Primary Examiner—Wayne Amsbury

Attorney, Agent, or Firm—Cooper & Dunham LLP

[57] **ABSTRACT**

A document stream operating system and method is disclosed in which: (1) documents are stored in one or more chronologically ordered streams; (2) the location and nature of file storage is transparent to the user; (3) information is organized as needed instead of at the time the document is created; (4) sophisticated logic is provided for summarizing a large group of related documents at the time a user wants a concise overview; and (5) archiving is automatic. The documents can include text, pictures, animations, software programs or any other type of data.

33 Claims, 6 Drawing Sheets



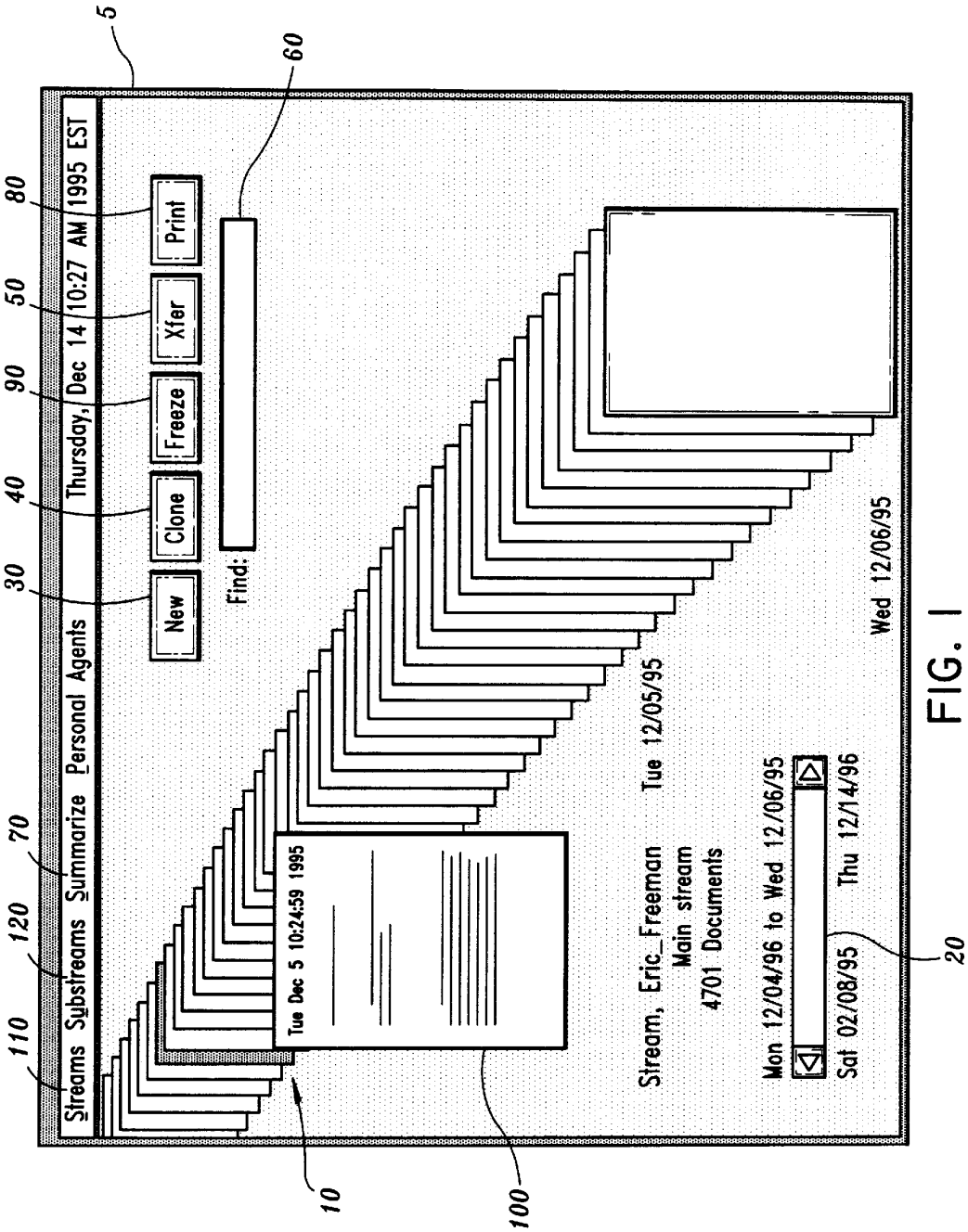

5
60
80
50
90
40
30
70
120
110
Thursday, Dec 14 10:27 AM 1995 EST
Print
Xfer
Freeze
Clone
New
Find:
Streams Substreams Summarize Personal Agents
Tue Dec 5 10:24:59 1995
Stream, Eric_Freeman Tue 12/05/95
Main stream
4701 Documents
Wed 12/06/95
Mon 12/04/96 to Wed 12/06/95
Sat 02/08/95 Thu 12/14/96
10
100
20

FIG. 1

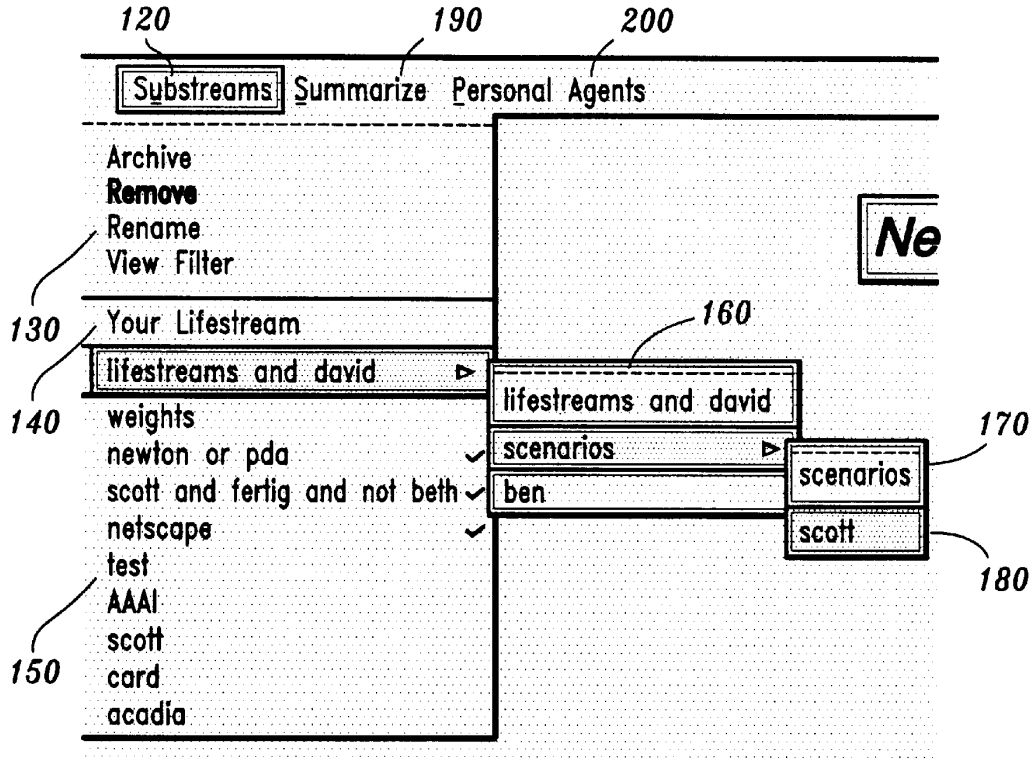

120
190
200
Substreams
Summarize
Personal Agents
Archive
Remove
Rename
View Filter
130
Your Lifestream
140
lifestreams and david
weights
newton or pda
scott and fertig and not beth
netscape
test
AAAI
scott
card
acadia
150
160
lifestreams and david
scenarios
ben
170
scenarios
scott
180

FIG. 2

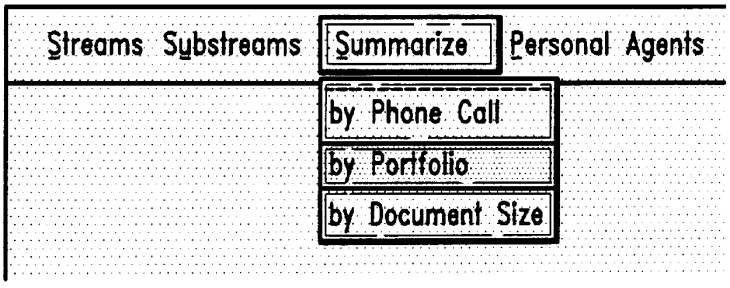

Streams
Substreams
Summarize
Personal Agents
by Phone Call
by Portfolio
by Document Size

FIG. 3

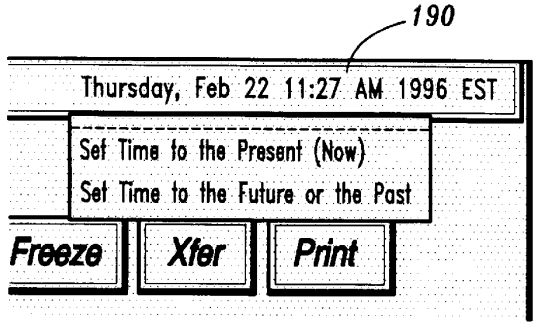

190
Thursday, Feb 22 11:27 AM 1996 EST
Set Time to the Present (Now)
Set Time to the Future or the Past
Freeze
Xfer
Print

FIG. 4

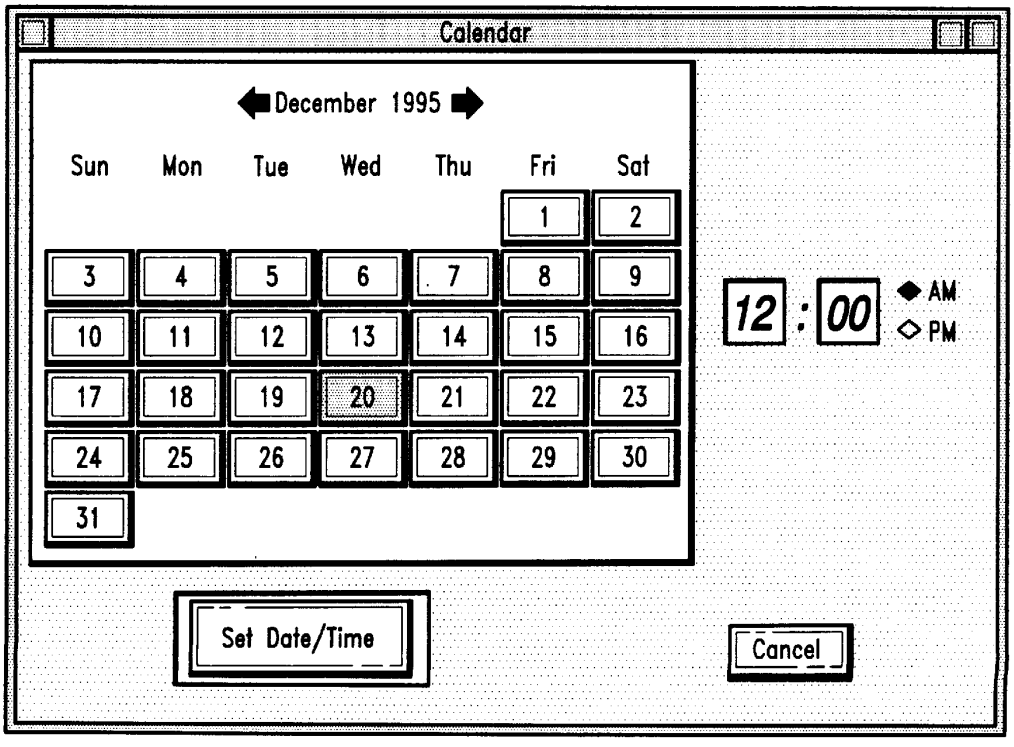

Calendar
December 1995
Sun Mon Tue Wed Thu Fri Sat
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
12 : 00
AM
PM
Set Date/Time
Cancel

FIG. 5

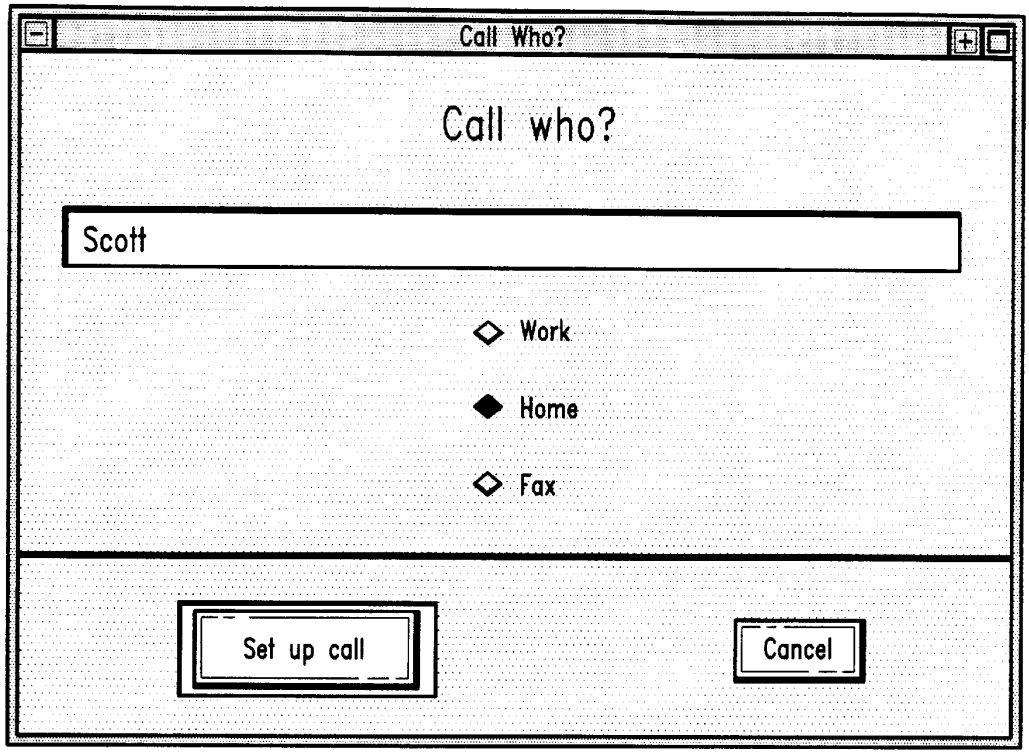


FIG. 6A

WHO	ON					AT	ABOUT
Scott Fertig	Tue	Aug	1	12:05	EDT 1995	432-6433	Port to PPC
Ward Mullins	Tue	Aug	1	11:57	EDT 1995	415 224-1912	Tcl/Java discussion
Beth Freeman	Tue	Aug	1	10:22	EDT 1995	432-1287	insurance

FIG. 6B

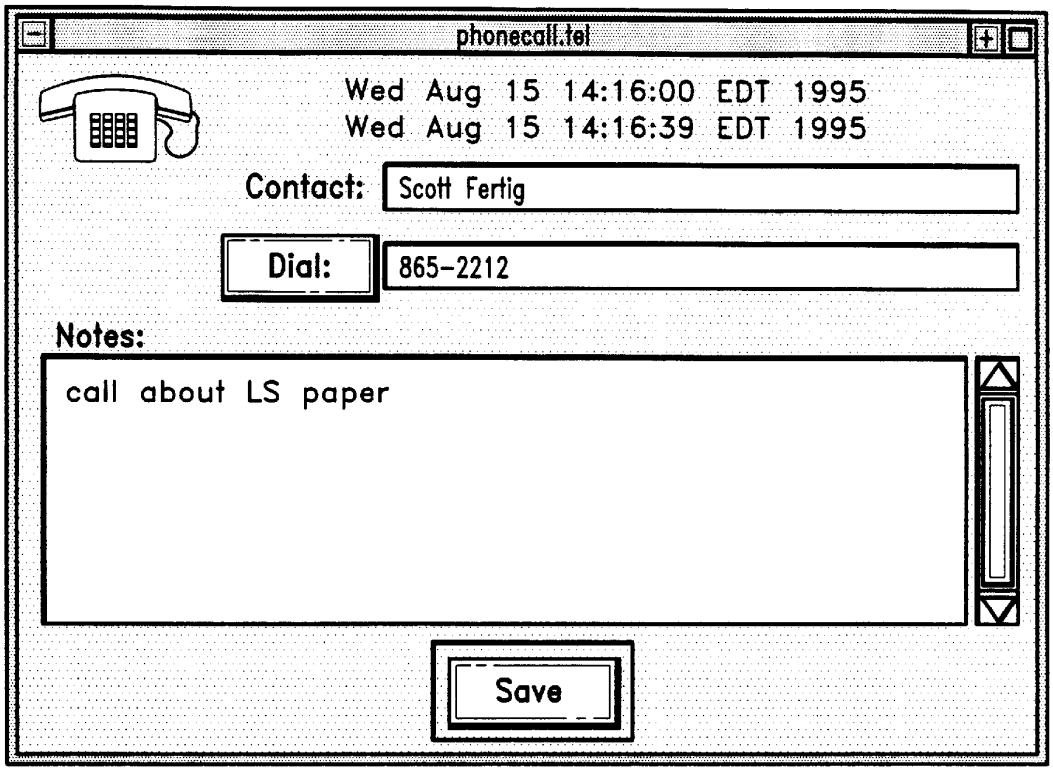


FIG. 7

Quote-O-Matic Stock Service for 5/16/95

GVIL	14.00
LMASX	20.84
ODWA	18.50
SPLS	27.12
TSA	19.25
lavtx	21.41

FIG. 8A

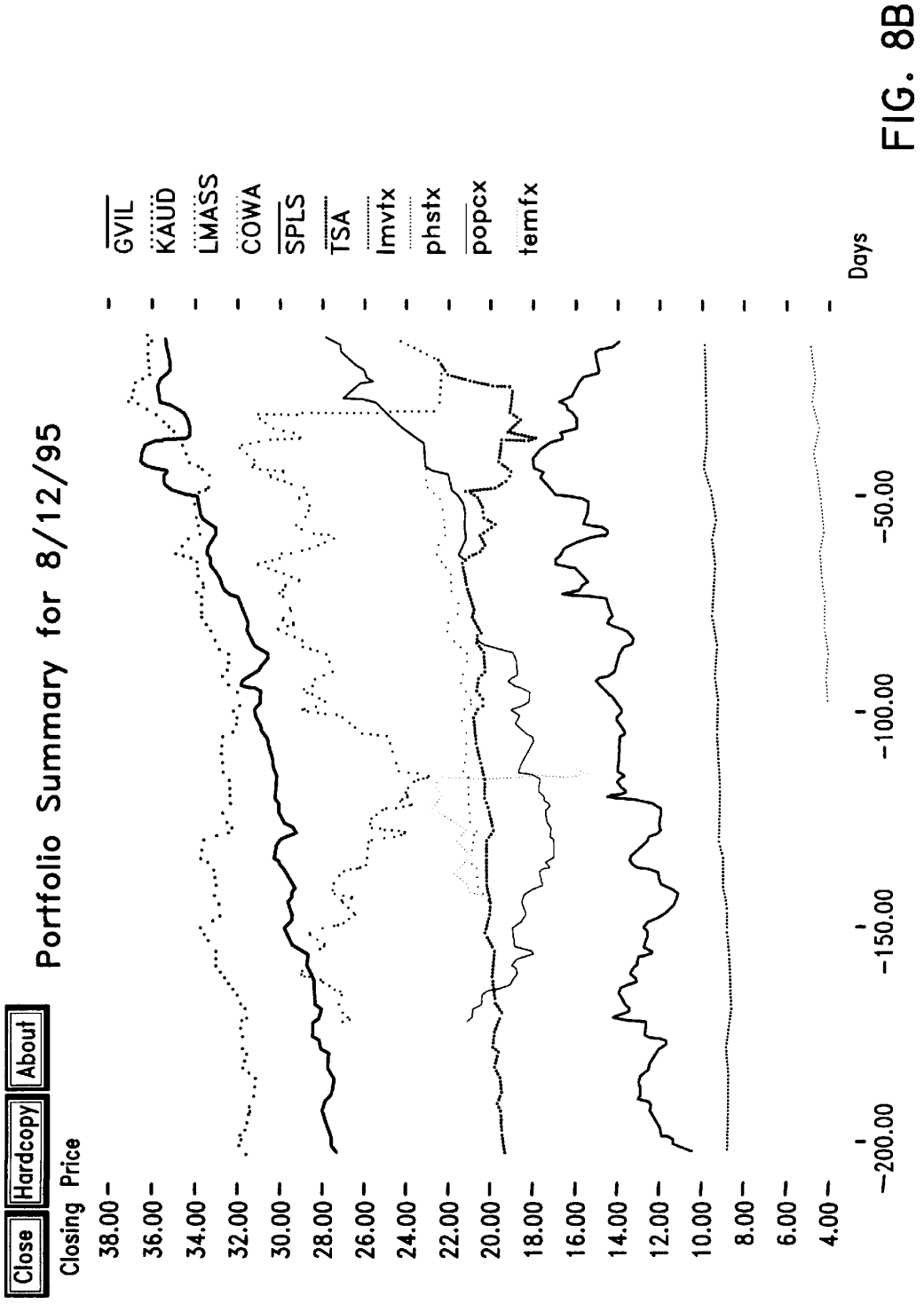

Close
Hardcopy
About
Portfolio Summary for 8/12/95
Closing Price
38.00
36.00
34.00
32.00
30.00
28.00
26.00
24.00
22.00
20.00
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
-200.00
-150.00
-100.00
-50.00
Days
GVIL
KAUD
LMASS
COWA
SPLS
TSA
lmvfx
phstx
popcx
temfx

FIG. 8B

DOCUMENT STREAM OPERATING SYSTEM

FIELD OF THE INVENTION

The present invention relates to an operating system in which documents are stored in a chronologically ordered "stream". In other words, that is, as each document is presented to the operating system, the document is placed according to a time indicator in the sequence of documents already stored relative to the time indicators of the stored documents.

Within this application several publications are referenced by arabic numerals within parentheses. Full citations for these and other references may be found at the end of the specification immediately preceding the claims. The disclosures of all of these publications in their entireties are hereby incorporated by reference into this application in order to more fully describe the state of the art to which this invention pertains.

BACKGROUND OF THE INVENTION

Conventional operating systems frequently confuse inexperienced users because conventional operating systems are not well suited to the needs of most users. For example, conventional operating systems utilize separate applications which require file and format translations. In addition, conventional operating systems require the user to invent pointless names for files and to construct organizational hierarchies that quickly become obsolete. Named files are an invention of the 1950's and the hierarchical directories are an invention of 1960's.

Some conventional operating systems employ a "desktop metaphor" which attempts to simplify common file operations by presenting the operations in the familiar language of the paper-based world, that is, paper documents as files, folders as directories, a trashcan for deletion, etc. Also, the paper-based model is a rather poor basis for organizing information where the state of the art is still a messy desktop and where one's choices in creating new information paradigms is constrained [1].

Thus, conventional operating systems suffer from at least the following disadvantages: (1) a file must be "named" when created and often a location in which to store the file must be indicated resulting in unneeded overhead; (2) users are required to store new information in fixed categories, that is directories or subdirectories, which are often an inadequate organizing device; (3) archiving is not automatic; (4) little support for "reminding" functions are provided; (5) accessibility and compatibility across data platforms is not provided and (6) the historical context of a document is lost because no tracking of where, why and how a document evolves is performed. "Naming" a file when created and choosing a location in which to place the file is unneeded overhead: when a person grabs a piece of paper and starts writing, no one demands that a name be bestowed on the sheet or that a storage location be found. Online, many filenames are not only pointless but useless for retrieval purposes. Storage locations are effective only as long as the user remembers them.

Data archiving is an area where conventional electronic systems perform poorly compared to paper-based systems. Paper-based systems are first and foremost archiving systems, yet data archiving is difficult in conventional desktop systems. Often, users throw out old data rather than undertaking the task of archiving and remembering how to get the data back. If archiving and retrieval of documents is convenient, old information could be reused more often.

Reminding is a critical function of computer-based systems [2] [3], yet current systems supply little or no support for this function. Users are forced either to use location on their graphical desktops as reminding cues or to use add-on applications such as calendar managers.

A solution to these disadvantages is to use a document stream operating system. One such system is outlined in a 1994 article [4]. However, this article fails to address many of the disadvantages of conventional operating systems.

SUMMARY OF THE INVENTION

One object of the present invention is to provide a document stream operating system and method which solves many, if not all, of the disadvantages of conventional operating systems.

Another object of the present invention is to provide a document stream operating system in which documents are stored in one or more chronologically ordered streams.

An additional object of the present invention is to provide an operating system in which the location and nature of file storage is transparent to the user, for example, the storage of the files is handled automatically and file names are only used if a user chooses to invent such names.

A further object of the present invention is to provide an operating system which takes advantage of the nature of electronic documents. For example, a conventional paper document can only be accessed in one place, but electronic documents can be accessed from multiple locations.

Another object of the present invention is to organize information as needed instead of at the time the document is created. For example, streams may be created on demand and documents may belong to as many streams as seems reasonable or to none.

An additional object of the present invention is to provide an operating system in which archiving is automatic.

A further object of the present invention is to provide an operating system with sophisticated logic for summarizing or compressing a large group of related documents when the user wants a concise overview. In addition, this summarizing can include pictures, sounds and/or animations. Also, no matter how many documents fall into a given category, the operating system is capable of presenting an overview in a form so that all the documents are accessible from a single screen.

Also, an object of the present operating system is to make "reminding" convenient.

Another object of the present invention is to provide an operating system in which personal data is widely accessible anywhere and compatibility across platforms is automatic. Accordingly, this invention provides that computers using the operating system of the present invention need not be independent data storage devices, but also act as "viewpoints" to data stored and maintained on external systems such as the INTERNET. Thus, in accordance with the present invention users can access their personal document streams from any available platform such as a UNIX machine, a Macintosh or IBM-compatible personal computer, a personal digital assistant (PDA), or a set-top box via cable.

According to one embodiment of the invention a computer program for organizing one or more data units is provided. The computer program includes: (1) means for receiving one or more of the data units, each of which is associated with one or more chronological indicators; and (2) means for linking each of the data units according to the

chronological indicators to generate one or more streams of data units. Other embodiments of the invention also provide: (1) chronological indicators including past, present, and future times; and (2) means for displaying the streams, wherein respective indicia representing the data units are displayed and each data unit includes textual data, video data, audio data and/or multimedia data. The means for displaying the streams may further include displaying selected segments of the streams corresponding to selected intervals of time. The means for receiving may further include means for receiving data units from the Word Wide Web or from a client computer.

According to another embodiment of the invention, a method of organizing one or more data units is provided including the steps of: (1) receiving one or more data units, each of which is associated with one or more chronological indicators; and (2) linking each of the data units according to the chronological indicators to generate one or more streams of data units. In other embodiments, the chronological indicators may include past, present, and future times. The method may further include the steps of: (1) displaying the streams, wherein respective indicia represent each data unit and each of the data units may be textual data, video data, audio data and/or multimedia data. The step of displaying the streams may further include the steps of: (1) receiving from a user one or more values indicative of one or more selected segments of the streams corresponding to selected intervals of time; and (2) displaying the segments of the streams corresponding to the selected intervals of time.

These and other advantages of the present invention will become apparent from the detailed description accompanying the claims and the attached figures.

DESCRIPTION OF THE DRAWINGS

FIG. **1** shows a viewport in one embodiment of the present invention;

FIG. **2** shows a substream menu in one embodiment of the present invention;

FIG. **3** shows a list of summary types for the substream chosen in

FIG. **2** of the present invention;

FIG. **4** shows the time display in one embodiment of the present invention;

FIG. **5** shows a calendar-based dialog box in one embodiment of the present invention;

FIG. **6***a* shows a dialog box in connection with a phone call in one embodiment of the present invention;

FIG. **6***b* shows a summary of phone calls in one embodiment of the present invention;

FIG. **7** shows a phone call record dialog box in one embodiment of the present invention;

FIG. **8***a* shows text data used by one embodiment of the present invention; and

FIG. **8***b* shows the result of a summarize operation in one embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

This invention is a new model and system for managing personal electronic information which uses a time-ordered stream as a storage model and stream filters to organize, locate, summarize and monitor incoming information. Together, streams and filters provide a unified framework that subsumes many separate desktop applications to accomplish and handle personal communication, scheduling, and search and retrieval tasks. One embodiment of the present invention utilizes a machine-independent, client/server open architecture so that users can continue to use the same conventional document types, viewers and editors.

A "stream" according to the present invention is a time-ordered sequence of documents that functions as a diary of a person or an entity's electronic life. Every document created and every document send to a person or entity is stored in a main stream. The tail of a stream contains documents from the past, for example starting with an electronic birth certificate or articles of incorporation. Moving away from the tail and toward the present and future, that is, toward head of the stream more recent documents are found including papers in progress or new electronic mail. A document can contain any type of data including but not limited to pictures, correspondence, bills, movies, voice mail and software programs. Moving beyond the present and into the future, the stream contains documents allotted to future times and events, such as, reminders, calendar items and to-do lists. Time-based ordering is a natural guide to experience. Time is the attribute that comes closest to a universal skeleton-key for stored experience. Accordingly, streams add historical context to a document collection with all documents eventually becoming read-only, analogously as history becomes "set in stone". The stream preserves the order and method of document creation. Also, like a diary, a stream records evolving work, correspondence and transactions because historical context can be crucial in an organizational setting.

One embodiment of this invention allows for basic operations to be perform on a stream: new, clone, transfer, find and summarize.

Users create documents by means of the new and clone operations. New creates a new, empty document and adds the document to the main stream. Clone duplicates an existing document and adds the duplicate to the main stream at a new time point. Documents can also be created indirectly through the transfer operation. The transfer operation copies a document from one stream to another stream. Creation of a document if "transparent" because documents, by default, are added to the at the present time point. Internally, the document is identified by a time indication so no name is required from the user for the document. Nevertheless, a user can optionally name a document is desired.

Some streams can be organized on the fly with the find operation. Find prompts for a search query, such as "all E-mail I haven't responded to," or "all faxes I've sent to Schwartz" and creates a substream. Substreams, unlike conventional, virtual or fixed directories which only list filenames, present the user with a stream "view" of a document collection. This view, according to the present invention contains all documents that are relevant to the search query. Also, unlike searches of conventional fixed directories, the substream is generated by default from all the documents in the main stream. Accordingly, individual substreams may overlap, that is, contain some documents that are the same and can be created and destroyed on the fly without affecting the main stream or other substreams.

The find operation creates a substream of the main stream or of another substream based on, for example, a boolean attribute-and-keyword expression or a 'chronological expression', for example, "my last letter to Schwartz". Also, substreams may point to the future, for example, "my next appointment".

Once created, substreams operate dynamically, that is, if a user allows a substream to persist, the substream will collect new documents that match the search criteria as documents arrive from outside the operating system or as the user creates the document. This dynamic operation provides automatic monitoring of information because the substream not only organizes the documents as received but also filters for incoming information. For example, a substream created with the query "find all documents created by other people" would subsume a user's mailbox and automatically collect all arriving mail from other users. A substream remains in existence until destroyed by the user and acts as a filter by examining each new document that enters the main stream.

Although a document may belong to any number of substreams, the document also enters and remains on the main stream. A substream, in other words, is a "subset" of the main stream document collection. In other words, a way of looking at the main stream so as to exclude certain documents temporarily.

The summarize operation "compresses" or "squish" a stream to generate one or more overview documents. The content of an overview document depends on the type of documents in the stream. For instance, if the stream contains the daily closing prices of all the stocks and mutual funds in a user's investment portfolio, the overview documents may contain a chart displaying the historical performance of particular securities and the user's net worth. As another example, if the substream contains a list of tasks a user needs to complete, the overview document might display a prioritized "to-do" list. Thus, the summarize operation collapses a stream into a summary document. This summary document is a "live" document which is updated as additional documents are added to the main stream.

The type of summary depends on the type of documents in the substream. In one embodiment of the present invention at least one general "squish" function is provided no matter which stream is to be squished. Typically, however, the user will have a number of different squishers to choose from, for example, one squisher might produce a summary in words, while another squisher might produce a graph. Also "custom squishers" may be supplied by third parties or created by the user.

Another aspect of the invention is that applications execute "inside" a stream document leaving any output in that document. Thus, running an application is a variant of the new operation. For example, to run a spreadsheet application such as Lotus 1-2-3, a user creates a new document at the head of the main stream, specifically, a "live" spreadsheet document. The application itself is stored on the main stream, or located by means of a calling card that points to another stream containing the application.

A stream has three main portions: past, present, and future. The "present" portion of the stream holds "working documents", which also includes the timepoint in the stream where new documents are created and where incoming documents are placed. As documents age and newer documents are added, older documents pass from the user's view and enter the "past portion" where the documents are eventually "archived". By disappearing from view old information is automatically cleared away so the old information will not clutter up the workspace. At some future point if documents in the past portion are needed, such documents can be located with the find operation even if the past document has already been archived.

The "future" portion of the stream allows documents to be created in the future. Future creation is a natural method of posting reminders, for example, meeting dates and scheduling information. The system allows users to dial to the future by selecting a future timepoint for a document. The present invention keeps the document until that future time occurs. When the time of documents timepoint arrives the reminder document is brought into view and the document enters the present portion of the stream.

One embodiment of the present invention is implemented in a client/server architecture running over the Internet. The server is the workhorse of this embodiment handling one or more streams by storing all main stream and substream documents. Each view of a stream is implemented as a client of the server and provides the user with a "viewpoint" interface to document collections, that is, streams. The "look and feel" of the viewport may be different for different computing platforms but each viewport should support the basic operations.

One embodiment of the present invention implements a client viewport using graphically based X Windows, another embodiment implement a client viewport solely with text in standard ASCII (American Standard Characters for Information Interchange) and yet another embodiment implements a client viewport for the NEWTON personal digital assistant (PDA). The X Windows viewport provides the full range of functionalities including picture and movie display. In contrast the text-only viewport has a mail-like interface although all the basic operations are available. Also, because the NEWTON PDA lacks substantial internal memory and relies on slow external communications, that is low bandwidth, a minimal stream-access method is provided.

The X Windows viewport embodiment is shown in FIG. **1**. The interface is based on a visual representation of the stream metaphor **5**. Users can slide the mouse pointer **10** over the document representations to "glance" at each document, or use the scroll bar **20** in the lower left-hand corner to move through time, either into the past or into the future portion of the stream.

Color and animation indicate important document features. A red border in one embodiment means "unseen" and a blue border means "writable". Open documents may be offset to the side to indicate when the document is being edited. In this embodiment incoming documents slide in from the left side and newly created documents pop down from the top and push the steam backwards by one document into the past.

External applications are used to view and edit documents which the user can select by clicking on the documents graphical representation. The external applications speed the learning process significantly because new users can continue to use familiar applications for example, conventional UNIX application such as emacs, xv, and ghostview, to create and view documents while using streams to organize and communicate the documents.

The X Windows interface prominently displays the basic operations, that is, New **30**, Clone **40**, Xfer **50** (that is, transfer), Find **60**, and Summarize **70** as buttons and/or menus. As discussed previously the New button creates a new document and adds the document to the stream at the "present" timepoint. The Clone button duplicates an existing document and places the copy in the stream. The Xfer button first prompts the user for one or more mail addresses and then forwards the selected document. The find operation is supported through a text entry box **60** that allows the user to enter a boolean search query which results in a new substream being created and displayed. The summarize menu **70** generates a new document which displays information from documents in a stream in a desired format, for example, a graph.

The X Windows interface of this document also provides additional buttons. The Print button **80** copies a selected document to a printer where documents may be either printed conventionally or moved to a printer stream. A software agent which can be associated with the stream forwards each new document to an appropriate printer. The Freeze button **90** makes a document read-only.

Pulldown menus are used to select documents from streams or existing substreams, create summaries, initiate personal agents and change the clock.

The Streams menu **110** allows the user to select from a list of locally available streams.

FIG. **2** shows the Substreams menu **120** of one embodiment of the present invention. This menu is divided into three sections. The first section **130** contains a list of operations that can be performed on substreams, for example, remove. The second section **140** contains one menu entry labeled "Your Lifestream", and causes the viewport to display a user's main stream. The third section **150** lists all of the user's substreams. As indicated by this third section, substreams can be created in an incremental fashion, that is, one substream generated from another resulting in a nested set of menus. In this example the nested menus were created by first creating a substream "lifestreams and david" **160** from the main stream and then creating two substreams from this substream, "scenarios" and "ben" **170**. Substream "scott" **180** was created from the "scenarios" substream. Semantically this incremental substreaming amounts to a boolean 'and' of each new query with the previous substream's query.

FIG. **3** shows the summarize menu **190** which lists the possible summary types. Choosing any of these menu options creates a substream summary and a new document containing the summary is placed on the stream.

The Personal Agents menu **200** lists a number of available software agent types. Personal software agents can be added to the user interface in order to automate common tasks.

The embodiment illustrated in FIG. **4** always displays the time in the upper right hand corner of the viewport interface. This time display also acts as a "time" pull-down menu **190** that allows the user to set the viewport time to the future or past via a calendar-based dialog box as illustrated in FIG. **5**. Setting the viewport time causes the cursor to point to that timepoint position in the stream such that all documents forward of that timepoint, that is, towards the head of the stream have a future timestamp and all documents behind that timepoint, that is, towards the tail, have a past timestamp. As time progresses, this cursor moves forward towards the head of the stream. When the cursor slips in front of the present timepoint "future" documents are added to the visible part of the stream in the viewpoint, just like new mail arrives.

The effect of setting the time to the future or past is to reset the time-cursor temporarily to a fixed position designated by the user. Normally the user interface displays all documents from the past up to the time-cursor. Setting the time-cursor to the future allows the user to see documents in the future part of the stream. Creating a document in the future results in a document with a future timestamp. Once the user is finished time-tripping, the user can reset to the present time by selecting the "Set time to present" menu option in the time menu.

In one embodiment of the present invention "browse cards" **100** are employed so that when the user touches a document in the stream-display with the cursor, a browse card appears. The purpose of the browse card is to help the user identify a document by providing the user some idea of the document's contents in a small window. The content of browse cards is an abbreviated version of a document which as been compressed into an micro-document like an index card. In one embodiment, the browse card creation operation does header stripping so that the browse card displays the first non-trivial words in a document. In another embodiment, complex analysis is performed on the document contents so that 'most important' words, pictures and/or sounds are presented.

Another embodiment of the present invention provides "calling cards" which represent or point to a stream or substream Every stream has a calling card and the only way to reference a stream is via this calling card. In this embodiment the find operation performs as follows: (1) the user provides a search query; (2) an appropriate substream is generated; (3) the substream's calling card is generated; and (4) the new calling card is deposited as a new document at the head of the main stream. Every duplicated calling card bears on the face text, an icon or both. In the case of the find operation, the new calling card is marked with the argument supplied by the user for the search query, for example "from: Schwartz and Lifestreams" or "last letter from Piffel". As a default in this embodiment, the interface will automatically display the new substream.

Another embodiment of the present invention allows documents to be grouped explicitly into a substream. With this feature the user marks, that is, selects all documents to be included in the substream and groups the selected documents into a substream by creating a new calling card. The new calling card comes equipped with a system-created icon which is marked on all documents that are part of the new stream and the user may add any other notation to the face of the new calling card, for example, "these should be merged together to produce the Zeppelin report."

In the embodiments with calling cards the "transfer" operation takes two arguments: a document and a calling card so that the document is copied onto the stream designated by the calling card. The document may itself be a calling card and depending on instructions from the user, either the calling card itself or the stream designated by the calling card is copied onto the new stream.

Each main stream in this embodiment has a calling card which allows 'inter-main stream' communication. To communicate a user includes on the face of the calling card for the user's main stream whatever information the user is willing to make public. Other users wanting to send that electronic mail will need a copy of that user's calling card, which might be, for example, "Rock Q. Public, Blimp Mechanic, Passaic N.J."

To give only limited access to user's stream, a user provides a copy of the calling card customized to provide the desired access. Minimal access gives other users append-only privileges, that is, user B can send user A mail, but cannot view anything on user A's stream. Access restrictions beyond "minimal" are stated in terms of substreams. In other words, a calling card gives access to all documents contained in the specified substream unless that document is also contained on one of specified excluded substreams.

The present invention allows a stream document to contain another stream, that is a 'stream envelope'. A stream envelope is equivalent to a 'value' calling card versus the 'reference' calling cards discussed above. In other words, rather than point to another stream with a calling card, the stream envelope contains a copy all the documents from the other stream. For example, user A transfers to user B a

substream consisting of all Zeppelin-manual correspondence which contains many documents. However, a single new document appears on user B's stream: a stream envelope. The stream envelope may be opened yielding the many documents of the forwarded stream.

According to the present invention, a text-editor designed specifically for stream A can treat a document as a stream of bytes so that software agents designed to 'ride' streams could ride documents as well. Also, the stream find operation can scan the streamed document and synchronization based on stream properties can be applied to the streamed document.

Streams can be copied and combined into new streams, that is, streams can be merged. For example, if a user acquires stream segments from ten electronic newspapers and magazines all covering the same one-month period, the segments can be merged in a sorted order into a single combined stream.

Another feature of the present invention is a card gallery which consists of some reasonable number of microdocuments, for example twelve, arranged in such a way that each is always fully visible on the viewport for example, in two columns of 6 each at the right of the display. Each micro-document is a calling card or a micro-browse-card (MBC) to a "regular" document on the stream or in a squish. The micro browse card in the card gallery represent documents a user has been working on. Whenever a user opens a document or creates a substream or squish, the corresponding micro browse card is added to the card gallery. A user can re-open the document, squish or have the viewport display the substream, by clicking on, or otherwise selecting, the corresponding micro browse card.

The micro browse card is administered as a least-recently-used cache, that is, new cards are dealt on top of the least-recently-used existing card, however, users can override this mechanism and place or lock a card in the gallery. For example, a live squish can act as weather-station, appointment calendar, stock ticker or other current-status reporter if the user locks the micro browse card for the squish in the gallery.

Because a users' card gallery includes by default the calling cards of streams the user has recently opened, the card gallery acts, to the extent streams are used as Web sites, as a World Wide Web "hot list."

In one embodiment of the present invention at least some part of the stream is in the form of a receding stack of upright rectangles, framed in such a way that only the top line of each document is visible. A foreshortened viewing angle yields a view that is approximately a right triangle, the bottom edge aligned with the bottom of the display and the left edge aligned with the display's left border.

In another embodiment of the inventive operating system a 'slide rule' bar display is provided which is labelled with the endpoints of the stream, that is, the dates of the farthest-past and farthest-future documents. The document density can be illustrated, for example, by the amount of color saturation of the bar at any point. This type of display aids the user because some days, weeks, months or other time period have more associated documents, some have fewer. The slide rule has a magnifier that the user can slide via a mouse, for example up and down the bar. The magnifier obscures the portion of the sliderule that lies beneath, but the obscured segment is replaced by an enlarged view of the small part of the stream starting at the point touched by the upper edge of the magnifier or, some similar protocol for defining the starting point of the magnified segment. By sliding the magnifier, you change the part of the stream currently displayed in the main perspective view.

In another embodiment of the present invention at least some part of the stream is in the form of a conventional calendar month display. With this display, the stream-segment associated with day n appears as a list of document headers in n's calendar box.

To contemplate the future instead of the past, according to one embodiment of the present invention, a user can reverse the stream so that the head of the stream now appears in front, with the nearest-future document immediately behind that document, next-nearest behind that and so forth. The user then looks from the present into the future so that furthest-away document in the display equals furthest-away in future time.

All documents older than some date d may be moved by the server from immediately-accessible storage to cheaper, long-term storage. When a document is archived in this way, however, the browse card of that document may remain available in immediately-accessible storage, so that the archived document appears in the regular way in the viewport. When a user opens an archived document, the user may incur some delay as the server locates and reloads the body of the document.

Automatic archiving is a feature of the standalone embodiment and user-managed web site embodiment. In either embodiment, the streams operating system monitors remaining disk space and when available space is low, the operating system asks the user to pop in some diskettes or other storage media. Similarly when an archived document needs to be reloaded, the operating system tells the user which diskettes or other storage media to insert. In another embodiment of the present invention a chat feature is provided. If two users want to chat online in UNIX TALK style; the user creates a new stream and each user focusses the viewport on that new stream. To make a comment, a user pops a new document on the stream head with the comment contained as text inside the document. The stream synchronization properties allow many users to manipulate a stream concurrently, and allow a user to block at the end waiting for the arrival of a new document which would mean in this case awaiting the next comment.

A chat stream by its nature provides: (1) permanent record and (2) support for multiple parties to a conversation. A chat stream is in this sense is a real-time bulletin board. In this regard a network bulletin board may be stored in a stream providing: (1) archived comments that can be searched and retrieved using the standard streams operations; (2) synchronization characteristics like a chat stream; and (3) a bulletin board that can be located via the find operation.

In another embodiment of the present invention any software agent with the necessary access can ride your stream. Therefore streams can be the basis of groupware systems implemented for example as a flock of agents. For example, when user A's wants to schedule a meeting, a software agent departs from user A's stream to visit the streams of each of the other intended participants. Each user's stream lists the current appointments in the stream's future portion, and each user also includes a document giving the user's general availability in pre-arranged terms so that the meeting-maker software agent can understand. When the software agent finds an appropriate meeting time, the software agent posts a document to each stream's future and creates a new stream for the meeting itself. The software agent forwards the calling card of the meeting stream to each participant. This new stream serves as a chat stream on

which the participants can discuss the meeting beforehand, accumulate any material developed during the meeting itself and persists when the meeting is over as a record and a vehicle for post-meeting discussions.

In the following embodiments a stream naturally provides a structure for storing technical an electronic versions of a newspaper or magazine.

In addition, a mail-order firm might store its catalog in a stream with each document describing one item. A top page can embed calling cards as hyperlinks so that the streams pointed-to are updated automatically by the persistent substream mechanism. Each user can also reformat the catalog to taste by creating a substream containing descriptions of whatever sort of object interests the user.

In another embodiment of the present invention a phone conversation is stored as a time-ordered sequence of spoken sounds or as electronic representations. When two users want to have a phone conversation, the users can use software such as a software agent, that creates a new stream and hands each user the calling card. Each user's 'phone agent' tosses digitized representations of speech frames onto the stream and grabs each new frame that appears, turning each speech frame into sound. In this scheme, phone and voicemail are integrated in the all-purpose stream context and can be manipulated using the standard stream operations.

Additionally, in another embodiment of the present invention a television source can be stored as a time-ordered sequence of sound-and-image frames. Such television information is an archive as well as a realtime source and can be searched and substreamed. A television set is merely a viewport. Also, scheduling information can be stored in the television stream's future and tuning into a television station only requires double-clicking on the appropriate calling card. Similar embodiments can provide for radio stations, music sources, etc.

A stream according to the present invention can be controlled by a voice-interface as well as a computer and thereby be accessed via a conventional phone. The voice interface would allow: (1) the stream to be searched and manipulated; (2) new objects to be installed; (3) objects to be transferred; and (4) other capability.

The following embodiment discusses how the present invention is used for electronic mail. To send a message, the user creates a new document, for example by clicking on the New button and composes the message using a favorite editor. After composition, the message document is sent with a push of the Xfer button. Similarly, existing documents are easily forwarded to other users, or can be cloned and replied to. While all mail messages, both incoming and outgoing, are intermixed with other documents in the stream, the user can create a mailbox by substreaming on documents created by other users. A user can also create substreams that contain a subset of the mailbox substream, such as "all mail from Bob", or "all mail I haven't responded to".

With the present invention, a reminder can be generated as future electronic mail, that is a user can send mail that will arrive in the future. If the user dials to the future before writing a message document, when the message document is transferred the message document will not appear on recipient's stream until either that time arrives or the recipient happens to dial the recipient's viewport to the set creation date. In the present, the document will be in the stream data structure but the viewport will not show the document. By appearing just-in-time and not requiring the user to switch to yet another application, these reminders are more effective than those included in a separate calendar or scheduling utility program.

One embodiment of the present invention supports an electronic business cards document type as well as a 'phone call record' document for noting the date and time of phone contacts. In addition, the task of creating a phone call record is automated through a personal agent. The personal software agent is automatically attached to the personal agent menu so that anytime a user wants to make a call the use chooses "Make Phonecall" from the personal agent menu. The agent is spawned and the dialog box in FIG. **6***a* appears. The user types in the name of the callee and the agent searches the current stream for a business card with that name. If the name is found, the software agent creates and fills in the appropriate entries of the phone call record as seen in FIG. **7**. This functionality is similar to the use of the personal assistant on the Newton personal digital assistant. The user can later use the streams summarize operation to summarize the phone calls made. This results in a report as shown in FIG. **6***b*.

In another embodiment of the present invention this functionality is extended to include the functions of a time manager. Time managers generally track the billable hours a professional spends on one or more projects. In streams this is easily accomplished by creating a timecard that marks the starting and ending time of each task. These timecards are just thrown onto the stream as used. Then, before each billing period, the stream is summarized by the timecards, resulting in a detailed billing statement for each contract.

Another embodiment of the present invention organizes a user's personal finance. Large number of users already track their checking accounts, savings, investments, and budgets with applications such as QUICKEN. The types of records and documents used in these applications such as electronic checks, deposits, securities transactions, reports are conveniently stored and generated by streams.

For example, a stock quote service may forward the daily closing prices of a given portfolio to a user's stream at the end of every business day. These documents are as shown in FIG. **8***a*. Such documents can list each stock and mutual fund along with its closing price, giving the user a method of calculating the value of the user's assets on a specific day. But if the user wants higher-level view of the portfolio over time the summarize operation can be used. For example, the user first selects a substream containing the stock quote documents and selects the "summarize by portfolio" menu item. This operation compresses the data into a single chart of historical data which summarize the portfolio documents in the substream. This result is illustrated in FIG. **8***b*.

Another embodiment of present invention provides a stream-based checking account. Each check written creates a record on the users stream. Some of these checks are electronic checks sent to companies with an online presence; other checks are transcribed from written checks. The user, in this embodiment, employs a personal software agent to help balance his checkbook. At year's end the user runs a tax summary which squishes the financial information in the users stream onto income tax forms which can be sent electronically to the Internal Revenue Service.

Streams can also be used for budgeting, tracking expenditures, etc. Streams contain everything a user deals with in the user's electronic life in a convenient and searchable location.

As discussed previously, every user can send out custom calling cards that grant access to a user's stream. Thus, the particular user's stream can function as a personal World

Wide Web site such that the web site is merely a subset of the user's main stream or a substream. For the convenience of external users, a user can generate a "guide to this stream" document that functions as a top page. In the context of the present invention, a hyperlink, or a bookmark is just a calling card. By double clicking, or some comparable mechanism, on a calling card the viewport displays the specified stream. Embedding a link from one document to another document means to embed calling cards.

The present invention's personal web site provides more features than a conventional worldwide user side because: (1) the web site and personal information site are unified and maintained simultaneously with the same toolset; (2) visitors to the site use the same interface as for the visitor's own stream, that is, the visitor can browse, create substreams and squish; (3) visitors can be given customized access levels so that friendly visitors get to see more; and (4) the personal web site can filter incoming documents.

Streams of the present invention are designed to work with conventional World Wide Web browsers, thus opening a document of type web bookmark causes the appropriate browser to fire-up as an application the way a text editor fires up when the user opens a text document. However, streams also provide an indigenous web-browsing model. Key features such as calling cards and find provide this functionality so that the viewport itself functions as the browser.

Streams may also be quite useful for managing information outside of the system. For example, keeping track of web bookmarks is difficult and bookmarks are inconvenient to pass to other users. Conventional systems accomplish those transactions by copying a Web address from a web browser to an electronic mail message which the recipient then copies from electronic mail back to recipient's browser and adds this web address as a bookmark. Streams solve both of these problems.

In one embodiment, an agent watches each user's bookmark file for each time a new bookmark is added and then adds the same bookmark to a stream as a new Web address document. The effect of opening a Web address document in a stream is that the web browser comes to the foreground and attempts to connect to the Web address. In this way streams create a bookmark substream while at the same time making the data in the bookmarks readily available to any other search a user may make.

Passing Web addresses around is trivial, the user merely copies the Web address document to another user's stream (a one-step process) and the Web address is automatically included in the recipient bookmark substream.

A stream is a data structure that can be examined and to the extent possible manipulated by many processes simultaneously. Also a process may block the end of a stream, that is, suspend the stream operation, until awakened when a new document appears on the stream head. Streams need to support the block-at-the-end operation so that a software agent or what amounts to the same thing, that is, a substream or a live squish document can examine each new document arriving at the stream.

A stream must support simultaneous access because: (1) a user creates many software agents which may need to examine the stream concurrently; and (2) a user may have granted other users limited access to the user's stream, and the user will want access to this stream even while the other users access the stream.

One embodiment of the present invention is configured such that each server may support three to four simultaneous users with stream sizes on the order of 100,000 documents (perhaps a year or two of documents for the average user). In another embodiment, the operating system is configured such that lifestreams may have millions of documents or more. The substreaming aspect of one embodiment of the present invention is efficiently implemented using an inverse index of the document collection maintained by the server. No real performance problems with respect to retrieval have occurred. Given the very large indices that are being used on the Internet the retrieval scheme is expected to scale to large document collections.

Since a user is unlikely look at 10,000 documents at once and discern any usable information, the present invention does not provide the user with an entire document collection at once. Instead "cursors" are used to allow the user to view segments of the document collection and to load in more segments as needed.

One embodiment of the invention provides a single-threaded server which allows a single point of access. Other embodiments of the present invention utilize a multi-server and multi-threaded approach which provides a more scalable architecture.

Regarding the term "agent" used in this application, it is noted that this term refers one of three kinds of embedded computations: personal agents, document agents, and stream agents. Personal agents are typically attached to the user interface and can automate tasks or can learn from the user's interactions with streams. Document agents live on documents and are spawned by various events, for example, the first time that a document is accessed. Stream agents are attached to streams and execute whenever the stream changes in some way, for example, a new document appears on the stream.

Further, regarding the term "document", it is noted that this term includes traditional text based files, electronic mail files, binary files, audio data, video data, and multimedia data.

Additionally, this document stream operating system can be implemented as an independent operating system with all required subsystems such as: a storage subsystems in software and/or hardware for writing documents to disc drive, tape drives and the like; interrupt handling subsystems; and input/output subsystems. However, the present invention also encompasses implementations which utilize subsystems from other operating systems such as the Disk Operating System (DOS), WINDOWS, and OPERATING SYSTEM 7. In such implementations, the graphic user interface (GUI) of the other operating system can be replaced by the present invention viewports. Alternatively, the present invention can operate as a document stream utility for the other operating system.

It must be noted that although the present invention is described by reference to particular embodiments thereof, many changes and modifications of the invention may become apparent to those skilled in the art without departing from the spirit and scope of the invention, which is only limited by the appended claims. For documents may have associated attributes used to locate the document during a search, for example, a special code word selected by the user.

References

[1] Theodor Nelson. The right way to think about software design. In The Art of Human-Computer Interface Sesign (Ed.) Brenda Laurel, 1990.

[2] Thomas W. Malone. How do people organize their desks?Implications for the design of office information

systems. ACM Transactions on Office Systems, 1(1):99–112, January 1983.

[3] M. Lansdale. The psychology of personal information management. Applied Ergonomics, March 1988.

[4] David Gelernter. The cyber-road not taken. The Washington Post, April 1994.

What is claimed is:

1. A computer system which organizes each data unit received by or generated by the computer system, comprising:

means for generating a main stream of data units and at least one substream, the main stream for receiving each data unit received by or generated by the computer system, and each substream for containing data units only from the main stream;

means for receiving data units from other computer systems;

means for generating data units by the computer system;

means for selecting a timestamp to identify each data unit;

means for associating each data unit with at least one chronological indicator having the respective timestamp;

means for including each data unit according to the timestamp in the respective chronological indicator in the main stream; and

means for maintaining the main stream and the substreams as persistent streams.

2. The computer system of claim **1**, wherein each timestamp is selected from the group consisting of: past, present, and future times.

3. The computer system of claim **1**, wherein each data unit includes textual data, video data, audio data and/or multimedia data.

4. The computer system of claim **1**, wherein the means for receiving further comprises means for receiving data units from the World Wide Web.

5. The computer system of claim **1**, wherein said means for receiving further comprises means for receiving data units from a client computer.

6. The computer system according to claim **1**, further comprising:

means for displaying alternative versions of the content of the data units.

7. A computer system according to claim **1** further comprising:

means for summarizing the contents of data units in one of the streams to generate one or more overview data units and for including the overview data unit in one of the streams.

8. A computer system according to claim **7**, wherein the means for summarizing further comprises means for continuously updating the overview data units to include changes in the contents of data units in the stream being summarized.

9. A computer system according to claim **1** further comprising:

means for archiving a data unit associated with a timestamp older than a specified time point while retaining the respective chronological indicator and/or a data unit having a respective alternative version of the content of the archived data unit.

10. The computer system of claim **1**, wherein the computer program further comprises:

means for operating on any of the streams using a set of operations selected by a user.

11. The computer system of claim **1** further comprising:

means to generate substreams from existing substreams.

12. A computer system as in claim **1**, further comprising:

means for generating a data unit comprising an alternative version of the content of another data unit; and

means for associating the alternative version data unit with the chronological indicator of the another data unit.

13. A method which organizes each data unit received by or generated by a computer system, comprising the steps of:

generating a main stream of data units and at least one substream, the main stream for receiving each data unit received by or generated by the computer system, and each substream for containing data units only from the main stream;

receiving data units from other computer systems;

generating data units in the computer system;

selecting a timestamp to identify each data unit;

associating each data unit with at least one chronological indicator having the respective timestamp;

including each data unit according to the timestamp in the respective chronological indicator in at least the main stream; and

maintaining at least the main stream and the substreams as persistent streams.

14. The method of claim **13**, wherein each timestamp is selected from the group consisting of: past, present, and future times.

15. The method of claim **13**, further comprising the step of displaying the streams on a display device as visual streams.

16. The method of claim **15**, wherein the step of displaying the streams further comprises the steps of:

a) receiving from a user one or more indications of one or more selected segments of the streams corresponding to one or more selected intervals of time, and

b) displaying the selected segments.

17. The method of claim **13**, wherein each data unit includes textual data, video data, audio data and/or multimedia data.

18. The method of claim **13**, further comprising the step of:

providing access to a first stream from a second stream by generating a data unit indicating the first stream.

19. The method of claim **13**, further comprising the steps of:

selecting access privileges to provide to a first stream from a second stream; and

providing access to the first stream from the second stream according to the access privileges.

20. The method of claim **13**, further comprising the step of:

displaying data from one of the data units in abbreviated form.

21. The method of claim **13**, further comprising the step of:

summarizing the contents of data units in a stream to generate one or more overview data units and including the overview data unit in one of the streams.

22. The method of claim **13**, further comprising the step of:

archiving data units having timestamps older than a specified time point.

23. A computer system for organizing each data unit received by or generated by the computer system, comprising:

means for generating a main stream of data units and at least one substream, the main stream for receiving each data unit received by or generated by the computer system, and each substream for containing data units only from the main stream; means for associating each data unit with at least one chronological indicator having a respective timestamp which identifies the data unit; means for including each data unit according to the timestamp in a respective chronological indicator in the main stream; means for maintaining the main stream and substreams as persistent streams;

means for generating a data unit having indicia to allow access to a first stream from a second stream;

means for including the data unit having the indicia in the second stream; and

means for providing access to the first stream from the second stream in accordance with the indicia.

24. A computer system according to claim **23** further comprising:

means for providing limited access to the first stream from the second stream by generating a data unit indicating access privileges to the first stream.

25. A computer system for organizing each data unit received by or generated by the computer system, comprising:

means for generating a main stream of data units and at least one substream, the main stream for receiving each data unit received by or generated by the computer system, and each substream for containing data units only from the main stream; means for associating each data unit with at least one chronological indicator having a respective timestamp which identifies the data unit; means for including each data unit according to the timestamp in a respective chronological indicator in the main stream; means for maintaining the main stream and the substreams as a persistent streams;

means for representing one or more data units of a selected stream on a display device as document representations, each document representation including the timestamp of the respective data unit and the order of appearance of each data representation on the display device determined by the timestamp of the respective data unit;

means for selecting which data units are represented on the display device by selecting one of the document representations and displaying document representations corresponding to data units having timestamps within a range of a timepoint; and

means for selecting one or more of the document representations with a pointing device so that the data units represented by the selected document representations are further displayed with a second document representation comprising an alternative version of the content of the respective data unit.

26. A computer system as in claim **25**, wherein the document representations form a visual stream having a three-dimensional effect.

27. A computer system as in claim **26**, wherein the three-dimensional effect further comprises a perspective view.

28. A computer system as in claim **25**, wherein each document representation comprises a polygon and the polygons overlap to form a visual stream of polygons.

29. A computer system as in claim **25**, wherein the alternative version is an abbreviated version.

30. A computer system as in claim **25**, wherein the alternative version is a caption version.

31. A computer system as in claim **25**, wherein the alternative version is an expanded version.

32. A computer system as in claim **25**, further comprising:

means for selecting one or more alternative versions of the content of a respective data unit to display another alternative version of the content of the data unit.

33. A computer system as in claim **25**, further comprising:

means for updating the display device to provide a document representation for data units associated with chronological indicators having timestamps which become the present time.

* * * * *

US006006227C1

(12) **EX PARTE REEXAMINATION CERTIFICATE** (8298th)

United States Patent
Freeman et al.

(10) **Number: US 6,006,227 C1**
(45) **Certificate Issued: Jun. 7, 2011**

(54) **DOCUMENT STREAM OPERATING SYSTEM**

(75) Inventors: **Eric Freeman**, Branford, CT (US); **David H. Gelernter**, Woodbridge, CT (US)

(73) Assignee: **Mirror Worlds, LLC**, Tyler, TX (US)

Reexamination Request:
No. 90/010,506, Apr. 23, 2009

Reexamination Certificate for:
Patent No.: **6,006,227**
Issued: **Dec. 21, 1999**
Appl. No.: **08/673,255**
Filed: **Jun. 28, 1996**

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/695**; 707/752; 707/783; 707/914; 707/916; 707/917; 707/999.002; 707/999.007; 707/999.102; 707/E17.008; 707/E17.058

(58) **Field of Classification Search** 709/232
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,586,035	A	4/1986	Baker et al.
4,831,758	A	5/1989	Williams et al.
5,031,346	A	7/1991	Herring et al.
5,060,135	A	10/1991	Levine et al.
5,063,495	A	11/1991	MacPhail
5,140,676	A	8/1992	Langelaan
5,150,410	A	9/1992	Bertrand
5,159,669	A	10/1992	Trigg et al.
5,241,671	A	8/1993	Reed et al.
5,247,437	A	9/1993	Vale et al.
5,283,864	A	2/1994	Knowlton
5,287,448	A	2/1994	Nicol et al.
5,297,032	A	3/1994	Trojan et al.
5,299,122	A	3/1994	Wang et al.
5,303,361	A	4/1994	Colwell et al.
5,355,602	A	10/1994	Menke et al.
5,402,526	A	3/1995	Bauman et al.
5,430,710	A	7/1995	Mueller et al.
5,448,729	A	9/1995	Murdock
5,479,602	A	12/1995	Baecker et al.
5,499,330	A	3/1996	Lucas et al.
5,504,852	A	4/1996	Thompson-Rohrlich
5,528,739	A	6/1996	Lucas et al.
5,530,859	A	6/1996	Tobias, II et al.
5,535,063	A	7/1996	Lamming

(Continued)

FOREIGN PATENT DOCUMENTS

JP	6-180661	6/1994
WO	WO 2003/001345 A1	1/2003

OTHER PUBLICATIONS

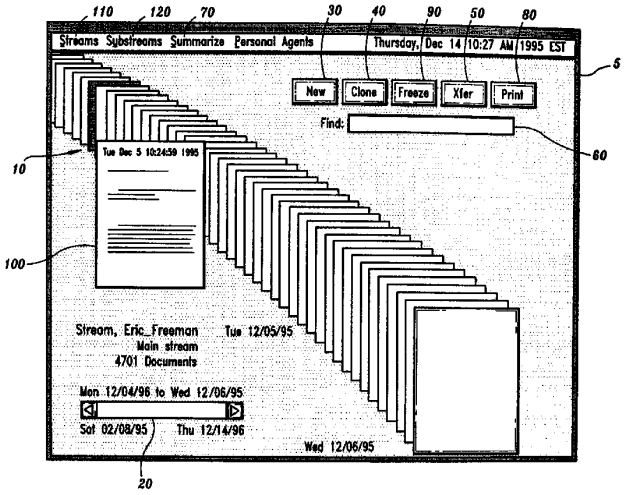
Carriero et al.. "The Lifestreams' Approach to Reorganize the Information World" Apr. 1995. (TR–1070v1").
Carriero et al.. "The Lifestreams' Approach to Reorganize the Information World" Apr. 1995. (TR–1070v2").
Freeman, Lifestreams Project Home Page, 3 pages, 1994–1996, Jan. 1, 1994.
W3C: A Little History of the World Wide Web, http://www.w3.org/History.html, (Accessed Jul. 15, 2010).

(Continued)

Primary Examiner—Christopher E Lee

(57) **ABSTRACT**

A document stream operating system and method is disclosed in which: (1) documents are stored in one or more chronologically ordered streams; (2) the location and nature of file storage is transparent to the user; (3) information is organized as needed instead of at the time the document is created; (4) sophisticated logic is provided for summarizing a large group of related documents at the time a user wants a concise overview; and (5) archiving is automatic. The documents can include text, pictures, animations, software programs or any other type of data.



U.S. PATENT DOCUMENTS

5,542,086	A		7/1996	Andrew et al.	
5,543,088	A		8/1996	Halbirt	
5,586,237	A		12/1996	Baecker et al.	
5,589,892	A		12/1996	Knee et al.	
5,600,833	A		2/1997	Senn et al.	
5,603,025	A		2/1997	Tabb et al.	
5,613,134	A		3/1997	Lucus et al.	
5,616,876	A		4/1997	Cluts	
5,621,874	A		4/1997	Lucas et al.	
5,621,906	A		4/1997	O'Neill et al.	
5,625,818	A		4/1997	Zarmer et al.	
5,649,182	A		7/1997	Reitz	
5,649,188	A		7/1997	Nomura et al.	
5,701,582	A		12/1997	DeBey	
5,724,567	A		3/1998	Rose et al.	
5,729,730	A		3/1998	Wlaschin et al.	
5,737,737	A		4/1998	Hikida et al.	
5,758,324	A		5/1998	Hartman et al.	
D395,297	S		6/1998	Cheng et al.	
5,764,972	A		6/1998	Crouse et al.	
5,771,355	A	*	6/1998	Kuzma	709/232
5,778,364	A		7/1998	Nelson	
5,784,620	A		7/1998	Isham	
D398,299	S		9/1998	Ballay et al.	
5,835,129	A		11/1998	Kumar	
5,835,789	A		11/1998	Ueda et al.	
5,890,177	A		3/1999	Moody et al.	
5,905,992	A		5/1999	Lucas et al.	
5,912,668	A		6/1999	Sciammarella et al.	
5,937,417	A		8/1999	Nielsen	
6,006,227	A		12/1999	Freeman et al.	
6,012,072	A		1/2000	Lucas et al.	
6,012,074	A		1/2000	Lucas et al.	
6,151,610	A		11/2000	Senn et al.	
6,178,409	B1		1/2001	Weber et al.	
6,199,082	B1		3/2001	Ferrel et al.	
6,202,058	B1		3/2001	Rose et al.	
6,243,724	B1	*	6/2001	Mander et al.	
6,262,732	B1		7/2001	Coleman et al.	
6,396,513	B1		5/2002	Helfman et al.	
6,401,097	B1		6/2002	McCotter et al.	
6,457,017	B2	*	9/2002	Watkins et al.	
6,466,237	B1	*	10/2002	Miyao et al.	
6,496,857	B1	*	12/2002	Dustin et al.	
6,523,048	B2	*	2/2003	DeStefano	
6,638,313	B1	*	10/2003	Freeman et al.	
6,639,876	B2		10/2003	Kalis et al.	
6,714,489	B2		3/2004	Kalis et al.	
6,725,427	B2	*	4/2004	Freeman et al.	
6,768,999	B2	*	7/2004	Prager et al.	

OTHER PUBLICATIONS

Gelernter, David, "The Cyber–Road Not Taken", The Washington Post, Apr. 3, 1994.

Bolt, Richard A., "Spatial Data–Management", Massachusetts Institute of Technology, 1979.

Lansadale, M. W., et al., "Memoirs: A personal Multimedia Information System", The Proceedings of the Fifth Conference of the British Computer Society Human Computer Interaction Specialist Group University of Nottingham, Sep. 1989.

Pemberton, Steven, "Not With A Whimper", SIGCHI, Apr. 1998.

Miller, James R., "The Apple Advanced Technology Group—An Introduction to the Special SIGCHI Bulletin Issue", SIGCHI, Apr. 1998.

Russell, Daniel M., "The ATG Knowledge Management Technologies Laboratory", SIGCHI, Apr. 1998.

Miller, James R., "An Overview of the ATG Intelligent Systems Program", SIGCHI, Apr. 1998.

Miller, James R., et al., "From Documents to Objects: An Overview of LiveDoc", SIGCHI, Apr. 1998.

Bonura, Thomas, et al., "Drop Zones: An Extension to LiveDoc", SIGCHI, Apr. 1998.

Boguraev, Branimir, et al., "An Architecture for Content Analysis of Documents and Its Use in Information and Knowledge Tasks", SIGCHI, Apr. 1998.

Boguraev, Branimir, et al., "Dynamic Document Presentation", SIGCHI, Apr. 1998.

Nardi, Bonnie, et al., "An Online Digital Photography Course for High School Teachers", SIGCHI, Apr. 1998.

Shulka, Shilpa V., "Hit Squads & Bug Meisters", SIGCHI, Apr. 1998.

Rose, Daniel E., "Beyond Search: The Information Access Research Group at Apple", SIGCHI, Apr. 1998.

Russell, Daniel M., "User Experience Research Group", SIGCHI, Apr. 1998.

Walker, William F., Rapid Prototyping of Awareness Services Using a Shared Information Server, SIGCHI, Apr. 1998.

Bellegarda, Jerome R., "Interaction–Driven Speech Input", SIGCHI, Apr. 1998.

Machiraju, N. Rao, "The ATG Learning Communities Laboratory: An Overview", SIGCHI, Apr. 1998.

Bellamy, Rachel, et al., "Learning Conservations", SIGCHI, Apr. 1998.

Houde, Stephanie, et al., "In Search of Design Principles for Tools and Practices to Support Communication Within a Learning Community", SIGCHI, Apr. 1998.

Roddy, Brian J., et al., "Interface Issues in Text Based Chat Rooms", SIGCHI, Apr. 1998.

Spohrer, Jim, "ATG Education Research: The Authoring Tools Thread", SIGCHI, Apr. 1998.

Graves, Mike, et al., "Unfamiliar Ground: Designing Technology to Support Rural Healthcare in India",SIGCHI, Apr. 1998.

Mountford, S. Joy, "A History of the Apple Human Interface Group", SIGCHI, Apr. 1998.

Harris, Jed, et al., "Discourse Architecture", SIGCHI, Apr. 1998.

Singer, Samuel Weller, "Researches into the History of Playing Cards", Printed by T. Bensley and Son, 1816.

"Playing Poker at Egan's" (image), http://www.orww.org/Harney_Cattle/Student_Reports/Images/11–Poker_1882.jpg(ca. 1882).

"Commercial America", Philadelphia Commercial Museum, Jul. 1911.

"Eleventh Annual Report of the Trustees of the Boston State Hospital", Published by Wright & Potter Printing Co., 1920.

"System—The Magazine of Business", A. W. Shaw Company, 1921.

Ibberson, J.R., MD, "The Business Side of General Practice", Canad. M.A.J., vol. 72, Apr. 15, 1955.

Carlbom, Ingrid, et al., "Planar Geometric Projections and Viewing Transformations", Computing Surveys, vol. 10, No. 4, pp. 465–502, Dec. 1978.

Schmandt, Christopher Martin, "Pages Without Paper", Bachelor of Science Thesis at the Massachusetts Institute of Technology, Dec. 1978.

Donelson, William C., "Spatial Management of Information", Architecture Machine Group, Massachusetts Institute of Technology, Proceedings of the 5th Annual Conference on Cumputer Graphics and Interactive Techniques, 1978.
Bolt, Richard A., "Put–That–There": Voice and Gesture at the Graphics Interface, Architecture Machine Group, Massachusetts Institute of Technology, ACM, 1980.
Halasz, Frank, et al., "Analogy Considered Harmful", Association for Computing Machinery, 1981.
Bolt, Richard A., "Gaze–Orchestrated Dynamic Windows", Architecture Machine Group, Massachusetts Institute of Technology,, Cambridge Massachusetts, ACM, Computer Graphics, vol. 15, No. 3, Aug. 1981.
Williams, Gregg, "The Lisa Computer System", BYTE Publications, Inc., Feb. 1983.
Williams, Gregg, "The Apple Macintosh Computer ", BYTE Publications, Inc., Feb. 1984.
Anderson, John J., "Atari 520ST: A Reborn Atari Once Again Points the Way to the Next Generation", Creative Computing, vol. 11, No. 10, Oct. 1985.
Greenberg, Saul, et al., "Issues and Experiences in the Design of a Window Management System", Canadian Information Processing Society—Edmonton Conference, 1986.
Henderson, D. Austin, Jr., et al., "Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window–Based Graphical User Interface", ACM Trans. On Graphics, vol. 5, No. 3, pp. 211–243, Jul. 1986.
Robek, Mary F., et al., "Information and Records Management", 3rd ed., Glencoe Publishing Co., pp. 160–163, 179–181, 192–193, 224–229, 248–251, (1987).
Lynch, Vincent, "American Jukebox—The Classic Years", Chronicle Books, pp. 36, 74–75, (1990).
Robertson, G. G., et al., "Cone Trees: Animated 3D Visualizations of Hierarchical Information", UIR–R–1991–06, (1991).
Card, S. K., et al., "The Information Visualizer: An Information Workspace", UIR–R–1991–01, (1991).
MacKinlay, Jock D., et al., "The Perspective Wall: Detail and Context Smoothly Integrated", UIR–R–1991–03, (1991).
Kay, Alan C., "The Early History of Smalltalk", ACM SIGPLAN Notices, vol. 28, No. 3, Mar. 1993.
Adams, Michael, et al., "Jukeboxes", Schiffer Publishing, Ltd., pp. 85–86, 103–104, 105, (1996).
Card, Stuart K., et al., "The WebBook and the Web Forager: An Information Workplace for the World–Wide Web", CHI '96 Conference on Human Factors in Computing Systems, Apr. 13–18, 1996.
Meyers, Brad A., "A Brief History of Human–Computer Interaction Technology", Interactions, pp. 44–54, 1998.
Bardini, Thierry, "Bootstrapping", Stanford University Press, Chapter 6, pp. 143–181, (2000).
Sutherland, Ivan Edward, "Sketchpad: A Man–Machine Graphical Communication System", PhD Dissertation at the Massachusetts Institute of Technology, Jan. 1963.
Engelbart, Doug, "The Click Heard Round the World", Wired Magazine, 2004.
Reimer, Jeremy, "A History of the GUI", arstechnica.com, May 5, 2005.
"The Library of Congress Classification System", Aug. 2006.
Barnes, Susan B, "Alan Kay: Transforming the Computer into a Communication Medium", IEEE Annals of the History of Computing, 2007.
Calamar, Gary et al., "Record Store Days: From Vinyl to Digital and Back Again", Sterling Publishing Co., 2009.
Masinter, Larry, "Document Management, Digital Libraries and the Web", Jun. 9, 1995.
Herot, Christopher F., "Spatial Management of Data", Computer Corporation of America, ACM Trans. On Database Systems, vol. 5, No. 4, pp. 493–514, Dec. 1980.
Newman, William M., "A System for Interactive Graphical Programming", Harvard University, ACM Spring Joint Computer Conference, 1968.
Schultz, Jan R., et al., "An Initial Operational Problem Oriented Medical Record System—for Storage, Manipulation and Retrieval of Medical Data", University of Vermont, ACM Spring Joint Computer Conference, 1971.
Donelson, William C., "Spatial Management of Data", Massachusetts Institute of Technology, 1977.
"Lotus Magellan Explorer's Guide", Lotus Development Corp., 1989.
Factor, Michael, et al., "Real–Time Data Fusion in the Intensive Care Unit", Yale University, Nov. 1991.
Gelernter, David, et al., "Graphics Interfaces & Sensor Activator Extensions for the Process Trellis Software", Yale University, Oct. 31, 1995.
Harrison, Beverly L., et al., "Timelines: An Interactive System for the Collection and Visualization of Temporal Data", Proceedings of Graphic Interface '94, pp. 141–148, (1994).
Gerlenter, David, "Applications and Systems for Large Scale Adaptive Parallelism", Mar. 22, 1997.
Date, C. J., "An Introduction to Database Systems, 3rd ed., The Systems Programming Series, vol. 1", IBM Corporation, Addison–Wesley Publishing Company, 1981.
"The Billboard" (magazine), Jan. 26, 1946.
"Billboard" (magazine), Jul. 4, 1992.
Swineheart, Daniel C., et al., "A Structural View of the Cedar Programming Environment", ACM Trans. On Programming Languages and Systems, vol. 8, No. 4, Oct. 1986.
Lucas, Peter, et al., "Workscape: A Scriptable Document Management Environment", Maya Design Group, Conference Companion CHI '94, Apr. 1994.
Bailay, Joseph M., "Designing Workscape: An Interdisciplinary Experience", Maya Design Group, Conference Companion CHI '94, Apr. 1994.
Perkins, Roderick, et al., "Inventing the Lisa User Interface", Interactions, pp. 40–53, (1997).
Feiner, Steven, et al., "An Experimental System for Creating and Presenting Interactive Graphical Documents", ACM Trans. On Graphics, vol. 1, No. 1, pp. 59–77, Jan. 1982.
Houde, Stephanie, "Iterative Design of an Interface for Easy 3–D Direct Manipulation", CHI '92, May 1992.
"Announcing Retrospect and Retrospect Remote 3.0", Business Wire, Aug. 8, 1995.
"Dantz Delivers Retrospect Archives", MacWEEK, Inc., Jun. 10, 1989.
"Archiving Easy on Retrospect: Uniform Interface to Storage Devices", MacWEEK, Inc., Feb. 21, 1989.
"Mac the Knife, GmbH Rumors", MacWEEK, Inc., Jan. 31, 1989.
"Dantz, ChrisMac drive over to CD–R", MacWEEK, Inc., Dec. 11, 1995.
"System 7.5.2 Fixes", MacWORLD, Oct. 1, 1995.
"Danz to Waltz in with Friendlier Retrospect", MacWEEK, Inc., Jul. 31, 1995.
"Dantz Eases Network Backup", InfoWorld, Jan. 11, 1993.

"Retrospect User's Guide", 1st Ed., Dantz Development Corporation, 1989.
"Retrospect User's Guide", 2nd Ed., Dantz Development Corporation, 1993.
Spence, Robert, et al., "Data Base Navigation: An Office Environment for the Professional", Behavior and Information Technology, vol. 1, No. 1, pp. 43–54, (1982).
Johnson, Jeff, et al., "The Xerox Star: A Retrospective", IEEE Computer, Sep. 1989.
Shneiderman, Ben, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations", University of Maryland, Proc. Visual Languages, Sep. 1996.
Plaisant, Catherine et al. "Lifelines: Visualizing Personal Histories", ACM CHI '96 Conference Proc., pp. 221–227, color plate 518, Apr. 1996.
Musthaler, Linda, "A Tall Order for Document Managers", Network World, pp. 35–40, Jul. 18, 1994.
Seiden, Peggy, et al., "Information Retrieval Systems for Microcomputers", Library Hi Tech, vol. 3, Issue 1, pp. 41–54, (1985).
"Zylab Includes Auto–Indexing in Zyindex 3.0", InfoWorld, vol. 10, Issue 18, p. 16, May 2, 1988.
Kappes, Sandra, et al., "Document Management for the Knowledge Worker System", US Army Corps of Engineers, USACERL ADP Report 95/38, Nov. 15, 1995.
"Verity K2 Toolkit Search System Administration Guide V2.2", Verity Incorporated, Jul. 20, 2000.
Cleveland, Gary, "Overview of Document Management Technology", International Federation of Library Associations and Institutions Universal Dataflow and Telecommunications Core Programme, Jun. 1995.
Factor, Michael, et al., "The Trellis Architecture for Intelligent Monitors", Department of Computer Science, Yale University, 1990.
Bush, Vannevar, "As We May Think", Life Magazine, Sep. 10, 1945.
Lansadale, M. W., et al., "Memoirs: A personal Multimedia Information System", Elseiver Science Publishers B. V., 1989.
Lansadale, M. W., et al., "Memoirs: A personal Multimedia Information System", 1995.
Bush, Vannevar, "As We May Think", The Atlantic Monthly, Jul. 1945.
"On Location Manual, v. 1.0", On Technology, Inc., 1990.
Coolidge, C. M., "Dogs Playing Poker (His Station and Four Aces)" (image), 1903.
Musthaler, Linda, "DMS's getting mix–and–match wardrobe," Network World, pp. 38–41, Jan. 8, 1996.
Kullberg, Robin Lee, "Dynamic Timelines: Visualizing Historical Information in Three Dimensions", Massachusetts Institute of Technology, 1995.
Cowart, Mastering Windows 3.1, 1992, chapter 12, pp. 396–417, Jan. 1, 1992.
Freeman, Lifestreams for the Newton, Steve Mann's Developers Corner, Oct. 31, 1995, 5 pages.
Freeman et al., Lifestreams: Organizing your Electronic Life, AAAI Fall Symposium, AI Applications in Knowledge Navigation and Retrieval, Nov. 30, 1995, Cambridge, MA, 6 pages.
Landsdale, M., The psychology of Personal Information Management, Applied Ergonomics, (Mar. 1988): pp. 55–66.
Malone, Thomas W., How Do People Organize Their Desks? Implications For The Design Of Office Information Systems, ACM Transaction On Office Systems, vol. 1, No. 1 (Jan. 1983): pp. 99–112.
Nelson, Theodor H., The Right Way to Think About Software Design. In The Art of Human–Computer Interface Design, Brenda Laurel, (Ed.), (1990): pp. 235–243.
Steinberg, Lifestreams, Wired 5.02, Feb. 28, 1997, 11 pages.
Getting Results with Microsoft Outlook, copyright Microsoft Corporation 1995–1996, pp. 28–29, Jan. 1, 1995.
Gobel, David P., Using Lotus Magellan, Que Corporation (1989).
Retrospect User's Guide, version 3, first edition, Dantz Development Corporation (1989–1995).
Rose, Daniel E., et al., "Content Awareness in a File System Interface: Implementing the 'Pile' Metaphor for Organizing Information", ACM–SIGIR '93 (1993).
Mander, Richard, et al., "A 'Pile' Metaphor for Supporting Casual Organization of Information", ACM–CHI '92 (May 3–7, 1992).
HyperCard Basics and HyperCard Stack Design Guidelines, Apple Computer, Inc. (1990).
"The 'Lifestreams' Approach to Reorganizing the Information World", Yaleu/DSC/TR–1070 (1995).
Gifford, David K., et al., "Semantic File Systems", ACM–SFS '91 (1991).
"On Location 2.0.01" by ON Technology (1990–1991).
Bolour, A., et al., "The Role of Time in Information Processing: A Survey", ACM–SIGART Bulletin, Issue 80 (Apr. 1982).
Rahm, Erhard, "Recovery Concepts for Data Sharing Systems" IEEE (1991).
Inside Macintosh, Addison–Wesley Publishing Company (Aug. 1992).
O'Toole, James W., et al., "Names should mean What, not Where" (1992).
Staples, Loretta, "Representation in Virtual Space: Visual Convention in the Graphical User Interface", ACM–Interchi '93 (Apr. 24–29, 1993).
Goldberg, David., et al., "Using collaborative filtering to weave an information Tapestry", Association for Computing Machinery, Inc. (1992).*
Gelernter, David, "The Cyber–Road Not Taken", The Washington Post (Apr. 3, 1994).*
Retrospect User's Guide, version 3 first edition, Dantz Development Corp., Orinda, CA, 1989–1995.
Lotus Magellan Explorer's Guide, Lotus Development Corporation, Cambridge, MA, 1989.
David P. Gobel, Using Lotus Magellan, Que Corporation, Carmel, IN, 1989.
Inside Macintosh, Addision–Wesley Publishing Company, Reading, Massachusetts, Aug. 1992.

* cited by examiner

EX PARTE REEXAMINATION CERTIFICATE ISSUED UNDER 35 U.S.C. 307

THE PATENT IS HEREBY AMENDED AS INDICATED BELOW.

Matter enclosed in heavy brackets [ ] appeared in the patent, but has been deleted and is no longer a part of the patent; matter printed in italics indicates additions made to the patent.

AS A RESULT OF REEXAMINATION, IT HAS BEEN DETERMINED THAT:

The patentability of claims **1-6**, **9-17**, **20**, **22** and **25-29** is confirmed.

New claims **34-47** are added and determined to be patentable.

Claims **7**, **8**, **18**, **19**, **21**, **23**, **24** and **30-33** were not reexamined.

34. A method as in claim 13 including displaying on a display device only the data units that are within a segment of a stream selected from the main stream and the at least one substream, without at the same time displaying data units that are within the selected stream but not within said segment, wherein the displayed data units maintain a time order of the data units as in the selected stream, and the segment corresponds to a selected interval of time shorter than the time interval of time-ordered data units in the entire selected stream, said displaying taking place while the entire selected stream persists, and said segment moving toward an earlier or a later time in response to an input from a user interface to thereby change the set of data units being displayed.

35. The method as in claim 34 wherein said input from the user interface comprises a manipulation of a cursor relative to a scroll bar or slide on the display device to thereby move the selected segment toward an earlier or later time in the selected stream and thus display on the display device a different set of data units from the selected stream.

36. The method as in claim 35 in which said displaying comprises displaying on the display device a time-based identification of the time interval corresponding to the chronological indicators of the concurrently displayed data units that are within said segment.

37. The method of claim 36 in which said displaying comprises displaying said data units as representations thereof in a stack of partly overlapping representations using perspective to create an illusion of increasing distance from a viewpoint implied by the displayed representations, wherein the representations at one place in the stack appear further away than the representations at another place in the stack.

38. The method of claim 37 in which said displaying comprises including in the displayed data units at least one data unit associated with a future time.

39. The method as in claim 34 in which said displaying comprises displaying on the display device a time-based identification of the time interval corresponding to the chronological indicators of the concurrently displayed data units that are within said segment.

40. The method of claim 39 in which said displaying comprises displaying said data units as representations thereof in a stack of partly overlapping representations using perspective to create an illusion of increasing disptance from a viewpoint implied by the displaying representations, wherein the representations at one place in the stack appear further away than the representations at another place in the stack.

41. The method of claim 40 in which said displaying comprises including in the displayed data units at least one data unit associated with a future time.

42. The method of claim 13 including displaying on a display device at least some of the data units of a stream selected from one of said main stream and said at least one substream as data unit representations in a stack of partly overlapping representations using perspective to create an illusion of increasing distance from a viewpoint implied by the displayed representations, wherein the representations at one place in the stack appear further away than the representations at another place in the stack.

43. The method of claim 42 in which said displaying comprises including in the displayed data units at least one data unit associated with a future time .

44. The method of claim 13 including selectively displaying on a display device at least some of said data units as a receding, foreshortened stack of partly overlapping data unit representations so that only a part of each of said representations in the displayed stack, after the first document in the stack, is visible to a user.

45. A method as in claim 44 in which said displaying comprises displaying only the data units that are within a segment of a stream selected from the main stream and the at least one substream, without at the same time displaying data units that are within the selected stream but not within said segment, wherein the displayed data units maintain a time order of the data units as in the selected stream, and the segment corresponds to a selected interval of time shorter than the time interval of time-ordered data units in the entire selected stream, said displaying taking place while the entire selected stream persists, and said segment moving toward an earlier or a later time in response to an input from a user interface to thereby change the set of data units being displayed.

46. A system as in claim 1 in which the means for generating a main stream of data units and at least one substream, receiving data units from other computer systems, gernerating data units, selecting a timestamp, including each data unit in the main stream, and maintaining comprise a programmed computing platform including at least one of a personal computer and a server.

47. A system as in claim 25 in which the means for generating a main stream of data units and at least one substream, allocating, including, maintaining, representing, selecting which data units are represented on the display device, and selecting one or more of the document representations comprise a programmed computing platform including at least one of a personal computer and a server.

* * * * *

US006006227C2

(12) **EX PARTE REEXAMINATION CERTIFICATE** (8770th)

United States Patent
Freeman et al.

(10) **Number: US 6,006,227 C2**
(45) **Certificate Issued: Dec. 20, 2011**

(54) **DOCUMENT STREAM OPERATING SYSTEM**

(75) Inventors: **Eric Freeman**, Branford, CT (US); **David H. Gelernter**, Woodbridge, CT (US)

(73) Assignee: **Mirror Worlds, LLC**, Tyler, TX (US)

Reexamination Request:
No. 90/011,605, Mar. 29, 2011

Reexamination Certificate for:

Patent No.:	**6,006,227**
Issued:	**Dec. 21, 1999**
Appl. No.:	**08/673,255**
Filed:	**Jun. 28, 1996**

Reexamination Certificate C1 6,006,227 issued Jun. 7, 2011

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/695**; 707/752; 707/783; 707/914; 707/916; 707/917; 707/999.002; 707/999.007; 707/999.102; 707/E17.008; 707/E17.058

(58) **Field of Classification Search** ..... None
See application file for complete search history.

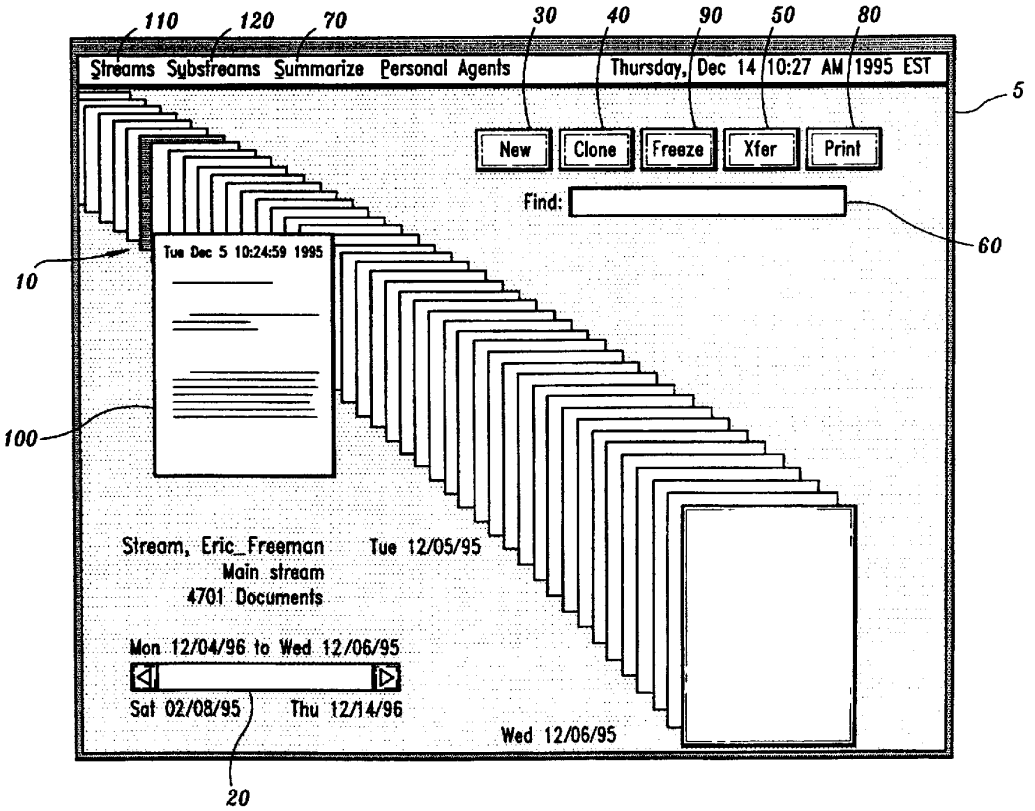
(56) **References Cited**

To view the complete listing of prior art documents cited during the proceeding for Reexamination Control Number 90/011,605, please refer to the USPTO's public Patent Application Information Retrieval (PAIR) system under the Display References tab.

Primary Examiner—Christopher E Lee

(57) **ABSTRACT**

A document stream operating system and method is disclosed in which: (1) documents are stored in one or more chronologically ordered streams; (2) the location and nature of file storage is transparent to the user; (3) information is organized as needed instead of at the time document is created; (4) sophisticated logic is provided for summarizing a large group of related documents at the time a user wants a concise overview; and (5) archiving is automatic. The documents can include text, pictures, animations, software programs or any other type of data.



EX PARTE REEXAMINATION CERTIFICATE ISSUED UNDER 35 U.S.C. 307

THE PATENT IS HEREBY AMENDED AS INDICATED BELOW.

Matter enclosed in heavy brackets [ ] appeared in the patent, but has been deleted and is no longer a part of the patent; matter printed in italics indicates additions made to the patent.

AS A RESULT OF REEXAMINATION, IT HAS BEEN DETERMINED THAT:

The patentability of claims **13** and **22** is confirmed.

New claims **48-55** are added and determined to be patentable.

Claims **1-12**, **14-21** and **23-47** were not reexamined.

48. The method of claim 13, further comprising displaying only document representations that are within a selected segment of said at least one substream, without at the same time displaying document representations of data units that also are within the substream but are not within the selected segment, wherein the displayed document representations maintain a time order according to said chronological indicators, and the displayed segment corresponds to a selected interval of time shorter than the time interval of the substream, said displaying of the segment taking place while the entire substream persists, and said displaying comprises moving said displayed segment toward an earlier or a later time in response to an input from a user interface to thereby change the set of data units that are being displayed as document representations.

49. The method as in claim 48 wherein said input from the user comprises a sliding motion by the user communicated by the user to said computer system.

50. The method of claim 49 wherein said sliding motion comprises a motion relative to a scroll bar on a computer screen.

51. The method of claim 50 wherein said motion relative to a scroll bar comprises moving a cursor relative to the scroll bar.

52. The method of claim 49 further including displaying on a display device a time-based identification of the time interval corresponding to the concurrently displayed segment.

53. The method of claim 48 wherein said displaying comprises displaying the document representations as partly overlapping document representations using perspective to create an illusion of increasing distance from a viewpoint implied by the displayed document representations, wherein the document representations at one place in the stack appear further away than the document representations toward another place in the stack.

54. The method of claim 48, further comprising displaying a glance view of a data item in response to user interaction by touching the displayed document representation that corresponds to said data item.

55. The method of claim 13 further comprising:

identifying a set of data units from said main stream or at least one substream based on search criteria provided by a user,

displaying document representations for some, but not all, of the set of data units identified in the preceding step; and

displaying at different times different plural document representations of the data units in the set in response to a user selecting through a user interface element said different document representations.

* * * * *

US007865538B2

(12) **United States Patent**
Prager et al.

(10) **Patent No.: US 7,865,538 B2**
(45) **Date of Patent: Jan. 4, 2011**

(54) **DESKTOP, STREAM-BASED, INFORMATION MANAGEMENT SYSTEM**

(75) Inventors: **Randy Prager**, New York, NY (US); **Peter Sparago**, Cheshire, CT (US); **Stephen MarcAurele**, Meriden, CT (US); **David Gelernter**, Woodbridge, CT (US); **Eric Freeman**, Branford, CT (US)

(73) Assignee: **Mirror Worlds, LLC**, Tyler, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 671 days.

(21) Appl. No.: **11/528,070**

(22) Filed: **Sep. 26, 2006**

(65) **Prior Publication Data**

US 2008/0228714 A1 Sep. 18, 2008

Related U.S. Application Data

(63) Continuation of application No. 09/892,385, filed on Jun. 26, 2001, now abandoned, and a continuation-in-part of application No. 09/398,611, filed on Sep. 17, 1999, now Pat. No. 6,638,313, which is a continuation of application No. 08/673,255, filed on Jun. 28, 1996, now Pat. No. 6,006,227.

(60) Provisional application No. 60/274,575, filed on Mar. 9, 2001, provisional application No. 60/240,480, filed on Oct. 13, 2000.

(51) **Int. Cl.**
G06F 17/30 (2006.01)
G06F 12/00 (2006.01)

(52) **U.S. Cl.** **707/828**; 707/822

(58) **Field of Classification Search** 715/526; 707/205, 10, 822, 828
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,586,035	A *	4/1986	Baker et al.	345/157
4,831,758	A	5/1989	Williams et al.	
5,031,346	A	7/1991	Herring et al.	
5,060,135	A	10/1991	Levine et al.	
5,063,495	A	11/1991	MacPhail	
5,140,676	A	8/1992	Langelaan	
5,150,410	A	9/1992	Bertrand	

(Continued)

FOREIGN PATENT DOCUMENTS

JP 6-180661 6/1994

(Continued)

OTHER PUBLICATIONS

User Guide; Mirror Worlds Technologies, Inc. (New Haven, Connecticut).

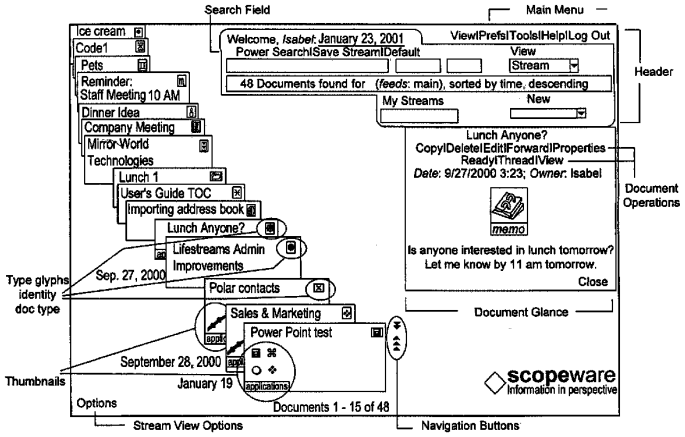
(Continued)

Primary Examiner—Kuen S Lu
(74) *Attorney, Agent, or Firm*—Cooper & Dunham LLP

(57) **ABSTRACT**

A steam-based document storage and retrieval system accepts documents that are in diverse formats and come from diverse application, automatically creates document model objects describing these documents in a consistent format and associating time stamps with the documents to automatically create a main stream in chronological order. The stream, or sub-streams meeting selected search criteria, are displayed in a variety of forms, including a receding, partly overlapping stack with aids that facilitate user interaction.

27 Claims, 11 Drawing Sheets



U.S. PATENT DOCUMENTS

5,159,669	A		10/1992	Trigg et al.
5,241,671	A		8/1993	Reed et al.
5,247,437	A		9/1993	Vale et al.
5,283,864	A		2/1994	Knowlton
5,287,448	A		2/1994	Nicol et al.
5,297,032	A		3/1994	Trojan et al.
5,299,122	A	*	3/1994	Wang et al. 707/1
5,303,361	A		4/1994	Colwell et al.
5,355,602	A		10/1994	Menke et al.
5,402,526	A		3/1995	Bauman et al.
5,430,710	A		7/1995	Mueller et al.
5,448,729	A		9/1995	Murdock
5,479,602	A		12/1995	Baecker et al.
5,499,330	A		3/1996	Lucas et al.
5,504,852	A		4/1996	Thompson-Rohrlich
5,528,739	A		6/1996	Lucas et al.
5,530,859	A		6/1996	Tobias, II et al.
5,535,063	A		7/1996	Lamming
5,542,086	A	*	7/1996	Andrew et al. 707/6
5,543,088	A		8/1996	Halbirt
5,586,237	A	*	12/1996	Baecker et al. 345/670
5,589,892	A		12/1996	Knee et al.
5,600,833	A		2/1997	Senn et al.
5,603,025	A	*	2/1997	Tabb et al. 707/2
5,613,134	A		3/1997	Lucus et al.
5,616,876	A		4/1997	Cluts
5,621,874	A		4/1997	Lucas et al.
5,621,906	A		4/1997	O'Neill et al.
5,625,818	A		4/1997	Zarmer et al.
5,649,182	A		7/1997	Reitz
5,649,188	A		7/1997	Nomura et al.
5,701,582	A		12/1997	DeBey
5,724,567	A		3/1998	Rose et al.
5,729,730	A		3/1998	Wlaschin et al.
5,737,737	A	*	4/1998	Hikida et al. 707/104.1
5,758,324	A		5/1998	Hartman et al.
D395,297	S		6/1998	Cheng et al.
5,764,972	A		6/1998	Crouse et al.
5,778,364	A		7/1998	Nelson
5,784,620	A	*	7/1998	Isham 719/316
D398,299	S		9/1998	Ballay et al.
5,835,129	A		11/1998	Kumar
5,835,789	A	*	11/1998	Ueda et al. 710/33
5,890,177	A		3/1999	Moody et al.
5,905,992	A		5/1999	Lucas et al.
5,912,668	A		6/1999	Sciammarella et al.
5,937,417	A	*	8/1999	Nielsen 715/207
6,006,227	A	*	12/1999	Freeman et al. 707/7
6,012,072	A		1/2000	Lucas et al.
6,012,074	A		1/2000	Lucas et al.
6,151,610	A		11/2000	Senn et al.
6,178,409	B1	*	1/2001	Weber et al. 705/79
6,199,082	B1	*	3/2001	Ferrel et al. 715/205
6,202,058	B1		3/2001	Rose et al.
6,639,876	B2		3/2001	Kalis et al.
6,243,724	B1	*	6/2001	Mander et al. 715/273
6,262,732	B1		7/2001	Coleman et al.
6,396,513	B1		5/2002	Helfman et al.
6,401,097	B1		6/2002	McCotter et al.
6,457,017	B2		9/2002	Watkins et al.
6,466,237	B1		10/2002	Miyao et al.
6,496,857	B1		12/2002	Dustin et al.
6,523,048	B2		2/2003	DeStefano
6,638,313	B1		10/2003	Freeman et al.
6,714,489	B2		3/2004	Kalis et al.
6,725,427	B2		4/2004	Freeman et al.
6,768,999	B2		7/2004	Prager et al.

FOREIGN PATENT DOCUMENTS

WO	WO2003/001345	A1	1/2003

OTHER PUBLICATIONS

Cowart, Mastering Windows 3.1, 1992, chapter 12, pp. 396-417, Jan. 1, 1992.

Freeman, Lifestreams Project Home Page 3 pages, 1994-1996, Jan. 1, 1994 (*not available*).

Freeman, Lifestreams for the Newton, Steve Mann's Developers Corner, Oct. 31, 1995, 5 pages.

Freeman et al., Lifestreams: Organizing your Electronic Life, AAAI Fall Symposium, AI Applications in Knowledge Navigation and Retrieval, Nov. 30, 1995, Cambridge, MA, 6 pages.

Landsdale, M., The psychology of Personal Information Management, Applied Ergonomics, (Mar. 1988): pp. 55-66.

Malone, Thomas W., How Do People Organize Their Desks? Implications For The Design Of Office Information Systems, ACM Transaction On Office Systems, vol. 1, No. 1 (Jan. 1983): pp. 99-112.

Nelson, Theodor H., The Right Way to Think About Software Design, In The Art of Human-Computer Interface Design, Brenda Laurel, (Ed.), (1990): pp. 235-243.

Steinberg, Lifestreams, WIRED 5.02, Feb. 28, 1997, 11 pages.

Getting Results with Microsoft Outlook, copyright Microsoft Corporation 1995-1996, pp. 28-29, Jan. 1, 1995.

User Guide; Mirror Worlds Technologies, Inc. (New Haven, Connecticut) (*not available*).

Gobel, David P., Using Lotus Magellan, Que Corporation (1989).

Retrospect User's Guide, version 3, first edition, Dantz Development Corporation (1989-1995).

Rose, Daniel E., et al., "Content Awareness in a File System Interface: Implementing the 'Pile' Metaphor for Organizing Information", ACM-SIGIR '93 (1993).

Mander, Richard, et al., "A 'Pile' Metaphor for Supporting Casual Organization of Information", ACM-CHI '92 (May 3-7, 1992).

HyperCard Basics and HyperCard Stack Design Guidelines, Apple Computer, Inc. (1990).

"The 'Lifestreams' Approach to Reorganizing the Information World", YALEU/DCS/TR-1070 (1995).

Gifford, David K., et al., "Semantic File Systems", ACM-SFS '91 (1991).

"On Location 2.0.01" by ON Technology (1990-1991).

Bolour, A., et al., "The Role of Time in Information Processing: A Survey", ACM-SIGART Bulletin, Issue 80 (Apr. 1982).

Rahm, Erhard, "Recovery Concepts for Data Sharing Systems" IEEE (1991).

Inside Macintosh, Addison-Wesley Publishing Company (Aug. 1992).

O'Toole, James W., et al., "Names should mean What, not Where" (1992).

Staples, Loretta, "Representation in Virtual Space: Visual Convention in the Graphical User Interface", ACM-Interchi '93 (Apr. 24-29, 1993).

Goldberg, David., et al., "Using collaborative filtering to weave an information Tapestry", Assocation for Computing Machinery, Inc. (1992).

Gelernter, David, "The Cyber-Road Not Taken", The Washington Post (Apr. 3, 1994).

Solomon, Gitta, "Piles Metaphor ATG" (video), Apple Computer, Inc., Mar. 1992.

"Spatial Data Management System" (video), MIT, Jan. 1980.

Milash, Brett, et al., "Lifelines: Visualizing Personal Histories" (video), University of Maryland, Apr. 1995.

Lucas, Peter, "CHI '94 Video" (video), MAYA Design Group, CHI '94, 1994.

Lucas, Peter, "Workscape Video" (video), http://www.youtube.com/watch?v=H5-T_S50Sr4 (screen capture video), (1993).

"200 Points of Light" (video), http://www.youtube.com/watch?v=H9F17JrG-SE (screen capture video), (1990).

W3C: A Little History of the World Wide Web, http://www.w3.org/History.html, (Accessed Jul. 15, 2010).
Gelernter, David, "The Cyber-Road Not Taken", The Washington Post, Apr. 3, 1994.
Bolt, Richard A., "Spatial Data-Management", Massachusetts Institute of Technology, 1979.
Lansadale, M.W., et al., "Memoirs: A personal Multimedia Information System", The Proceedings of the Fifth Conference of the British Computer Society Human Computer Interaction Specialist Group University of Nottingham, Sep. 1989.
Pemberton, Steven, "Not With a Whimper", SIGCHI, Apr. 1998.
Miller, James R., "The Apple Advanced Technology Group - An Introduction to the Special SIGCHI Bulletin Issue", SIGCHI, Apr. 1998.
Russell, Daniel M., "The ATG Knowledge Management Technologies Laboratory", SIGCHI, Apr. 1998.
Miller, James R., "An Overview of the ATG Intelligent Systems Program", SIGCHI,' Apr. 1998.
Miller, James R., et al., "From Documents to Objects: An Overview of LiveDoc", SIGCHI, Apr. 1998.
Bonura, Thomas, et al., "Drop Zones: An Extension to LiveDoc", SIGCHI, Apr. 1998.
Boguraev, Branimir, et al., "An Architecture for Content Analysis of Documents and Its Use in Information and Knowledge Tasks", SIGCHI, Apr. 1998.
Boguraev, Branimir, et al., "Dynamic Document Presentation", SIGCHI, Apr. 1998.
Nardi, Bonnie, et al., "An Online Digital Photography Course for High School Teachers", SIGCHI, Apr. 1998.
Shulka, Shilpa V., "Hit Squads & Bug Meisters", SIGCHI, Apr. 1998.
Rose, Daniel E., "Beyond Search: The Information Access Research Group at Apple", SIGCHI, Apr. 1998.
Russell, Daniel M., "User Experience Research Group", SIGCHI, Apr. 1998.
Walker, William F., Rapid Prototyping of Awareness Services Using a Shared Information Server, SIGCHI, Apr. 1998.
Bellegarda, Jerome R., "Interaction-Driven Speech Input", SIGCHI, Apr. 1998.
Machiraju, N. Rao, "The ATG Learning Communities Laboratory: An Overview", SIGCHI, Apr. 1998.
Bellamy, Rachel, et al., "Learning Conversations", SIGCHI, Apr. 1998.
Houde, Stephanie, et al., "In Search of Design Principles for Tools and Practices to Support Communication Within a Learning Community", SIGCHI, Apr. 1998.
Roddy, Brian J., et al., "Interface Issues in Text Based Chat Rooms", SIGCHI, Apr. 1998.
Spohrer, Jim, "ATG Education Research: The Authoring Tools Thread", SIGCHI, Apr. 1998.
Graves, Mike, et al., "Unfamiliar Ground: Designing Technology to Support Rural Healthcare in India", SIGCHI, Apr. 1998.
Mountford, S. Joy, "A History of the Apple Human Interface Group", SIGCHI, Apr. 1998.
Harris, Jed, et al., "Discourse Architecture", SIGCHI, Apr. 1998.
Singer, Samuel Weller, "Researches into the History of Playing Cards", Printed by T. Bensley and Son, 1816.
"Playing Poker at Egan's" (image), http://www.orww.org/Harney_Cattle/Student_Reports/Images/11-Poker_1882.jpg (ca. 1882).
"Commercial America", Philadelphia Commercial Museum, Jul. 1911.
"Eleventh Annual Report of the Trustees of the Boston State Hospital", Published by Wright & Potter Printing Co., 1920.
"System - The Magazine of Business", A.W. Shaw Company, 1921.
Ibberson, J.R., Md, "The Business Side of General Practice", Canad. M.A.J., vol. 72, Apr. 15, 1955.
Carlbom, Ingrid, et al., "Planar Geometric Projections and Viewing Transformations", Computing Surveys, vol. 10, No. 4, pp. 465-502, Dec. 1978.
Schmandt, Christopher Martin, "Pages Without Paper", Bachelor of Science Thesis at the Massachusetts Institute of Technology, Dec. 1978.
Donelson, William C., "Spatial Management of Information", Architecture Machine Group, Massachusetts Institute of Technology, Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques, 1978.
Bolt, Richard A., "Put-That-There": Voice and Gesture at the Graphics Interface, Architecture Machine Group, Massachusetts Institute of Technology, ACM, 1980.
Halasz, Frank, et al., "Analogy Considered Harmful", Association for Computing Machinery, 1981.
Bolt, Richard A., "Gaze-Orchestrated Dynamic Windows", Architecture Machine Group, Massachusetts Institute of Technology,, Cambridge Massachusetts, ACM, Computer Graphics, vol. 15, No. 3, Aug. 1981.
Williams, Gregg, "The Lisa Computer System", BYTE Publications, Inc., Feb. 1983.
Williams, Gregg, "The Apple Macintosh Computer", BYTE Publications, Inc., Feb. 1984.
Anderson, John J., "Atari 520ST: A Reborn Atari Once Again Points the Way to the Next Generation", Creative Computing, vol. 11, No. 10, Oct. 1985.
Greenberg, Saul, et at, "Issues and Experiences in the Design of a Window Management System", Canadian Information Processing Society - Edmonton Conference, 1986.
Henderson, D. Austin, Jr., et al., "Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface", ACM Trans. On Graphics, vol. 5, No. 3, pp. 211-243, Jul. 1986.
Robek, Mary F., et al., "Information and Records Management", 3rd ed., Glencoe Publishing Co., pp. 160-163, 179-181, 192-193, 224-229, 248-251, (1987).
Lynch, Vincent, "American Jukebox - The Classic Years", Chronicle Books, pp. 36, 74-75, (1990).
Robertson, G. G., et al., "Cone Trees: Animated 3D Visualizations of Hierarchical Information", UIR-R-1991-06, (1991).
Card, S. K., et al., "The Information Visualizer: An Information Workspace", UIR- R-1991-01, (1991).
MacKinlay, Jock D., et al., "The Perspective Wall: Detail and Context Smoothly Integrated", UIR-R-1991-03, (1991).
Kay, Alan C., "The Early History of Smalltalk", ACM SIGPLAN Notices, vol. 28, No. 3, Mar. 1993.
Adams, Michael, et al., "Jukeboxes", Schiffer Publishing, Ltd., pp. 85-86, 103-104, 105, (1996).
Card, Stuart K., et al., "The WebBook and the Web Forager: An Information Workspace for the World-Wide Web", CHI '96 Conference on Human Factors in Computing Systems, Apr. 13-18, 1996.
Meyers, Brad A., "A Brief History of Human-Computer Interaction Technology", Interactions, pp. 44-54, 1998.
Bardini, Thierry, "Bootstrapping", Stanford University Press, Chapter 6, pp. 143-181, (2000).
Sutherland, Ivan Edward, "Sketchpad: A Man-Machine Graphical Communication System", PhD Dissertation at the Massachusetts Institute of Technology, Jan. 1963.
Engelbart, Doug, "The Click Heard Round the World", Wired Magazine, 2004.
Reimer, Jeremy, "A History of the GUI", arstechnica.com, May 5, 2005.
"The Library of Congress Classification System", Aug. 2006.
Barnes, Susan B, "Alan Kay: Transforming the Computer into a Communication Medium", IEEE Annals of the History of Computing, 2007.
Calamar, Gary et al., "Record Store Days: From Vinyl to Digital and Back Again", Sterling Publishinj Co., 2009.
Masinter, Larry, "Document Management, Digital Libraries and the Web", Jun. 9, 1995.
Herot, Christopher F., "Spatial Management of Data", Computer Corporation of America, ACM Trans. On Database Systems, vol. 5, No. 4, pp. 493-514, Dec. 1980.

Newman, William M., "A System for Interactive Graphical Programming", Harvard University, ACM Spring Joint Computer Conference, 1968.
Schultz, Jan R., et al., "An Initial Operational Problem Oriented Medical Record System - for Storage, Manipulation and Retrieval of Medical Data", University of Vermont, ACM Spring Joint Computer Conference, 1971.
Apple Macintosh (screenshot image).
Carriero, Nicholas, et al., "Bauhaus Linda", Department of Computer Science, Yale University.
"Small Talk on the Alto" (image).
Donelson, William C., "Spatial Management of Data", Massachusetts Institute of Technology, 1977.
"Lotus Magellan Explorer's Guide", Lotus Development Corp., 1989.
Factor, Michael, et al., "Real-Time Data Fusion in the Intensive Care Unit", Yale University, Nov. 1991.
Gelernter, David, et al., "Graphics Interfaces & Sensor Activator Extensions for the Process Trellis Software", Yale University, Oct. 31, 1995.
"Lifestream" (image), 1995.
Harrison, Beverly L., et al., "Timelines: An Interactive System for the Collection and Visualization of Temporal Data", Proceedings of Graphic Interface '94, pp. 141-148, (1994).
"IBM OS2" (screenshot image).
Gerlenter, David, "Applications and Systems for Large Scale Adaptive Parallelism", Mar. 22, 1997.
Date, C. J., "An Introduction to Database Systems, 3rd ed., The Systems Programming Series, vol. 1", IBM Corporation, Addison-Wesley Publishing Company, 1981.
"The Billboard" (magazine), Jan. 26, 1946.
"Billboard" (magazine), Jul. 4, 1992.
"Library of Congress Classification" (accessed May 6, 2010).
Swineheart, Daniel C., et al., "A Structural View of the Cedar Programming Environment", ACM Trans. On Programming Languages and Systems, vol. 8, No. 4, Oct. 1986.
"Commodore Amiga 1000 Computer" (screenshot image).
"Atari 520ST Computer" (screenshot image).
Lucas, Peter, et al., "Workscape: A Scriptable Document Management Environment", Maya Design Group, Conference Companion CHI '94, Apr. 1994.
Bailay, Joseph M., "Designing Workscape: An Interdisciplinary Experience", Maya Design Group, Conference Companion CHI '94, Apr. 1994.
Perkins, Roderick, et al., "Inventing the Lisa User Interface", Interactions, pp. 40-53, (1997).
Feiner, Steven, et al., "An Experimental System for Creating and Presenting Interactive Graphical Documents", ACM Trans. On Graphics, vol. 1, No. 1, pp. 59-77, Jan. 1982.
Houde, Stephanie, "Iterative Design of an Interface for Easy 3-D Direct Manipulation", CHI '92, May 1992.
"Announcing Retrospect and Retrospect Remote 3.0", Business Wire, Aug. 8, 1995.
"Dantz Delivers Retrospect Archives", MacWEEK, Inc., Jun. 10, 1989.
"Archiving Easy on Retrospect: Uniform Interface to Storage Devices", MacWEEK, Inc., Feb. 21, 1989.
"Mac the Knife, GMBH Rumors", MacWEEK, Inc., Jan. 31, 1989.
"Dantz, ChrisMac drive over to CD-R", MacWEEK, Inc., Dec. 11, 1995.
"System 7.5.2 Fixes", MacWORLD, Oct. 1, 1995.
"Dantz to Waltz in with Friendlier Retrospect", MacWEEK, Inc., Jul. 31, 1995.
"Dantz Eases Network Backup", InfoWorld, Jan. 11, 1993.
"Retrospect User's Guide", 1st Ed., Dantz Development Corporation, 1989.
"Retrospect User's Guide", 2nd Ed., Dantz Development Corporation, 1993.
"Rowe CD100b Jukebox" (image).
Spence, Robert, et al., "Data Base Navigation: An Office Environment for the Professional", Behavior and Infotutation Technology, vol. 1, No. 1, pp. 43-54, (1982).
Johnson, Jeff, et al., "The Xerox Star: A Retrospective", IEEE Computer, Sep. 1989.
Shneiderman, Ben, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations", University of Maryland, Proc. Visual Languages, Sep. 1996.
Rittsteiger, Andreas, "AMI Jukeboxes" (images).
Apple Computer Inc. v. *Microsoft Corporation*, 35 F.3d 1435 (9th Cir. 1994).
Windows 1.0 (screenshot image).
Windows 2.0 (screenshot image).
Windows 3.0 (screenshot image).
Windows 95 (screenshot image).
Windows NT3.1 (screenshot image).
Plaisant, Catherine et al. "Lifelines: Visualizing Personal Histories", ACM CHI '96 Conference Proc., pp. 221-227, color plate 518, Apr. 1996.
Musthaler, Linda, "A Tall Order for Document Managers", Network World, pp. 35- 40, Jul. 18, 1994.
Seiden, Peggy, et al., "Information Retrieval Systems for Microcomputers", Library Hi Tech, vol. 3, Issue 1, pp. 41-54, (1985).
"Zylab Includes Auto-Indexing in Zyindex 3.0", InfoWorld, vol. 10, Issue 18, p. 16, May 2, 1988.
Kappes, Sandra, et al., "Document Management for the Knowledge Worker System", US Army Corps of Engineers, USACERL ADP Report 95/38, Nov. 15, 1995.
"Verity K2 Toolkit Search System Administration Guide V2.2", Verity Incorporated, Jul. 20, 2000.
Cleveland, Gary, "Overview of Document Management Technology", International Federation of Library Associations and Institutions Universal Dataflow and Telecommunications Core Programme, Jun. 1995.
Factor, Michael, et al., "The Trellis Architecture for Intelligent Monitors", Department of Computer Science, Yale University, 1990.
Bush, Vannevar, "As We May Think", Life Magazine, Sep. 10, 1945.
Lansadale, M.W., et al., "MEMOIRS: A personal Multimedia Information System", Elseiver Science Publishers B.V., 1989.
Lansadale, M.W., et al., "MEMOIRS: A personal Multimedia Information System", 1995.
Bush, Vannevar, "As We May Think", The Atlantic Monthly, Jul. 1945.
"On Location Manual, v. 1.0", On Technology, Inc., 1990.
Coolidge, C. M., "Dogs Playing Poker (His Station and Four Aces)" (image), 1903.
Musthaler, Linda, "DMS's getting mix-and-match wardrobe," Network World, pp. 38-41, Jan. 8, 1996.
Kullberg, Robin Lee, "Dynamic Timelines: Visualizing Historical Information in Three Dimensions", Massachusetts Institute of Technology, 1995.
USDC - EDTX 6:08-cv-88 *Led Mirror Worlds* v. *Apple Inc.*; Dkt# 221 Apple Inc's Motion for Summary Judgement (partially redacted).
USDC - EDTX 6:08-cv-88 *Led Mirror Worlds* v. *Apple Inc.*; Dkt# 221-25 Expert Rebuttal Report of John Levy, Ph.D. (partially redacted).
USDC - EDTX 6:08-cv-88 *Led Mirror Worlds* v. *Apple Inc.*; Expert Report of Steven K. Feiner, PhD. (confidential).
Mirror Worlds, LLC v. *Apple, Inc.* Memorandum Opinion and Order Docket Entry No. 302 dated Aug. 11, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Mirror Worlds, LLC Notice of Dismissal with Prejudice of Certain Infringement Claims Docket Entry No. 359 dated Sep. 1, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Mirror Worlds, LLC Defendant and Counterclaim Plaintiff Apple Inc's Notice of Asserted Prior Art for Trial Docket Entry No. 365 dated Sep. 6, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Memorandum Opinion and Order Docket Entry No. 385 dated Sep. 16, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Verdict form dated Oct. 1, 2010.

* cited by examiner

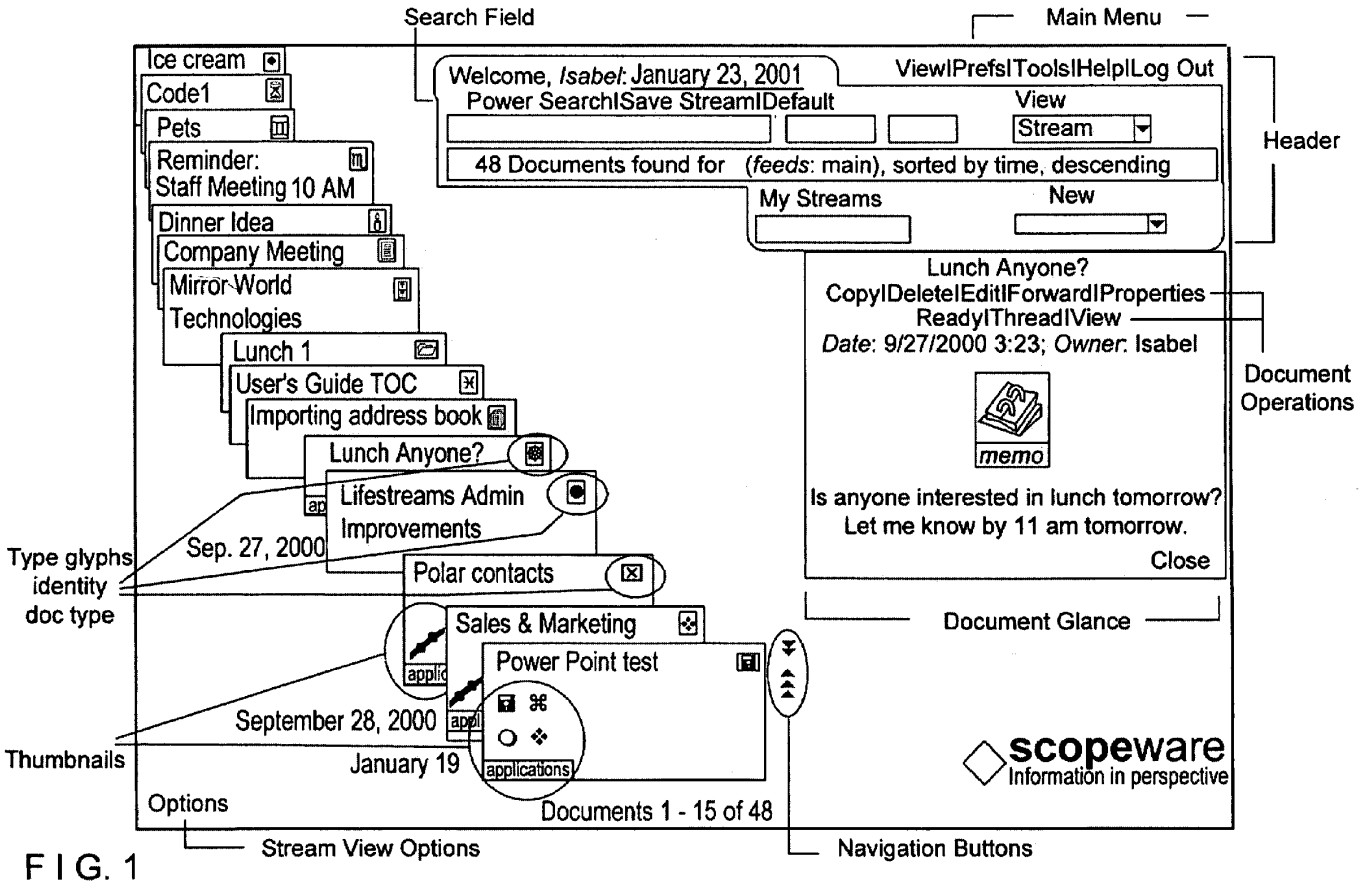
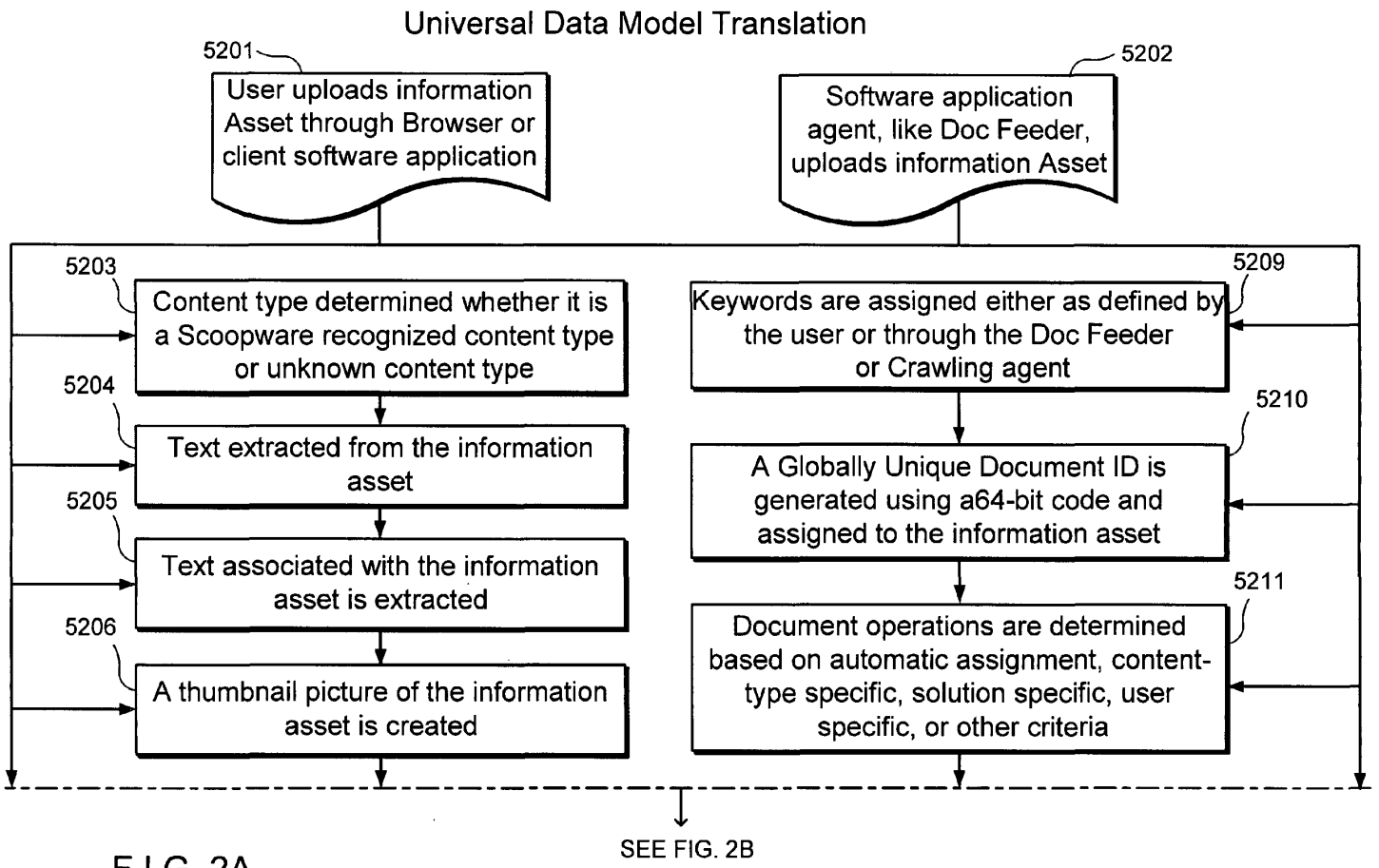
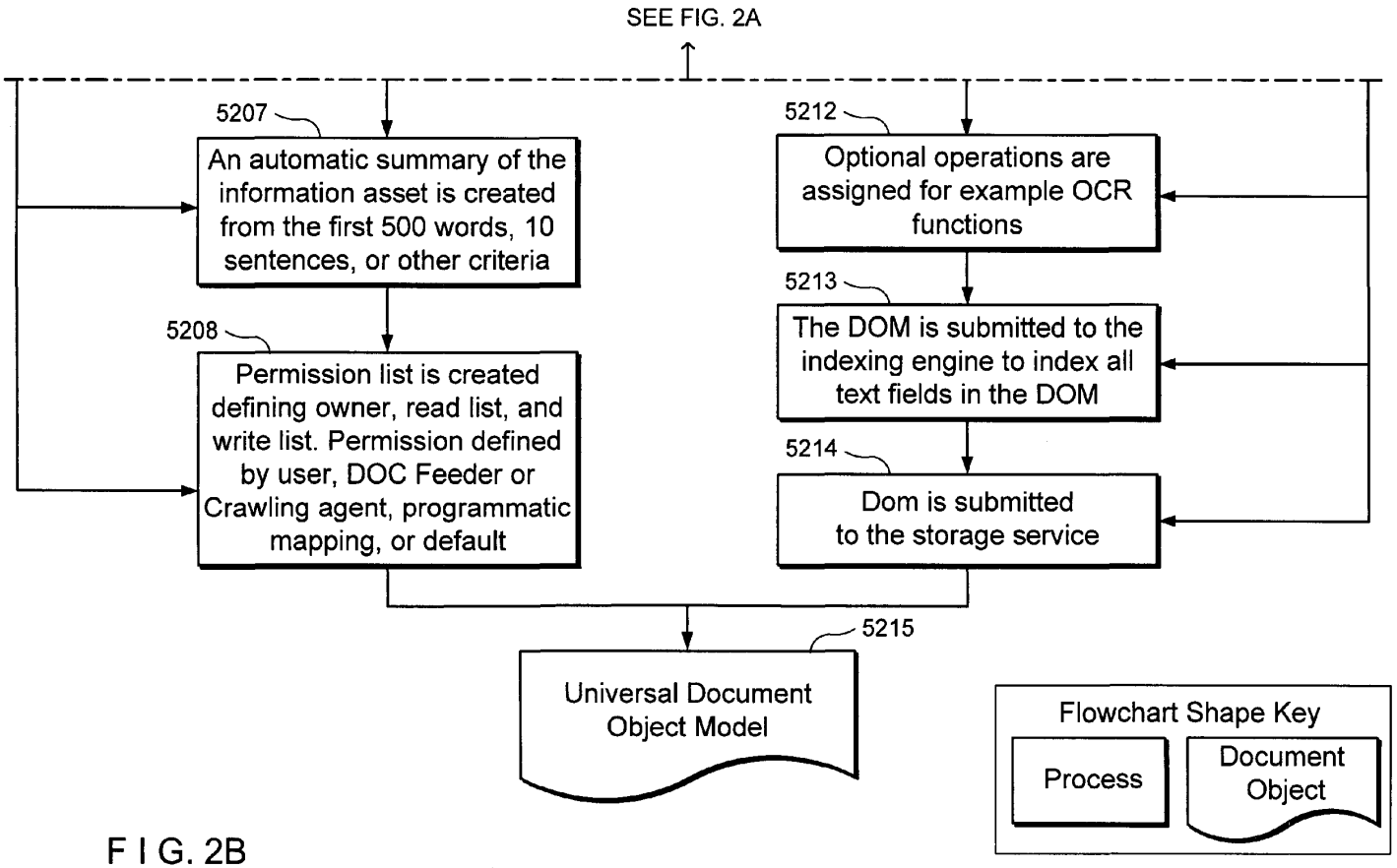

Search Field
Main Menu
Ice cream
Code1
Pets
Reminder:
Staff Meeting 10 AM
Dinner Idea
Company Meeting
Mirror World
Technologies
Lunch 1
User's Guide TOC
Importing address book
Lunch Anyone?
Lifestreams Admin
Improvements
Polar contacts
Sales & Marketing
Power Point test
Welcome, Isabel: January 23, 2001
ViewIPrefsIToolsIHelpILog Out
Power SearchISave StreamIDefault
View
Stream
48 Documents found for (feeds: main), sorted by time, descending
My Streams
New
Header
Lunch Anyone?
CopyIDeleteIEditIForwardIProperties
ReadyIThreadIView
Date: 9/27/2000 3:23; Owner: Isabel
memo
Is anyone interested in lunch tomorrow?
Let me know by 11 am tomorrow.
Close
Document Operations
Document Glance
Type glyphs
identity
doc type
Sep. 27, 2000
September 28, 2000
January 19
Thumbnails
applications
scopeware
Information in perspective
Documents 1 - 15 of 48
Options
Stream View Options
Navigation Buttons

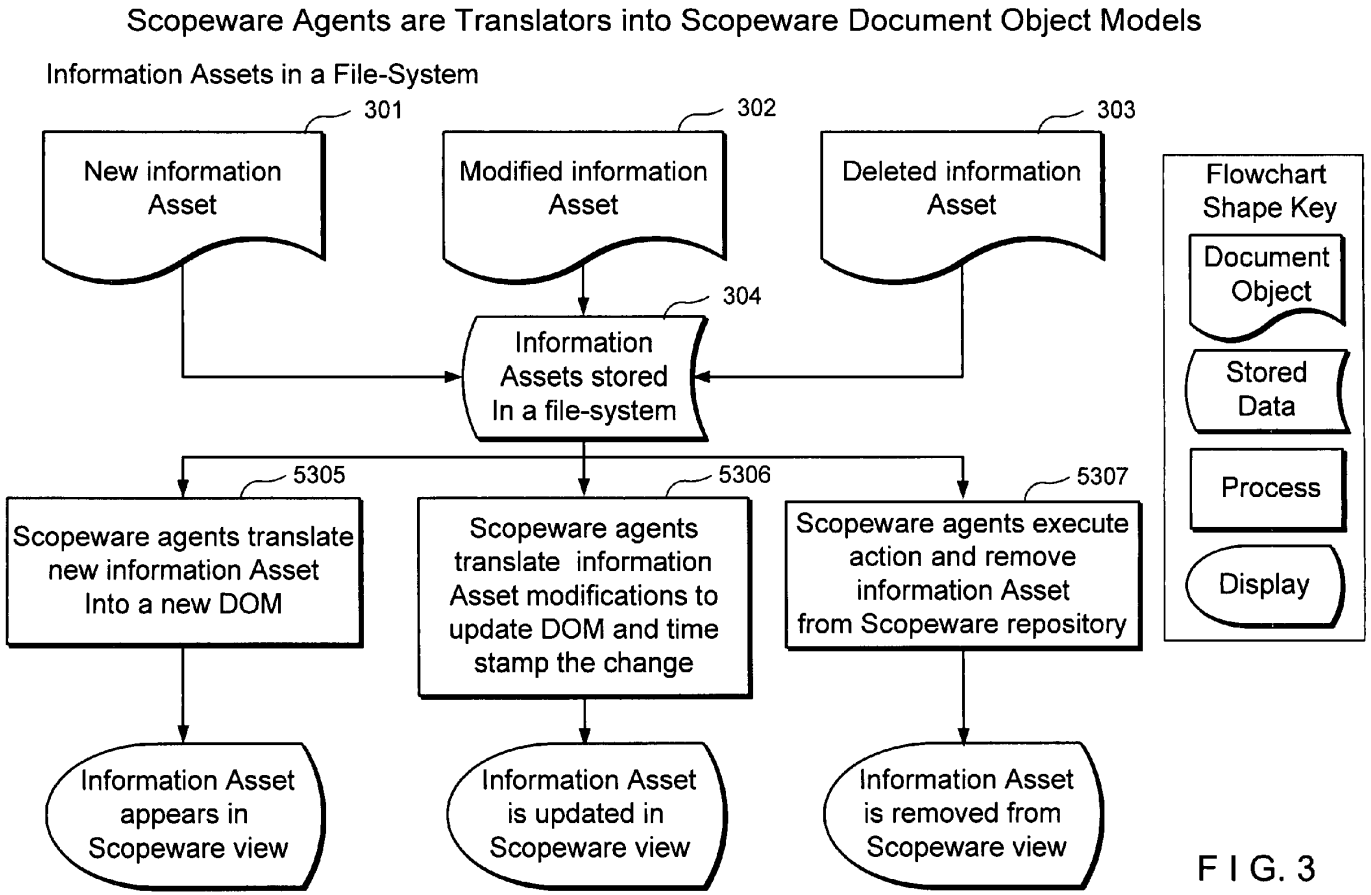
FIG. 1


Universal Data Model Translation
5201
User uploads information Asset through Browser or client software application
5202
Software application agent, like Doc Feeder, uploads information Asset
5203
Content type determined whether it is a Scoopware recognized content type or unknown content type
5204
Text extracted from the information asset
5205
Text associated with the information asset is extracted
5206
A thumbnail picture of the information asset is created
5209
Keywords are assigned either as defined by the user or through the Doc Feeder or Crawling agent
5210
A Globally Unique Document ID is generated using a64-bit code and assigned to the information asset
5211
Document operations are determined based on automatic assignment, content-type specific, solution specific, user specific, or other criteria
SEE FIG. 2B

F I G. 2A


SEE FIG. 2A
5207
An automatic summary of the information asset is created from the first 500 words, 10 sentences, or other criteria
5208
Permission list is created defining owner, read list, and write list. Permission defined by user, DOC Feeder or Crawling agent, programmatic mapping, or default
5212
Optional operations are assigned for example OCR functions
5213
The DOM is submitted to the indexing engine to index all text fields in the DOM
5214
Dom is submitted to the storage service
5215
Universal Document Object Model
Flowchart Shape Key
Process
Document Object

F I G. 2B


Scopeware Agents are Translators into Scopeware Document Object Models
Information Assets in a File-System
301
New information Asset
302
Modified information Asset
303
Deleted information Asset
304
Information Assets stored In a file-system
5305
Scopeware agents translate new information Asset Into a new DOM
5306
Scopeware agents translate information Asset modifications to update DOM and time stamp the change
5307
Scopeware agents execute action and remove information Asset from Scopeware repository
Information Asset appears in Scopeware view
Information Asset is updated in Scopeware view
Information Asset is removed from Scopeware view
Flowchart Shape Key
Document Object
Stored Data
Process
Display

F I G. 3

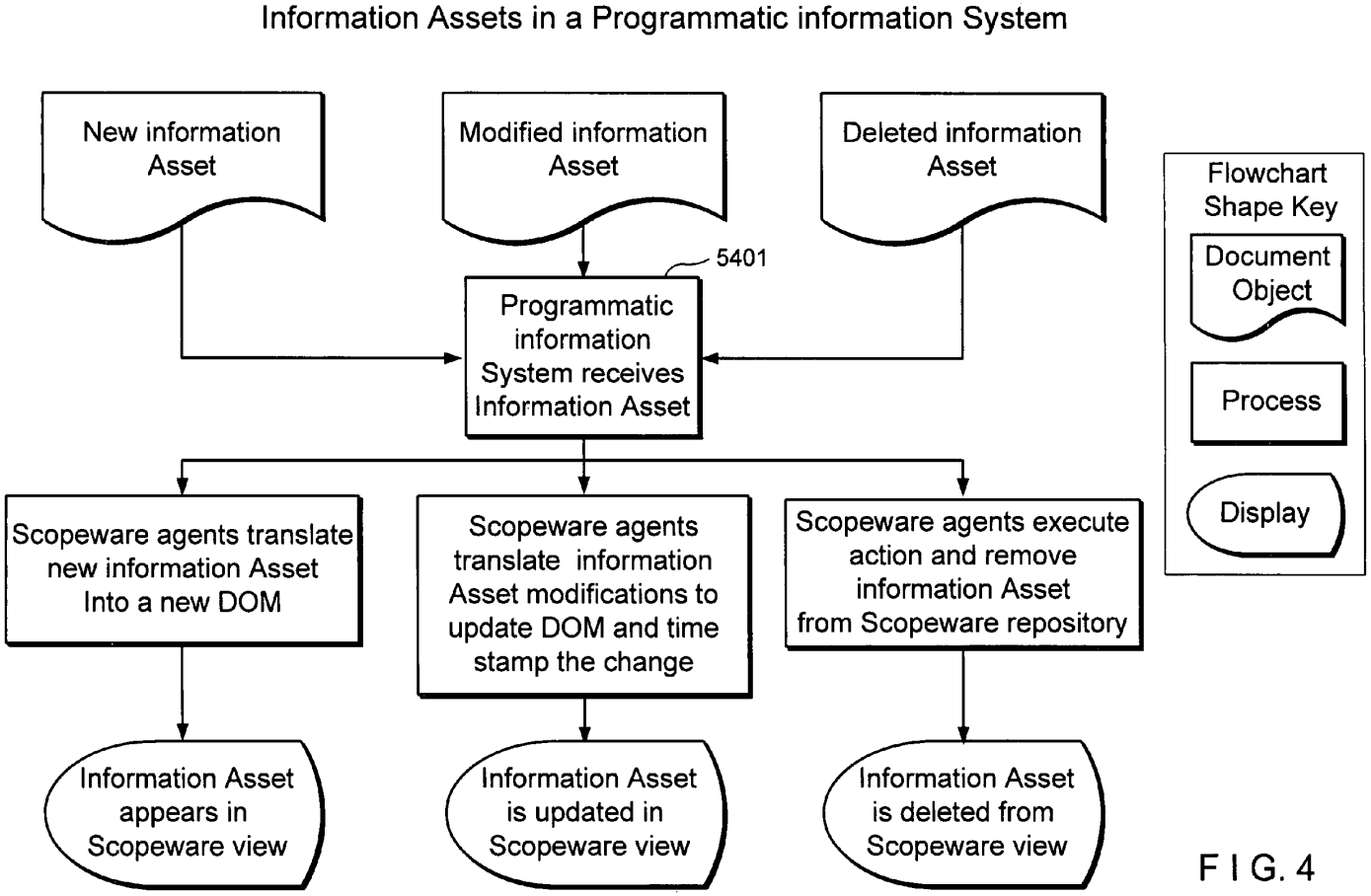
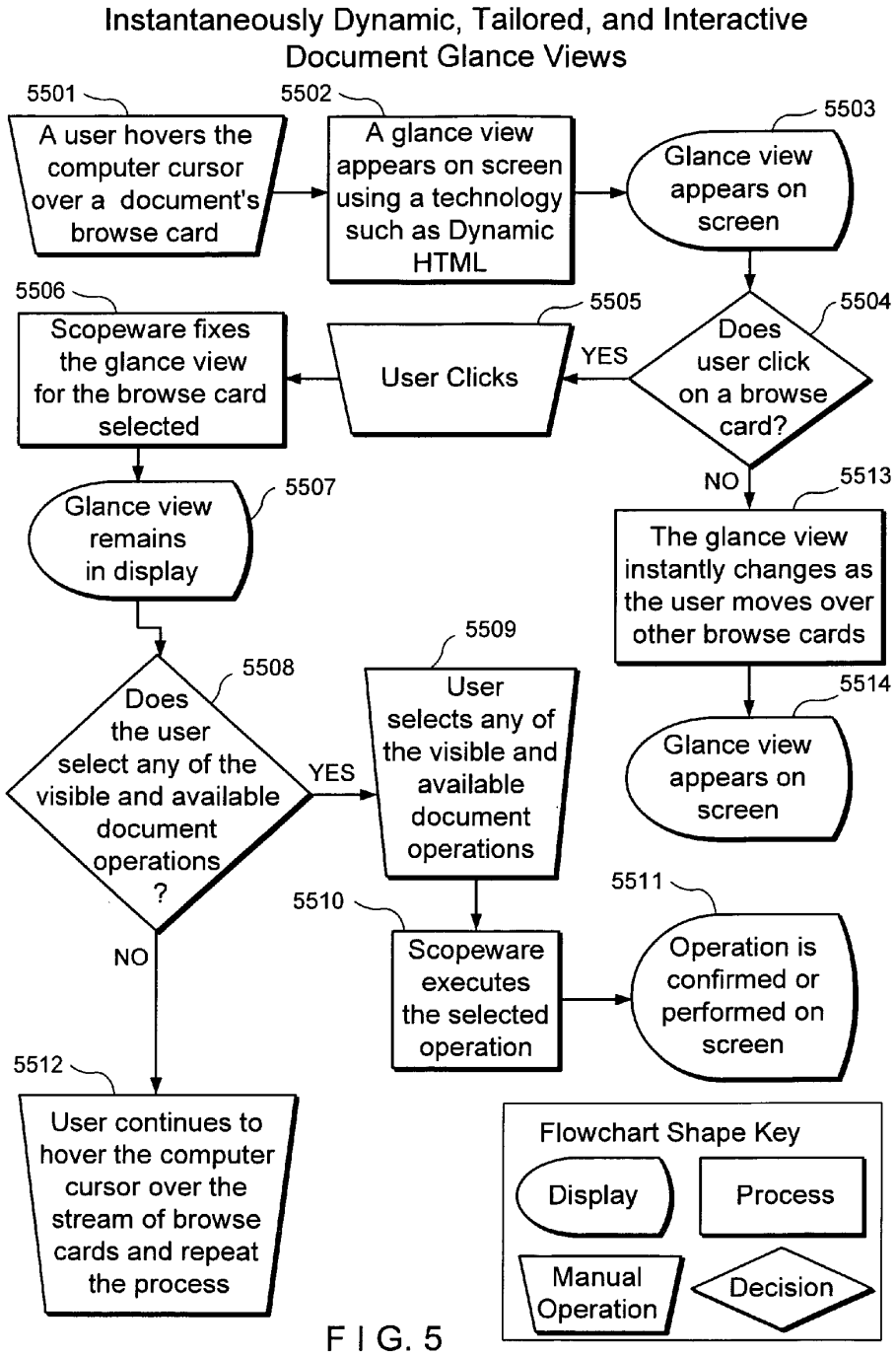

Information Assets in a Programmatic information System
New information Asset
Modified information Asset
Deleted information Asset
5401
Programmatic information System receives Information Asset
Scopeware agents translate new information Asset Into a new DOM
Scopeware agents translate information Asset modifications to update DOM and time stamp the change
Scopeware agents execute action and remove information Asset from Scopeware repository
Information Asset appears in Scopeware view
Information Asset is updated in Scopeware view
Information Asset is deleted from Scopeware view
Flowchart Shape Key
Document Object
Process
Display

FIG. 4


Instantaneously Dynamic, Tailored, and Interactive Document Glance Views
5501
A user hovers the computer cursor over a document's browse card
5502
A glance view appears on screen using a technology such as Dynamic HTML
5503
Glance view appears on screen
5504
Does user click on a browse card?
YES
5505
User Clicks
5506
Scopeware fixes the glance view for the browse card selected
NO
5513
The glance view instantly changes as the user moves over other browse cards
5514
Glance view appears on screen
5507
Glance view remains in display
5508
Does the user select any of the visible and available document operations ?
YES
5509
User selects any of the visible and available document operations
NO
5510
Scopeware executes the selected operation
5511
Operation is confirmed or performed on screen
5512
User continues to hover the computer cursor over the stream of browse cards and repeat the process
Flowchart Shape Key
Display
Process
Manual Operation
Decision

F I G. 5

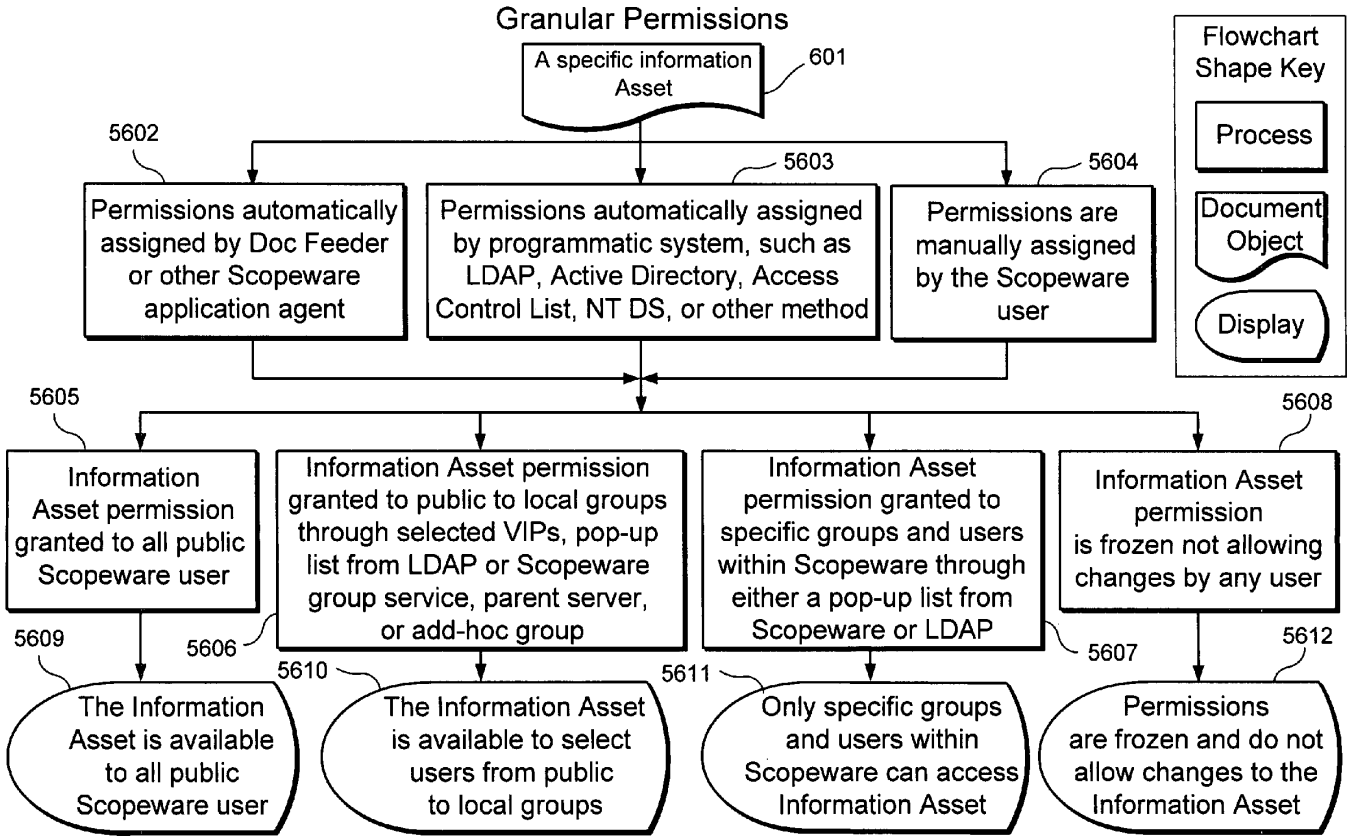

Granular Permissions
A specific information Asset
601
Flowchart Shape Key
Process
Document Object
Display
5602
Permissions automatically assigned by Doc Feeder or other Scopeware application agent
5603
Permissions automatically assigned by programmatic system, such as LDAP, Active Directory, Access Control List, NT DS, or other method
5604
Permissions are manually assigned by the Scopeware user
5605
Information Asset permission granted to all public Scopeware user
5606
Information Asset permission granted to public to local groups through selected VIPs, pop-up list from LDAP or Scopeware group service, parent server, or add-hoc group
5607
Information Asset permission granted to specific groups and users within Scopeware through either a pop-up list from Scopeware or LDAP
5608
Information Asset permission is frozen not allowing changes by any user
5609
The Information Asset is available to all public Scopeware user
5610
The Information Asset is available to select users from public to local groups
5611
Only specific groups and users within Scopeware can access Information Asset
5612
Permissions are frozen and do not allow changes to the Information Asset

FIG. 6

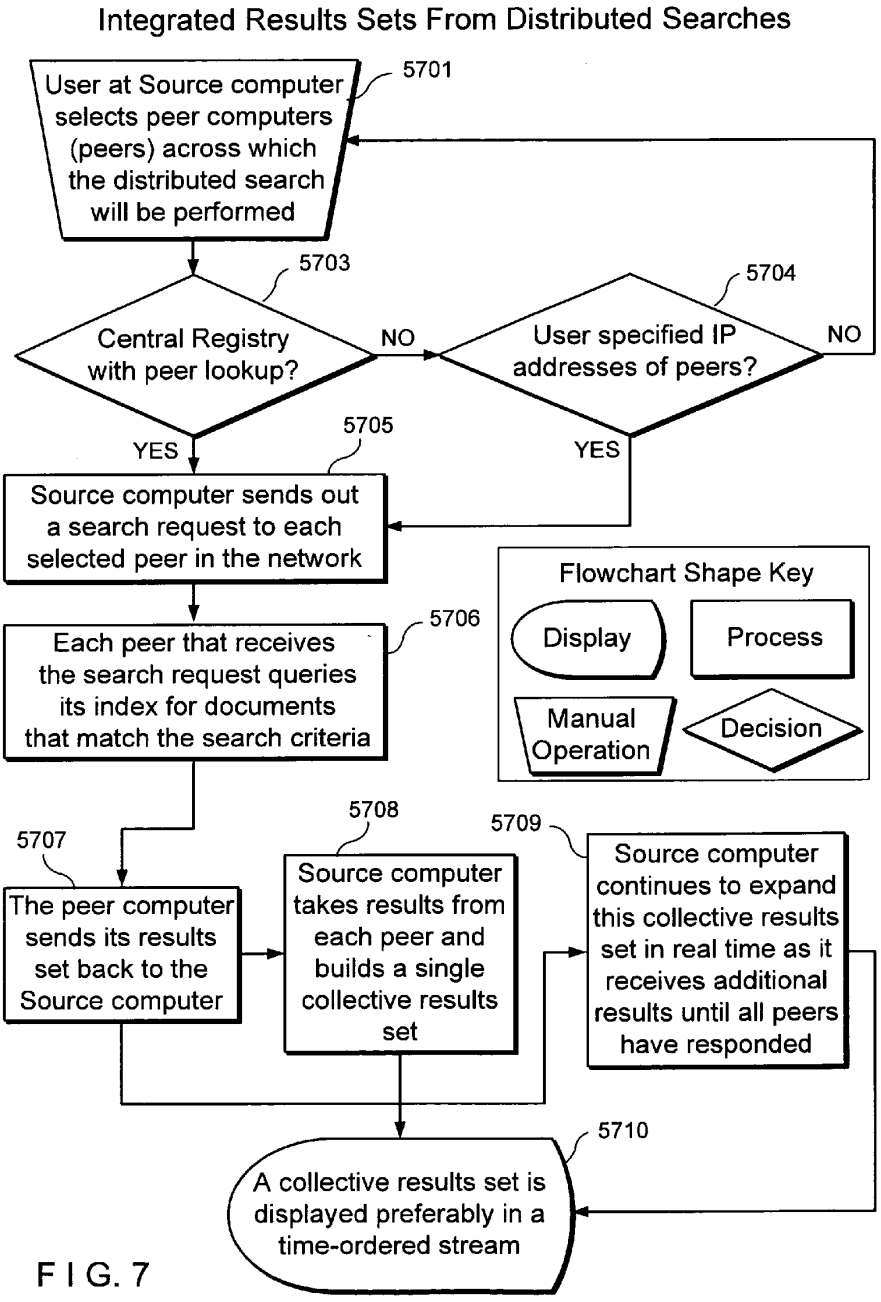

Integrated Results Sets From Distributed Searches
User at Source computer selects peer computers (peers) across which the distributed search will be performed
5701
5703
Central Registry with peer lookup?
NO
5704
User specified IP addresses of peers?
NO
YES
YES
5705
Source computer sends out a search request to each selected peer in the network
Flowchart Shape Key
Display
Process
Manual Operation
Decision
5706
Each peer that receives the search request queries its index for documents that match the search criteria
5707
The peer computer sends its results set back to the Source computer
5708
Source computer takes results from each peer and builds a single collective results set
5709
Source computer continues to expand this collective results set in real time as it receives additional results until all peers have responded
5710
A collective results set is displayed preferably in a time-ordered stream

FIG. 7

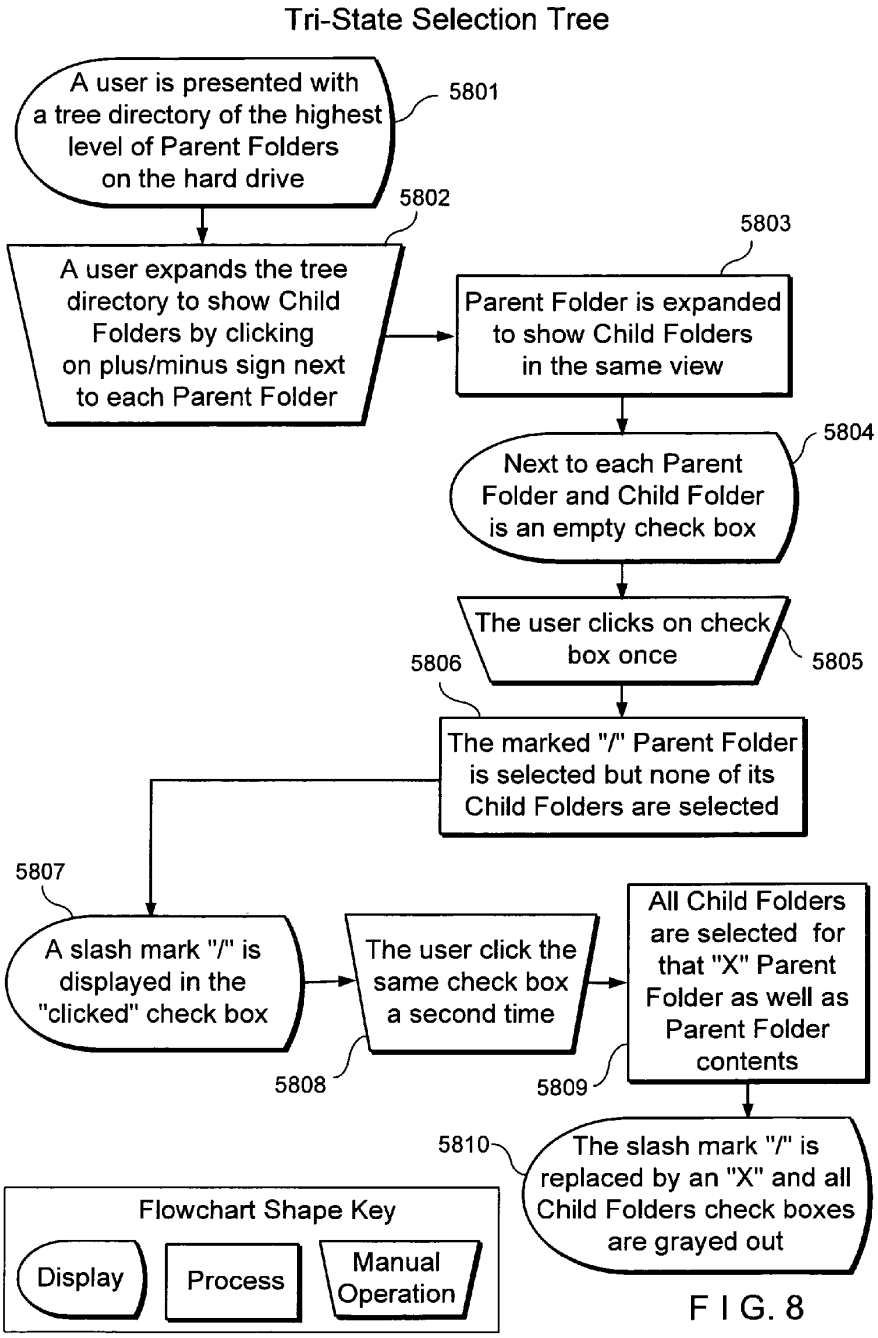

Tri-State Selection Tree
A user is presented with a tree directory of the highest level of Parent Folders on the hard drive
5801
5802
A user expands the tree directory to show Child Folders by clicking on plus/minus sign next to each Parent Folder
5803
Parent Folder is expanded to show Child Folders in the same view
5804
Next to each Parent Folder and Child Folder is an empty check box
The user clicks on check box once
5805
5806
The marked "/" Parent Folder is selected but none of its Child Folders are selected
5807
A slash mark "/" is displayed in the "clicked" check box
The user click the same check box a second time
5808
All Child Folders are selected for that "X" Parent Folder as well as Parent Folder contents
5809
5810
The slash mark "/" is replaced by an "X" and all Child Folders check boxes are grayed out
Flowchart Shape Key
Display
Process
Manual Operation

FIG. 8

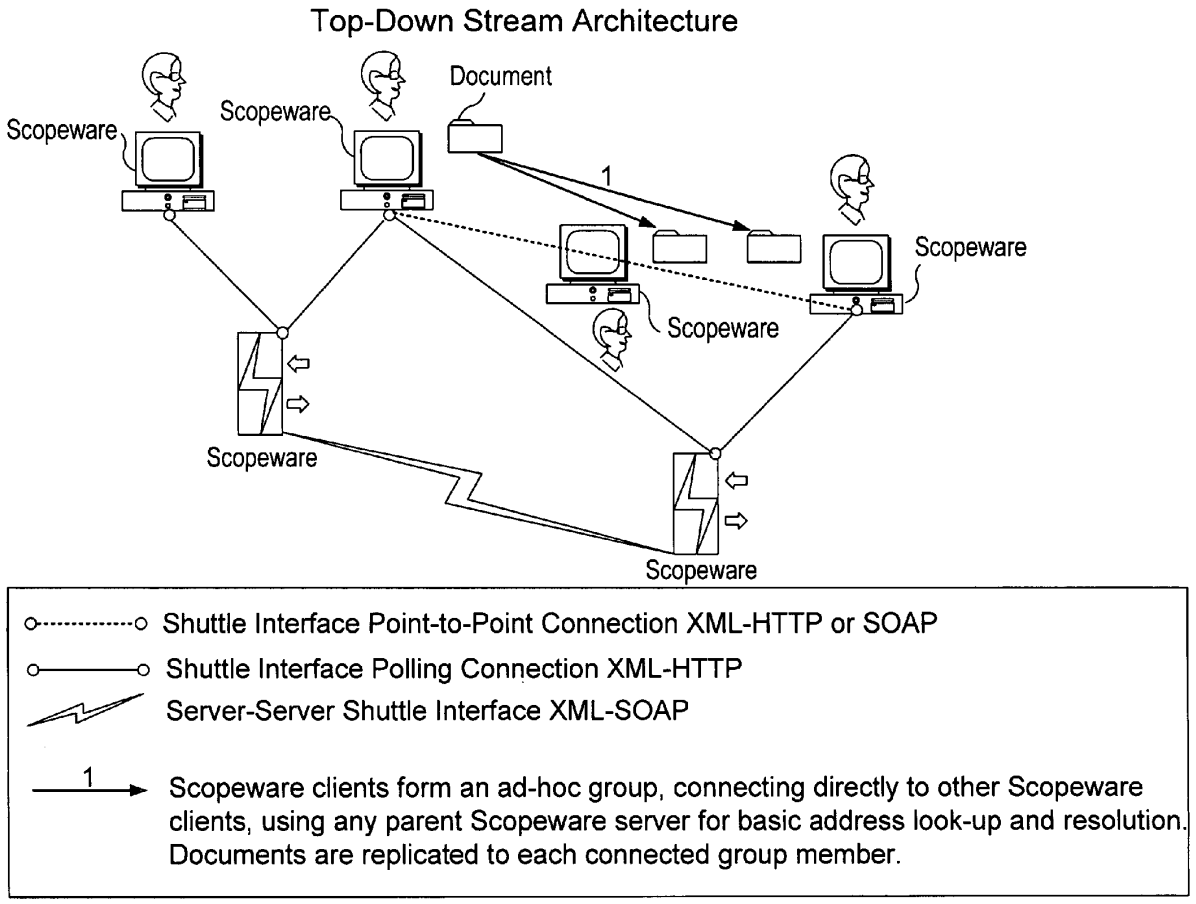

Top-Down Stream Architecture
Scopeware
Scopeware
Document
1
Scopeware
Scopeware
Scopeware
Scopeware
Shuttle Interface Point-to-Point Connection XML-HTTP or SOAP
Shuttle Interface Polling Connection XML-HTTP
Server-Server Shuttle Interface XML-SOAP
1
Scopeware clients form an ad-hoc group, connecting directly to other Scopeware clients, using any parent Scopeware server for basic address look-up and resolution. Documents are replicated to each connected group member.

FIG. 9

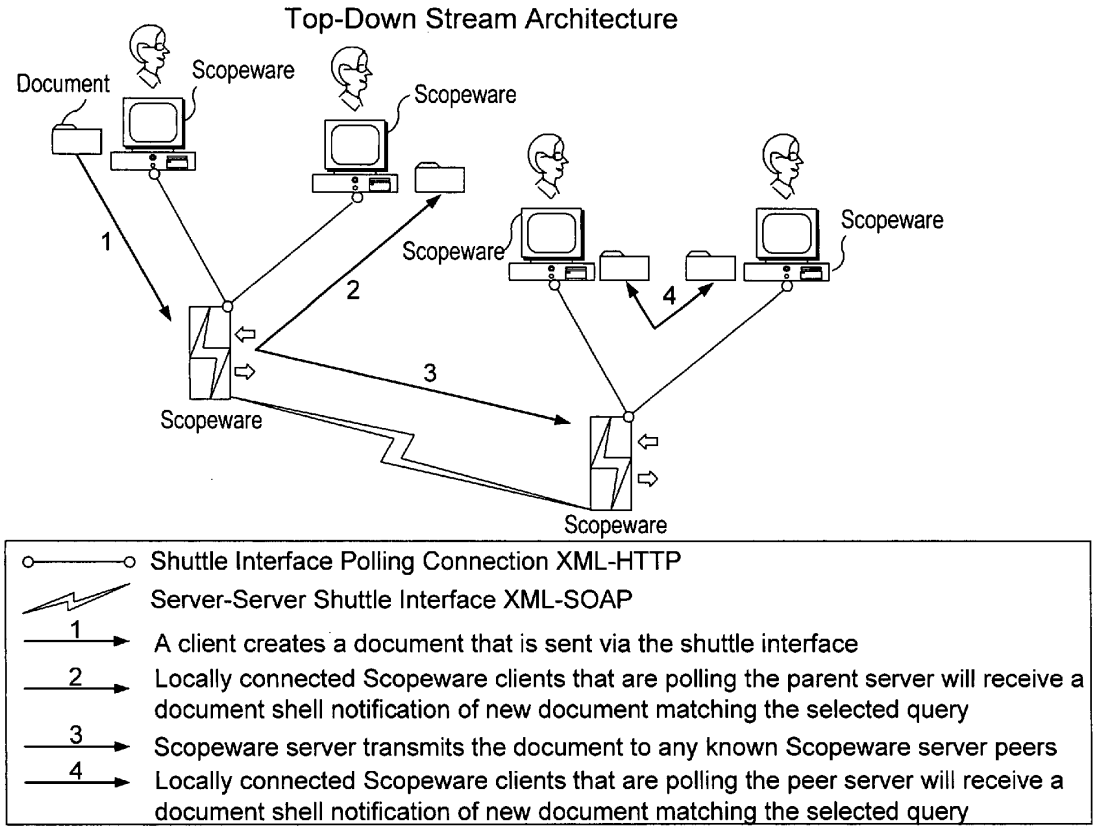

Top-Down Stream Architecture
Document
Scopeware
Scopeware
Scopeware
Scopeware
Scopeware
Scopeware
1
2
3
4
Shuttle Interface Polling Connection XML-HTTP
Server-Server Shuttle Interface XML-SOAP
1 A client creates a document that is sent via the shuttle interface
2 Locally connected Scopeware clients that are polling the parent server will receive a document shell notification of new document matching the selected query
3 Scopeware server transmits the document to any known Scopeware server peers
4 Locally connected Scopeware clients that are polling the peer server will receive a document shell notification of new document matching the selected query

FIG. 10

DESKTOP, STREAM-BASED, INFORMATION MANAGEMENT SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

The application is a continuation of Ser. No. 09/892,385, filed Jun. 26, 2001 now abandoned, which (1) claims the benefit of provisional applications 60/274,575 filed Mar. 9, 2001 and 60/240,480 filed Oct. 13, 2000, (2) is a continuation-in-part of patent application Ser. No. 09/398,611 filed Sep. 17, 1999 now U.S. Pat. No. 6,638,313, which in turn is a continuation of patent application Ser. No. 08/673,255 filed Jun. 28, 1996 and now U.S. Pat. No. 6,006,227, and (3) hereby incorporates by reference said prior applications in their entireties, as though fully set forth herein.

INCORPORATION BY REFERENCE OF MATERIAL ON COMPACT DISC

This patent specification incorporates by reference the contents of the compact disc attached hereto in duplicate (Copy 1 and Copy 2). Each disc is labeled in accordance with Rule 1.53(e)(6), with the collective names Scopeware 2.0 and Vision 1.0. The date of creation of the files on the disc is Jun. 25, 2001. The computer code on the compact disc was generated from correspondingly named source code. The names of individuals files on the disc within these collective names, as well as the size of he individual files, are identified in the list of files attached as an appendix to this specification. The contents of the compact disc submitted herewith in duplicate and the contents of the list of files attached as an appendix to this specification are hereby incorporated by reference in this application as thought fully set forth herein.

FIELD

This patent specification is in the field of systems for handling information by computer and more specifically relates to an enhanced system for handling heterogeneous items of information to store, manage, customize, organize and/or deliver such information regardless of its source and type in particularly efficient, easy-to-use, and intuitively understood.

BACKGROUND AND SUMMARY

Traditional information management systems store and retrieve documents on the basis of attributes such as the name and storage location of a document. This, however, can get very unwieldy in typical usage, as more and more names and locations of documents become a part of the storage and retrieval scheme. Although it is possible in some cases to search or order documents by other attributes, such as content and time of creation or revision, it may still be necessary to specify which file folders, directories, or storage devices to search. If a user no longer remembers how a particular item of information was stored in a traditional system, it may be difficult or impractical to retrieve it efficiently.

In an effort to alleviate these and other concerns with traditional storage and retrieval systems, and to provide a more effective and natural approach that better fits the way people tend to work with and think of items of information, a new system described herein uses approaches that rely primarily on an intuitive, time-associated way of dealing with information. The system is stream-based in that it creates time-ordered streams of information items or assets, beginning with the oldest and continuing through current and on to

future items. An information item or asset in this system can be any type—a file, an email message, bookmark, IRL, memo, draft, scanned image, calendar note, photo, shopping list, voicemail, rolodex or business card, a video clip, etc. When a user tunes in a stream, ordinarily a receding parade of documents appears on the screen. The closest are nearest in time. When a new document arrives, for example when a new email message comes in, it appears at the head of the stream, at the front of the parade. (When a newer message arrives, it steps in front of the parade.) Further-away documents are older.

Ordinarily, a user stands at the line current in time and looks into the past, but the stream also extends into the future. If the user has a meeting next Tuesday at 10 AM, a note to that effect goes into the stream's future, and a note about a meeting Wednesday goes in the stream in front of the note about next Tuesday. Documents in the stream flow steadily onward, as time does. Documents in the future part of the stream flow toward the present; documents in the present flow toward the past. Newly arriving documents push older documents further into the past.

The receding parade of documents is an efficient way to present information on a computer screen. The display uses foreshortening for a perspective effect to pack more information into limited space. For easy browsing, when the user touches a document on the screen with the cursor, a summary of that document with a thumbnail vies appears immediately, without requiring clicking or other user action, as a browse card—a dedicated small window besides the receding parade of time ordered documents. The user controls the displayed stream with VCR-type controls, to move forward or back, to go toward or to the beginning or the end of time in the stream, to now, or to any date or time, past or future.

An item of information in a stream need not be given a name, or a designation of storage location. In a traditional system, a requirement that all documents have names can have implications beyond the necessity of inventing and remembering names. For example, emails may not have names of their own but may need to be stashed inside some other file; to search for an email the user may need to go to this special mail file and search that file. In the system disclosed here, items of information such as emails do not need to be named and can be searched along with any other types of information items.

Searches in the disclosed system can be by a combination of three methods, search, browse, and time-order.

Time-order in itself often makes it possible to locate documents. Often the user needs a document that showed up recently, this morning, or two days ago, or at some time that can be pinned down with some degree of accuracy. Time-order together with browsing through the stream (and its glance views) makes it possible to glance quickly through the documents that are from the approximate time of interest and quickly pull out the right one. (While traditional systems can time-order documents it often is difficult to intersperse in the list all recent emails, news updates, bulletin-board postings, URLs and other documents, let alone voicemail messages. Without a browse feature for a stream as disclosed herein, such a list can be of little value, whereas with browse and an all-encompassing stream that gets updated promptly with new material, one can sweep over large numbers of documents, get instance glances (summary, thumbnail, etc.) of each and find the right one fast.)

When searching in a stream in the disclosed system, the user gets a new stream—a substream. One can search on any word or phrase, as every word in every document is indexed, on document types and metadata, and on time-related data

(e.g., show me all email from last March). If the user searches for an entity called Schwartz Bottling, the new substream will the narrative or documentary history of all dealings with that entity—first contacts, subsequent internal documents or communications, reports, calendar items, and so on.

A substream in the disclosed system is in some ways similar to a folder or directory in a traditional system. Instead of a "Schwartz Bottling" folder in which the user has put documents by so naming them, he/she has created a substream with those document, and can save it for later use or create it again as needed. The substream can do all a folder can but is much more powerful than a folder. A substream collects documents automatically; the use r has to put documents in a folder by hand, one by one. A substream can persist in that it continues to trap newly created or received documents that match it. If a user looks at the "Schwartz Bottling" substream tomorrow, she/he may find it has grown to include a new email or other documents that were interspersed automatically. A substream can tell a story, and include the future. A substream is non-exclusive, in that a document can belong to many substreams. A folder in a traditional system imposes on computers many of the obsolete, irrelevant limitations of a physical filing cabinet drawer or folder. A substream is an organizational tool that can make more efficient use of computer characteristics than an analog of filing an retrieving physical documents.

One reason for the efficiency of the disclosed system is that it handles all types of different documents, or items of information, in essentially the same way, even if the document is of a type or format unknown to the system. Each document when created, received or otherwise encountered is treated consistently according to a universal Document Object Model (DOM). As described below in more detail, the system processes the document to create its Document Object Modes that includes various aids such as significant information about the document including items such as summary, type of document, thumbnail of the document, who is the document owner, who has permission to access the document, keywords, command options, time stamp, index, etc. This creation of a document's DOM is done automatically, although the user can aid the process. It can be done by a translator agent or programmatically.

The system creates a glance view or browse card of each document that has the same overall format to make searching for and working with a document more intuitive but also is specific to the documents in many ways. One important difference from traditional systems is that the browse card has command buttons that match the type of documents. While the command set for traditional systems may use the same command button set for different types of documents, in the disclosed system the command set that shows in the displayed browse card is specific to the document—it has the unique combination of command buttons that make sense for that document. The command buttons unique to the browse card can be shown on the card itself or separately.

The browse card comes on the screen automatically when the cursor is over the corresponding document in the displayed stream; the user need not take any other action such as clicking on the document or taking an action calling a program that can open or work with the document.

The universal DOM of a document is created automatically as a new document of any type is added to the basic stream of information items. It is done for any existing, legacy documents, when the system is first installed on a computer, and is done as any additional documents are created or otherwise come in. Metadata such as owner, date, access permission and keywords are created as part of this automatic process.

Access permission is a part of a document's metadata, so permission levels need have the constraints of traditional information handling systems where a group or an individual typically has access to all documents in a particular folder or directory, or has a particular type of access to a folder.

Search results are integrated into a substream, at the right place, when and as they become available. The user can start using an incomplete substream and watch it build up. If the search must extend over a number of computers or even servers, and some are unavailable at the time, the results that come in when any become available are integrated into the substream at the right places.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a screen that can serve as a default view when a software product according to a preferred embodiment is opened on a computer; the labels that are added are not normally a part of the displayed screen.

FIGS. **2-8** are flowcharts illustrating processes in an example of a preferred embodiment.

FIGS. **9** and **10** are examples of configurations in a preferred embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

FIG. **1** illustrates a default screen seen on a PC or other equipment working with the disclosed system. It can show up upon turning on the computer, or upon calling the disclosed system. As seen in FIG. **1**, the screen illustrates a receding stream of documents, with the most recent documents at the front. Passing the cursor over a document in the stream causes that document's "glance view" or "browse card" to appear on the screen. The glance view of a document is so labeled in FIG. **1**. The screen also includes the following features appropriately labeled in FIG. **1**: (a) the Search Field is an area in which the user can type one or more words for which the system will search in documents (information assets) in the displayed part of the stream and/or in additional information assets that might not be displayed; (b) the Main Menu is where the user sets preferences, finds help information, logs out, and/or performs other operations; (c) the Header contains information such as links, command buttons and choice boxes used to navigate; (d) the Stream View Options allow the user to configure the presentation of the stream of information assets; (e) the Document Glance allows quick scanning of information assets that are visible on the screen, and presentation of more detailed information on the selected information asset; (f) the Type Glyphs identify the nature of an information asset at a glance (e.g., a Word document); and (g) the Thumbnails is a graphic representation of the type of document (e.g., an audio file, an email, an event, etc.). The User Guide published by the assignee hereof (a copy is submitted concurrently with the filing of this application with an IDS form) further describes the operation of a relevant example and, together with the programs contained in the compact disc submitted herewith, provides a more detailed disclosure of a preferred embodiment.

Certain particularly novel features of the disclosed system are described below by reference to flowcharts and block diagrams. More detailed information on a particular example of implementation of these and other features of the system are evident from the software on the attached compact disc, which is the best mode known to the inventors at the time of filing this patent application.

FIG. **2** illustrates creation of a universal data object model of a documents in accordance with a preferred embodiment. This is an important part of the disclosed system that helps make possible the efficient handling of heterogeneous document types in a manner that users find easy and intuitive. A document object model (DOM) can be thought of as a document shell of the information asset (IA) that contains, anon other items, a thumbnails of the information asset, permission rights, and metadata. The DOM is created from the IA and is stored in a desktop computer and/or a server, either independently of the IA itself or with a replica (copy) of the IA. From there, the system makes the DOM (with a pointer to its IA or replicated IA) to the desktop user or to users that have access to the document through some computer connection.

As seen in FIG. **2**, the process of creating a DOM starts with the uploading at step S**201** of information assets (documents) through a browser or a client software application, or step S**202** with uploading using a software application agent called Doc Feeder in a specific embodiment of the disclosed system. At the following steps, which need not be performed in the order of their description below, a DOM of the IA is created. The IA uploaded at step S**201** or S**202** can comprise structured or unstructured data. At step S**203** the process determines the content type of the IA, e.g., if it is a type that the system recognizes. If it is, the system includes content-type specific metadata in the document's DOM: MIME/content type information, a glyph of the application that creates/views the content-type, and/or the system assigns other content-type data to the DOM shell. If step S**203** determines that the IA is an unknown content type, it assigns to the DOM a content-type for "unknown content-type.". Step S**204** extracts text from the information asset, for example, in a text document, this step extracts the text of the document. Step S**205** extracts text that may not be within but may be associated with the information asset, for example, the time stamp of the document, the owner of the document, and possibly other textual information that is or can be associated with the document. Other possible examples are attributes of the IA such as file reference path, database/repository path, file metrics such as size, encryption, other identification information, etc. Step S**206** generates a thumbnail picture of the IA. The thumbnail can be a reduced-size picture of the document, for example of the first page, and can be converted to a graphic image format. Other examples of thumbnails are JPEG, MPEG, BMP, GIF, AVI, or other still or moving image files representative of some aspect of the IA. Step S**207** produces an automatic summary of the IA, e.g., a replica of its first 500 words, or first 10 sentences, or some other information copied or otherwise derived from the IA. Step S**208** creates a permission list unique to the IA that defines the owner of the IA (e.g., its creator), and lists of people or entities and groups that can access the IA or the DOM of that IA for reading and/or writing purposes. This permission list can be defined by the user for the particular IA or for a class of IAs, or can be created automatically, e.g., by software agents called Doc Feeder or Crawling agent in a particular embodiment of the described system, or by programmatic mapping such as LDAP, Active Directory, NTDS or some other mapping. Alternatively, at least for some documents, the permission list can be default setting.

Step S**209** assigns keywords to the information asset. The software agents Doc Feeder or Crawler can assign keywords, and the user can manually assign or add keywords. Step S**210** generates and assigns to the IA a Globally Unique Document ID, e.g. as 64 bit code unique to the IA. Step S**211** determines and assigns to the IA document operations that are unique to the IA. Depending on the IA, these operations or command buttons can be basic, such as "View" and "Reply." They can be content-specific, such as "Play" for multimedia information assets. They can be solution-specific, such as "Fax" of Purchase." They can be user-specific, such as "Delete" allowed to only certain users. An important point is that the operations or command buttons assigned to a particular IA match the IA and need not be the same for different information assets, as is the typical case with traditional information management systems. Step S**212** assigns optional operations or command buttons to the IA. They include, for example, commands to send the IA to an optical character recognition (OCR) service that can be a separate service, IP, HTTP-based or an asynchronous operation. Alternatively, the optional operation can be another OCR operation that can perform OCR on a selected part of the IA, or on digital graphic portions or can involve multi-part associations. At step S**213**, the information asset is submitted to an indexing engine (asynchronous service) Again, this can be a separate service, IP, HTTP-based. This step can index all or selected fields of the IA, including but not limited to the IA summary, title, permissions, IA text, keywords, time, metadata, and content-type. At step S**214** the DOM created as described above is submitted to a storage service. This can be a database that is a file reference with a pointer to the actual location of the IA on a network or a local file system, or it can be a database that contains the actual IA in a repository such as a user's computer or a centralized repository. The document object model so generated is made available for use in step S**215**.

FIGS. **3** and **4** illustrate methods of creating document object models from information assets. As seen in FIG. **3**, three type of information assets are involved—new information assets **301**, modified information assets **301**, and deleted information assets **303**. All come to a file system **304**. At step S**305**, agents specific to the disclosed embodiment of the system known as Scopeware 2.0 translate the IA into a DOM, i.e., create a DOM shell for the IA, with attributes as discussed in connection with FIG. **2**. At step S**306**, Scopeware agents translate the IA modifications into an updated DOM and time-stamp the change so the new time-stamp becomes a part of the DOM and the modified IA can be places in the stream of documents at a place reflecting the new time-stamp. At step S**307**, Scopeware agents execute actions for removing the deleted IA from the repository of documents. The display, such as that seen in FIG. **1** reflects the actions takes at steps S**305**, S**306** and S**307**. As a result of step S**305**, the stream on the display shows at **308** the new IA (provided the time period where the new IA fits is being displayed). As a result of step S**306**, the modified IA appears at **309** in its correct place in the displayed receding stream of documents. As a result of step S**307**, the deleted documents is removed at **310** from the displayed stream, and the remaining In FIG. **4**, a programmatic information system received new, modified and deleted information assets for storage and distribution to appropriate translation agents as illustrated. In other respects, the FIG. **4** arrangement corresponds to that of FIG. **3**, so the description of corresponding portions will not be repeated.

At least some of the document object model created as described above becomes a part of a glance view or browse card of the type illustrated in FIG. **1**. An important feature of the system disclosed here is to conveniently display such a glance view in a natural and intuitively accepted way to facilitate operations.

Traditional user interfaces for computers typically present lists or graphical icons of "documents" (including but not limited to computer files, emails, web pages, images and other types of electronic information). These lists and icon displays provide only a limited amount of information about

the document—typically, title and application type only, although additional information as well in some cases. This can make it difficult for users to identify the document without downloading and/or opening the document with its associated application. For example, in Windows 2000, the user interface displays a small temporary pop-up window of the document's title, application type, author and size when the user hovers his cursor on the document icon; however, the pop-up window appears only after a brief delay, usually 1-2 seconds and is for documents that are on the screen at the time, which tend to be a small part of the many documents typically stored in or accessible through a user's computer.

In contrast, the disclosed system creates a pop-up window for heterogeneous documents of known and unknown application types that appears instantly, as perceived by the user, as he/she hovers the cursor over the document's representation in the user interface. In the example of FIG. **1**, this representation is an index card in a cascading flow of overlapping index cards (called "browse cards"), and the pop-up window is called a "glance view". This glance view not only contains the document's title, application type and owner, but also may contain rich multimedia cues (such as a thumbnail image of the first page of the document, a WAV or MP3 preview of an audio file, or an animated GIF preview of a video file), text summaries and document operations specific to the document's application type and access permissions. For example, if the user has write permission for a document, the "Edit" operation will be visible and available; however, if not, the Edit operation will not be visible or available. These document operations are interactive, allowing users to select available operations directly.

Referring to FIG. **5** for an illustration of the instantaneously dynamic, tailored, and interactive document glance view feature of the disclosed system, at S**501** a user hovers his or her computer cursor over a document's browse card. Essentially instantly, at least as perceived by the user, and without any mouse clicking or other action on the part of the user, step S**502** processes the information needed for a glance view to appear on the screen, and at S**503** the glance view appears next to the browse card, using a technology such as Dynamic HTML. If the user clicks on a document's browse card, as detected by the test at step S**504**, and as executed by the user at S**505**, step S**506** causes the glance view to become fixed and step S**507** causes it to remain in the display. The glance view does not change until the user clicks on another document's browse card. If the user does not click on any browse card, as determined by the test of step S**504**, the glance view will instantly change as the user moves his cursor over other browse cards, to reflect the glance view of the underlying browse card. If the user has clicked on a browse card to fix the glance view as a stationary window, the user can then select any of the visible and available document operations, by taking the "yes" branch of step S**508** and selecting at S**509** an available operation (as earlier described, the operations or command buttons that show are specific to the document). At step S**510** the system executes the selected operation (command) and the display reflects this at S**511**. If at step S**508** the user takes the "no" branch, she can continue to hover the cursor over the stream of browse cards and repeat the process, at step S**512**. If at S**504** the system determines that the user has not clicked to fix a glance view, the glance view information essentially instantly changes at S**513** as the user moves the cursor over other browse cards, and the new glance views appear on the screen at S**514**.

FIG. **6** illustrates a process involving another important feature of the disclosed system—granular permissions for access to information assets that allows clients to receive seamless and uniform access to contents without necessitating changes to existing network security and access rights. In traditional systems, a network administrators typically would grant access to specific network drives and file folders. The permission typically would allow a user to access the entire folder or drive, or would deny access to an entire folder or drive, rather than to a particular information asset or document.

In the disclosed system, each information asset is accessible through specific access permission for each client or designated group of clients. Examples of access stage permissions are read, write, and aware. Read permissions allow a client to view the full information asset. Write permissions allow the client to view and edit the document. Aware permission alerts the client that an information asset exists, for example by providing a document shell in the client's stream of documents, but does not allow the client to view or edit the document. A group of clients who want to collaborate on a project or event can establish a designated group that can be assigned permissions to relevant documents for the project or event. Thus, each member can receive real-time additions to his or her stream of documents and information assets are posted. The clients can assign permission to the other group members themselves, by so designating the appropriate documents to be shared, without involving a network administrator. Some documents, such as personal to-do lists, can be accessible only to a specified user, but the user can change this at any time to allow access, full or partial, to other designated persons. Assignments of permissions for access can be done as granularly as an individual client level or individual document, or as diffuse as a departmental or enterprise level.

As seen in FIG. **6**, an information asset **601** can have permission levels assigned to it in several ways. At step S**602**, a software agent such as Doc Feeder can automatically assign permissions; at step S**603** a programmatic system such as SDAP, Active Directory, Access Control Lists, NT DS, of some other system assigns permissions to the document; and/or at step S**604** the user manually assigns permissions to the document. Examples of processes relevant to different types of permissions are: step S**605** grants access to all public users of the system; step S**606** assigns permissions to groups as illustrated; step S**607** assigns permissions to specific groups as illustrated, and step S**608** freezes permissions and does not allow the document to be changed. The display, of the type illustrated in FIG. **1**, can provide information representative of the permissions, as illustrated at steps S**609** through S**612** in FIG. **6**.

Another important feature of the disclosed system is illustrated in FIG. **7** and pertains to integrating search results from distributed searches. In traditional systems, search requests in a client/server model with a central index usually return a single, well-defined results set. In a peer-to-peer network, however, search results may come back to the "Source" computer (the computer that issues the search query) in a haphazard manner because of network latency (variable traffic speed and bandwidth across a distributed network) and variable peer presence (peer computers can be turned on and off, or removed from network at times).

The disclosed system asynchronous responses to a distributed query across a peer-to-peer network of computers to integrate the results from diverse sources, arriving at different times, and comprising diverse types of documents, into a single unified results set. One preferred embodiment leverages the time-ordered presentation interface earlier described in so that search results are integrated into a time-ordered stream according to each document's original time-stamp,

regardless of when the document's search results set was received by the Source computer.

As seen in FIG. **7**, at step **701** a user at a Source computer selects peer computers ("Peers") across which the distributed search will be performed. If the test at S**703** determines that there is no central registry with peer hookup, and the test at S**704** determines there is no user-specified IP address of peers, the process returns to S**701**, where the user can specify addresses or they can be provided in some other way. The central registry with lookup of Peers can involve Online/offline status, IP/DNS resolution service and Optional public/private key authentication. When the test at S**703** or at S**704** leads to the "yes" branch, at step S**705** the Source computer sends out a search request that travels to each selected Peer in the network. At S**706**, each Peer that receives the search request queries its index for documents that match the search criteria, and at S**707** the peer computer then sends its results set back to the Source computer. The response can be XML-based, a binary byte stream, or an in-band and out-of-band transfer. At S**708** the Source computer takes the results set from each Peer and builds a single collective results set. In a preferred embodiment, this collective results set is organized as a time-ordered stream of documents, as seen in FIG. **1**. This can involves an on-the-fly browser combination with XML & XSL with time-sort algorithm, XML to presentation layer with time-sort algorithm, and in-band and out-of-band transfer. Importantly, at S**709**, the Source computer continues to expand this collective results set, essentially in real time as it receives additional results sets from Peers until all Peers have responded or some other relevant event has taken place. At S**710**, the collective results are displayed as soon as results have come in at the Source computer, and the display is updated as additional results come in, even when a Peer that was off-line comes on line and sends results at a later time.

Yet another feature of the disclosed system is a particularly convenient tri-state tree. In a single scrolling tree directory of the contents of a hard drive (or hard drives in a network), a user may want to select "Parent Folders" (folders containing subfolders) and "Child Folders" (subfolders contained within a folder) that can be further operated on. This feature allows users to select folders in one or more of the following combinations:

1. All Parent Folders and all Child Folders
2. Some Parent Folders and all their Child Folders
3. Some Parent Folders and some of their Child Folders
4. No Parent Folders and no Child Folders (the do nothing option)

This selection tree has useful application beyond the particular example of information handling disclosed here; it can be used to select folders for any computer operation. For example, it can enable users to discretely select software application or operating system components to install or remove.

A single scrolling tree directory of Parent and Child Folders that can expand and contract to show the contents of Parent and Child Folders is known—Microsoft Windows Explorer is an example of one. A Tri-State Selection mechanism also is known—Microsoft Add/Remove Windows Components is an example of another way of selecting various Parent and Child Folders. However, the Microsoft Add/Remove Windows Components feature does not display all Parent and Child Folders within a single scrolling tree directory; Child Folder and other contents of a Parent Folder are displayed in a separate window only after the user clicks on a Details button. In addition, only the contents of one Parent Folder can be displayed at a time.

The Tri-State Selection Tree described here combines the elements of a single scrolling tree directory with a tri-state selection mechanism in a new and unique way to enable users to discretely select specific Parent and/or Child Folders all in one single view.

Referring to FIG. **8** for an illustration, at step S**801** a user is first presented with a tree directory of the highest level of Parent Folders on a hard drive or network. At S**802** the user can expand the tree directory to show Child Folders by clicking on a plus/minus sign next to each Parent Folder, and the directory so expands at S**803**. At S**804**, the display shows a check box next to each Parent Folder (e.g., to the right of the plus/minus sign). By default, all check boxes are empty, indicating that no Parent or Child Folders are selected. If at step S**805** the user clicks on a check box once, the process at step S**806** selects the marked "/" Parent Folder but none of its Child Folders are selected, and step S**807** shows this on the display. If at step S**808** the user clicks the check box a second time, the slash mark is replaced by an "X" and all the Child Folders' check boxes are then selected and grayed out at S**809**, indicating that all Child Folders are selected for that Parent Folder, and this is displayed at S**810**.

Thus, by expanding the tree and clicking on check boxes, the user can systematically and efficiently select a discrete number of folders on which to perform an operation.

Yet another feature of the disclosed system is an arrangement of a redundant array of inexpensive servers (RAIS). Processing of a large set of information or document requires benefits of a centralized architecture—reliability and scalability, and RAIS is a novel approach to provide benefits of a centralized architecture—namely reliability and scalability with numerous inexpensive computers. Thus, RAIS can deliver essentially infinite scalability, can allow inexpensive smaller computers to be used to solve enterprise computational problems rather then expensive larger platforms, cheaper/faster.

For example, consider:

Set of Information, D, with specific documents D**1**, D**2**, D**3**; D{D**1**,D**2**,D**3**}

RAIS of N×N size here with N=3; RowN,ColN

Replication factor is number of columns

Scalability factor is number of rows

1. Here N=3, with 9 computers

	Col1	Col2	Col3
Row1	A	A	A
Row2	B	B	B
Row3	C	C	C

2. To post a Document, Dn, one copy is sent to a sub-server in each ColN, so

	Col1	Col2	Col3
Row1	A(Dn)	A(Dn)	A(Dn)
Row2	B	B	B
Row3	C	C	C

3. Thus Dn is replicated N times (N=3) and thus if Col**1**: Row**1** computer is unavailable there are two other computers with the same Dn. This is RAIS replication.

4. To post a universe, or set of documents, D{D1,D2,D3}, can use simple (round-robin) or complex (latency, closest path, spanning tree) routing, sending each document to a different RowN.

	Col1	Col2	Col3
Row1	A(D1)	A(D1)	A(D1)
Row2	B(D2)	B(D2)	B(D2)
Row3	C(D3)	C(D3)	C(D3)

5. Thus to reassemble the entire universe or set of documents, D, need to send a request to each RowN. To reconstruct, D, for an N×N RAIS requires N request/responses.
6. Multiple smaller requests can be used instead of one mammoth request. This reduces latency, bandwidth and process constraints. This is RAIS scalability.
7. Note that any one of the computers in Row1 can be used to re-construct the total set D found in Col1. For example, if Row1:Col1 computer is unavailable, then Row1:Col3 computer has a copy of the data. In fact, D is can be constructed from any arrangement that completes a ColN.
8. To increase either replication or scalability simply increase N.

Scopeware Software Agents, either desktops or servers, can be installed on each computer in a RAIS matrix to achieve this functionality.

The disclosed system can be implemented in a variety of ways in terms of physical information storage—for example, physical information storage can be centralized or decentralized. Decentralized storage, physical storage of information with multiple servers and/or clients, is possible through network agents called Doc Feeders, which may be located at a server or client level. The Doc Feeder allows a storage location of a client, for example a file folder on a desktop hard drive, to be included in the system level data repository for use throughout an organization or enterprise. Depending upon implementation, the Doc Feeders can replicate the information asset (IA) to a server or maintain a constant pointer to the physical storage location while populating the system with the document object model (DOM). As earlier described, a DOM is a document shell of the IA that contains, among other items, a thumbnail of the IA, permission rights, and metadata. A DOM is created from the IA and placed on the Scopeware server, either independent of the IA or with a replication of the IA. From there, the Scopeware server will share the DOM (with constant pointer to the IA or replicated IA) with other connected system servers and clients in order to make the IA available to all clients connected to the network. Thus, the system servers and network agents (Doc Feeders) act as document proxies for both storage and retrieval of IAs.

In addition, the system servers within the network need not be physically close in proximity. For example, a client in a truly global organization with locations and system servers on several continents can query and retrieve sales results across all system servers and clients through a federated search. In essence, the disclosed system creates a virtual store from all documents accessible to any system server or client either centralized or decentralized.

The physical information storage of the disclosed system follows three models: duplication, replication, and document reference. The duplication model physically stores a duplicate IA on the parent Scopeware server that was created by the client. Other clients polling the parent Scopeware server have full access to the IA, depending upon permissions, whether or not the original document is available from its native storage location (i.e. client PC is turned off). The replication model replicates the IA from the parent Scopeware server to the peer Scopeware servers within a federated network. All clients within the federated network have full access to the IA, depending upon permissions, whether or not the original document is available from its native storage location (i.e. client PC is turned off). An example of the replication model is the concept of a redundant array of inexpensive servers. This concept, which is described in detail in the distributed enterprise model, utilizes client machines in place of a singe server. The document reference model "parks" only a DOM of the IA on all Scopeware servers and maintains a constant pointer to the actual physical location of the IA rather than storing a full copy of the IA on the Scopeware server. Other clients will only be able to gain access to the IA when the physical location of the IA is connected to the network (i.e. client PC is turned on).

There are to primary types of streams in accordance with the disclosed system: Bottom-Up and Top-Down. Through the use of both Bottom-Up and Top-Down methodologies, Scopeware creates a living stream for the client with new DOMs appearing automatically as content arrives. The Scopeware distributed enterprise model can make use of both server-based resources and client-based resources where appropriate. Both types of streams can be used simultaneously and interchangeably.

Bottom-Up streams are comprised of information collaboration formed by ad-hoc groups of Scopeware clients. A bottom-up stream is composed of information created by the clients of a transitory group. Information shared and created by this group is be replicated via point-to-point connections (i.e. from client PC to client PC). In this way, bottom-up groups can form and disperse frequently, and without notification, while its members will still have access to the shared information. FIG. **9** illustrates this configuration.

Top-Down streams are more permanent, generally more administrative streams or collections of information, such as company-wide distribution lists, or groups like 'Accounting' and 'Development'. In these groups, information is "parked" to the server from the desktop. The server then sends the information to other known servers. Each client maintains a polling connection to the server to retrieve "parked" documents that have recently arrived from other remote servers or from local clients. FIG. **10** illustrates this configuration.

As earlier described, the user interface within the Scopeware product portfolio has unique characteristics. The DOM provides certain information that allows quick perusal of the information retrieval results via a proprietary "browse card" or "glance view" which is similar to an index card that contains data on the underlying IA. A unique "browse card" or "glance view" is created for each IA. The "browse card" or "glance view" includes metadata for the document, which is comprised of a title, identification number unique to Scopeware document referencing, date/time stamp, and owner information. The "browse card" or "glance view" also presents a thumbnail image of the IA and a summary of the IA contents. Finally, the "browse card" or "glance view" contains a list of operations appropriate for the IA's application that include, but are not limited to, copy, forward, reply, view, and properties.

The "browse card" or "glance view" arrives in the stream of those clients that have permission to view the IA. The owner can grant access to other clients or groups by granting read, write, or aware permissions through the properties of the

"browse card" or "glance view." Permission can be granted as granular as an individual-by-individual basis from the DOM, or through predetermined administrative groups via the Scopeware server.

The "browse card" or "glance view" is presented in a time-ordered sequence starting in the present going back into the past. The "browse card" or "glance view" is available in a number of views. The primary view is the stream. Other formats include a grid, Q, list, and thumbnails. The various views address the client's personal preferences for accessing time-ordered content in their most logical way. These views all contain the information presented in a "browse card" or "glance view" but are organized in a different method. Other specialized views include the address book and calendar.

An advantage of the "browse card" or "glance view" approach is the ease of browsing, searching, and retrieving IAs. In the stream view, the "browse card" or "glance view" of each IA are aligned much like cards in a recipe box. For each item, the title and application icon are viewable on the "browse card" or "glance view" in the stream. When the client passes over the "browse card" or "glance view" in the stream with the mouse pointer, the full "browse card" or "glance view" is presented to the client for easy viewing. From the "browse card" or "glance view," the client can perform any of the aforementioned actions available to the IA, subject to permission access.

The disclosed system is suitable for a number of computing models servicing multiple clients including a single departmental server model, an enterprise server model, a distributed enterprise model, and a peer-to-peer model (absent a dedicated Scopeware server or common server). In addition, the software enables wireless computing independent of or in conjunction with any or all of the aforementioned models. Wireless clients include WAP enabled phones, PDAs, Pocket PCs, and other similarly capable devices capable of receiving and transmitting data across a network. All of the Scopeware Implementation Models make use of the components previously discussed, providing consistent interface available across different computing topologies, from monolithic single servers to peer-to-peer collaboration.

Access to the IA contained in the Scopeware repository can be achieved through two methods. The first method of access is through the thin-client method. The thin-client method utilizes a web browser, such as Microsoft's Internet Explorer or Netscape's Navigator, on the client device to gain access to the Scopeware repository residing on the Scopeware server. The second method of access is the desktop-client method. The desktop-client method involves a local installation of Scopeware on the client device. The client device is then capable of performing the storage, retrieval, extraction, and processing of IAs as they are introduced to the Scopeware repository. All the models below can utilize either method of access to the Scopeware repository, however the distributed enterprise and peer-to-peer models are optimized with the desktop-client method.

Single Server Model. A single server model makes content on one Scopeware server available to any client connected to the departmental server. The Scopeware software creates a unique DOM that represents to the user interface the relevant details of the IA physically stored by the server or client. Thus, when a client connected to the network requests access to and retrieval of IAs through scopeware, the client can view all documents contained within the network that satisfy the query parameters and access restrictions regardless of the document's native application. The documents available include those stored locally by the client, those saved to a central storage location, and those stored by peer clients with Doc Feeders connected to the shared server

Enterprise Server Model. In an enterprise server model, where multiple Scopeware servers are installed, federated access to and retrieval of IAs across the network is enabled. In federated information sharing, a client asks one Scopeware server for IAs that may reside on it or one of many connected peer Scopeware servers. In this model, the actual IA may reside on any network-connected client, the Scopeware server, or a centralized data storage location. Transparent to the client, the Scopeware servers shuffle the retrieval request and access restrictions to present a single, coherent stream to the client via the presentation architecture previously discussed (within the original patent document).

Distributed Enterprise Model. A distributed enterprise model utilizes the clients for storage, retrieval, and processing of IAs. Through the use of directory monitoring agents, similar to network agents, the physical location of an IA need not be on the Scopeware server, but rather can reside with any client. The Scopeware servers take on a secondary role as administration servers and content parking lots. This model pushes the processing tasks to the clients while using the servers to shuttle IAs throughout the enterprise. The indexing engine, thumbnailing engine, lightweight storage database will be based at the clients.

Taking Scopeware beyond distributed networking and the federated architecture—into a more distributed approach will be straightforward, given the way that the system has been designed. Key elements of the next stage of deployment are distributed document processing and scalable server arrays.

Distributed document processing consists of two different approaches. First, when information was created physically on a desktop machine, but was part of a larger application and intended for storage on a server (rather than on the desktop), the Desktop facilities could do the document extraction, indexing, thumbnailing, etc., and post the results to the Scopeware Server. Second, a Scopeware Server that was handed a document (perhaps from an OCR process or from a central email application) could hand the document off to an available Scopeware Desktop for the same processing. These strategies relieve the processing load on the Scopeware Server and leave it free to focus on handling searches and stream integration, allowing a given Scopeware Server to handle a much larger user load.

When an organization needs to support central processing of large document bases—and needs the reliability, accessibility and security of a centralized architecture—Scopeware Servers will support deployment in a novel architecture we have named RAIS—a redundant array of inexpensive servers.

In this architecture, imagine a square array of desktop machines—call each one a "sub-server." The array as a whole comprises the Scopeware Server. (This does not require wiring together an actual array or cluster; any interconnect such as a Ethernet sub-net or even HTTP over a broader network will work.) In these arrays, columns of servers provide redundancy for storage, while rows (within columns) provide redundant points of distribution.

To post document D, one copy of D is sent to a sub-server in each column of the array. To replicate everything five times such that losing any data requires the loss of five sub-servers, five columns are used. The number of columns in the array is managed to support exactly the degree of replication (and redundancy) desired. The write processes can be managed in a number of ways to ensure that the different rows in the columns are balanced.

To send a polling message or search request ("give me all the latest stuff"), a request is sent to each sub-server in one

column (note that the means to do this transparently to the user is an extension of the federated search technology). Each column of sub-servers absorbs one copy of every posting (because any write has gone into at least one row of the column); therefore, all the sub-servers in any one column collectively have copies of everything. Just a "replication factor," is chosen for data redundancy, a "distribution factor" is chosen for responsiveness and for data management, representing the number of rows in any column. To get ten small responses to a search request instead of one big response, or to distribute the total data-storage burden over ten machines instead of one, the array is implemented with ten sub-servers in every column.

The entire "Server" can be run with only one row (resulting in replication, but no distribution) or with only one column (resulting in distribution but no replication). In the limit, row size=column size=1, and the effect is to have a single conventional server.

This approach to distributed processing, scalability and reliability for large applications allows arbitrary sets of "smaller" computers (single/dual processor, inexpensive memory and disk storage) to be used in place of very large, expensive machines. This allows the application platform to be designed to the reliability and access requirements of the particular application, and then scaled incrementally (by adding more small machines into the array) as the actual application grows in terms of users served or information managed.

Distributed document processing and server arrays will give Scopeware almost infinite scalability while maintaining compatibility with early solutions or architectures. In addition to adding greater reliability, this architecture will support very large information processing applications. This will allow enterprise-scale, top-down applications—inbound support/sales email handling, customer service or even IRS-scale tax document processing.

Distributed document processing (with Scopeware Desktop) could be combined with either a "conventional" (1 processor array) Scopeware Server or with a more powerful array. This will allow organizations to create departmental or workgroup level solutions that can grow into enterprise applications if necessary.

At the same time, the system will allow users themselves to create self-organizing applications based on their specific and current needs. Ad hoc teams can create collaborative spaces that cross organizational boundaries if necessary. These applications can leverage either Scopeware Desktops or departmental-level Scopeware Servers.

Because the system has the architecture and capacity to support any level of centralization or decentralization concurrently, applications and their platforms can be engineered centrally or grown organically, and they can be tailored to the needs of their users and the organization on an ongoing basis.

Peer-to-Peer Model. The peer-to-peer (P2P) model allows multiple clients to share IA directly without the use of a dedicated Scopeware server. The P2P model allows for pure ad hoc collaboration among Scopeware clients. For example, a client can share IA via the Internet with identified Scopeware clients that have permission to access IA from the client, and vice versa. This is similar to the distributed enterprise environment except the dedicated Scopeware server has been removed as a storage, retrieval, and connection mechanism. Instead, Scopeware clients will connect point-to-point with other Scopeware clients through a general network connection such as the Internet.

Using P2P, a client can create a virtual shared stream that looks as though it is stored on a server but is in fact stored only by many clients. Historically, all clients would need access to a shared file folder on a common server in order to share information. With Scopeware, clients can share information that is located on each other's device and are not restricted to a common server or single physical storage location. To illustrate, five clients of Scopeware want to create a shared virtual stream to support a project. They call their group "Team One." Then, when any member of "Team One" posts a document to his or her stream, and marks it "readable by Team One," the system automatically sends a copy to every Scopeware client on the "Team One" list. Each Scopeware client receiving this document pops it into its client's local stream. Thus information created by a client who is a member of "Team One" (and flagged for Team One by the owner) winds up in the local stream of every member of Team One, whether the post is a document, an event (team meeting), task, or contact. It's as if he had sent his posting to a "client" server, and then everyone had polled the server, but in fact there's no server.

The invention claimed is:

1. A method of operating a computer system comprising:

providing the computer system with documents from diverse applications in respective formats unique to the respective applications;

causing the computer system to automatically, without user interaction and without requiring a user to designate directory structures or other pre-imposed document categorizations structures, store the provided documents as a time-ordered main stream of documents associated with respective automatically generated time indicators;

said time-ordered main stream being unbounded to thereby accommodate documents associated with time indicators related to past, present and future times;

said time-ordered main stream requiring no fixed beginning or end and being maintained and being selectively retrievable and searchable by the computer system;

said computer system maintaining the main stream live and responsive to subsequent events by automatically incorporating therein new documents as provided to the computer system while maintaining the thus expanded main stream time-ordered;

providing selected search criteria;

causing said computer system to search said time-ordered main stream according to said search criteria and use search results to create a time-ordered substream of documents from the main time-ordered stream;

further causing said computer system to maintain said substream live and responsive to subsequent events by automatically incorporating therein new document provided to the computer system that meet the search criteria while maintaining the thus expanded substream time-ordered;

displaying at least selected portion of the live main stream or substream on computer display means as a display reflecting the time-ordered nature thereof;

automatically showing on the display means a display of a glance view of a displayed document in response to touching with a cursor a screen area associated with the document;

said glance view being an abbreviated version of the document and indicative of content thereof; and

said showing of the glance view occurring essentially instantaneously in response to said touching with the cursor of the screen area associated with the document.

2. A method of operating a computer system as in claim **1** in which said displaying comprises displaying respective representations showing content of at least some of the documents in the substream.

3. A method of operating a computer system as in claim 2 in which said displaying comprises displaying, as said representations, abbreviated versions of the respective documents.

4. A method of operating a computer system as in claim 3 in which said displaying comprises displaying said representations as a receding, foreshortened stack of representations of documents.

5. A method of operating a computer system as in claim 1 in which said displaying comprises displaying at least some of the documents in the substream in the form of a card gallery of document representations showing document content.

6. A method of operating a computer system as in claim 1 in which the displayed glance view of a displayed document comprises buttons for operations unique to the type of the displayed document, said buttons when touched by a cursor initiating computer system operations related to said document.

7. A method of operating a computer system as in claim 1 including utilizing subsystems from another operating system in carrying out at least some of the steps of the method.

8. A method of operating a computer system as in claim 7 in which the another operating system comprises one of a Windows operating system and an Apple operating system.

9. A method of operating a computer system as in claim 1 wherein said documents reside as respective time ordered streams in at least one server and in a number of personal computers selectively communicating with each other and with the server, and said method comprises using one of the personal computers to search a number of said respective streams in order to create a time-ordered substream.

10. A computer readable medium tangibly embodying a program of instructions executable by a computer system to cause the computer system to perform a method comprising:

providing the computer system with documents from diverse applications in respective formats unique to the respective applications;

causing the computer system to automatically, without user interaction and without requiring a user to designate directory structures or other pre-imposed document categorizations structures, store the provided documents as a time-ordered main stream of documents associated with respective automatically generated time indicators;

said time-ordered main stream being unbounded to thereby accommodate documents associated with time indicators related to past, present and future times;

said time-ordered main stream requiring no fixed beginning or end and being maintained and being selectively retrievable and searchable by the computer system;

said computer system maintaining the main stream live and responsive to subsequent events by automatically incorporating therein new documents as provided to the computer system while maintaining the thus expanded main stream time-ordered;

providing selected search criteria;

causing said computer system to search said time-ordered main stream according to said search criteria and use search results to create a time-ordered substream of documents from the main time-ordered stream;

further causing said computer system to maintain said substream live and responsive to subsequent events by automatically incorporating therein new document provided to the computer system that meet the search criteria while maintaining the thus expanded substream time-ordered;

displaying at least selected portion of the live main stream or substream on computer display means as a display reflecting the time-ordered nature thereof;

automatically showing on the display means a display of a glance view of a displayed document in response to touching with a cursor a screen area associated with the document;

said glance view being an abbreviated version of the document and indicative of content thereof; and

said showing of the glance view occurring essentially instantaneously in response to said touching with the cursor of the screen area associated with the document.

11. A computer readable medium tangibly embodying a program of instructions executable by a computer system to cause the computer system to perform a method as in claim 10 in which said displaying comprises displaying respective representations showing content of at least some of the documents in the substream.

12. A computer readable medium tangibly embodying a program of instructions executable by a computer system to cause the computer system to perform a method as in claim 11 in which said displaying comprises displaying, as said representations, abbreviated versions of the respective documents.

13. A computer readable medium tangibly embodying a program of instructions executable by a computer system to cause the computer system to perform a method as in claim 12 in which said displaying comprises displaying said representations as a receding, foreshortened stack of representations of documents.

14. A computer readable medium tangibly embodying a program of instructions executable by a computer system to cause the computer system to perform a method as in claim 10 in which said displaying comprises displaying at least some of the documents in the substream in the form of a card gallery of document representations showing document content.

15. A computer readable medium tangibly embodying a program of instructions executable by a computer system to cause the computer system to perform a method as in claim 10 in which the displayed glance view of a displayed document comprises buttons for operations unique to the type of the displayed document, said buttons when touched by a cursor initiating computer system operations related to said document.

16. A computer readable medium tangibly embodying a program of instructions executable by a computer system to cause the computer system to perform a method as in claim 10 including utilizing subsystems from another operating system in carrying out at least some of the steps of the method.

17. A method computer readable medium tangibly embodying a program of instructions executable by a computer system to cause the computer system to perform a method as in claim 16 in which the another operating system comprises one of a Windows operating system and an Apple operating system.

18. A computer readable medium tangibly embodying a program of instructions executable by a computer system to perform a method as in claim 17 wherein said documents reside as respective time ordered streams in at least one server and a number of personal computers selectively communicating with each other and with the server, and said method comprises using one of the personal computers to search a number of said respective streams in order to create a time-ordered substream.

19. A computer system comprising:

inputs providing the computer system with documents from diverse applications in respective formats unique to the respective applications;

a processing system causing the computer system to automatically, without user interaction and without requiring a user to designate directory structures or other pre-

imposed document categorizations structures, store the provided documents as a time-ordered main stream of documents associated with respective automatically generated time indicators;

said time-ordered main stream being unbounded to thereby accommodate documents associated with time indicators related to past, present and future times;

said time-ordered main stream requiring no fixed beginning or end and being maintained and being selectively retrievable and searchable by the computer system;

said computer system maintaining the main stream live and responsive to subsequent events by automatically incorporating therein new documents as provided to the computer system while maintaining the thus expanded main stream time-ordered;

a source providing selected search criteria;

said processing system causing said computer system to search said time-ordered main stream according to said search criteria and use search results to create a time-ordered substream of documents from the main time-ordered stream;

said processing system further causing said computer system to maintain said substream live and responsive to subsequent events by automatically incorporating therein new document provided to the computer system that meet the search criteria while maintaining the thus expanded substream time-ordered;

said computer system:

displaying at least selected portion of the live main stream or substream on computer display means as a display reflecting the time-ordered nature thereof;

automatically showing on the display means a display of a glance view of a displayed document in response to touching with a cursor a screen area associated with the document;

said glance view being an abbreviated version of the document and indicative of content thereof; and

said showing of the glance view occurring essentially instantaneously in response to said touching with the cursor of the screen area associated with the document.

20. A computer system as in claim **19** in which said displaying comprises displaying respective representations showing content of at least some of the documents in the substream.

21. A computer system as in claim **20** in which said displaying comprises displaying, as said representations, abbreviated versions of the respective documents.

22. A computer system as in claim **21** in which said displaying comprises displaying said representations as a receding, foreshortened stack of representations of documents.

23. A computer system as in claim **19** in which said displaying comprises displaying at least some of the documents in the substream in the form of a card gallery of document representations showing document content.

24. A computer system as in claim **19** in which the displayed glance view of a displayed document comprises buttons for operations unique to the type of the displayed document, said buttons when touched by a cursor initiating computer system operations related to said document.

25. A computer system as in claim **19** in which said processing system utilizes subsystems from another operating system.

26. A computer system as in claim **25** in which the another operating system comprises one of a Windows operating system and an Apple operating system.

27. A computer as in claim **26** in which said documents reside as respective time ordered streams in at least one server and a number of personal computers selectively communicating with each other and with the server, and said processing system causes one of the personal computers to search a number of said respective streams in order to create a time-ordered substream.

* * * * *

US008255439B2

(12) **United States Patent**
Prager et al.

(10) **Patent No.: US 8,255,439 B2**
(45) **Date of Patent: Aug. 28, 2012**

(54) **DESK-TOP, STREAM-BASED, INFORMATION MANAGEMENT SYSTEM**

(75) Inventors: **Randy Prager**, New York, NY (US); **Peter Sparago**, Cheshire, CT (US); **Stephen Marcaurele**, Meriden, CT (US); **David Gelernter**, Woodbridge, CT (US); **Eric Freeman**, Bainbridge Island, WA (US)

(73) Assignee: **Mirror Worlds, LLC**, Tyler, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 67 days.

(21) Appl. No.: **12/966,809**

(22) Filed: **Dec. 13, 2010**

(65) **Prior Publication Data**

US 2011/0145246 A1 Jun. 16, 2011

Related U.S. Application Data

(63) Continuation of application No. 11/528,070, filed on Sep. 26, 2006, now Pat. No. 7,865,538, which is a continuation of application No. 09/892,258, filed on Jun. 26, 2001, now Pat. No. 6,768,999, which is a continuation-in-part of application No. 09/398,611, filed on Sep. 17, 1999, now Pat. No. 6,638,313, which is a continuation of application No. 08/673,255, filed on Jun. 28, 1996, now Pat. No. 6,006,227.

(51) **Int. Cl.**
G06F 17/30 (2006.01)
G06F 12/00 (2006.01)

(52) **U.S. Cl.** **707/828**; 707/822

(58) **Field of Classification Search** ..... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,586,035	A *	4/1986	Baker et al.	345/157
4,831,758	A	5/1989	Williams et al.	
5,031,346	A	7/1991	Herring et al.	
5,060,135	A	10/1991	Levine et al.	
5,063,495	A	11/1991	MacPhail	
5,140,676	A	8/1992	Langelaan	
5,150,410	A	9/1992	Bertrand	
5,159,669	A	10/1992	Trigg et al.	
5,241,671	A	8/1993	Reed et al.	
5,247,437	A	9/1993	Vale et al.	
5,283,864	A	2/1994	Knowlton	

(Continued)

FOREIGN PATENT DOCUMENTS

JP 6-180661 6/1994

(Continued)

OTHER PUBLICATIONS

Cowart, Mastering Windows 3.1, 1992, chapter 12, pp. 396-417, Jan. 1, 1992.

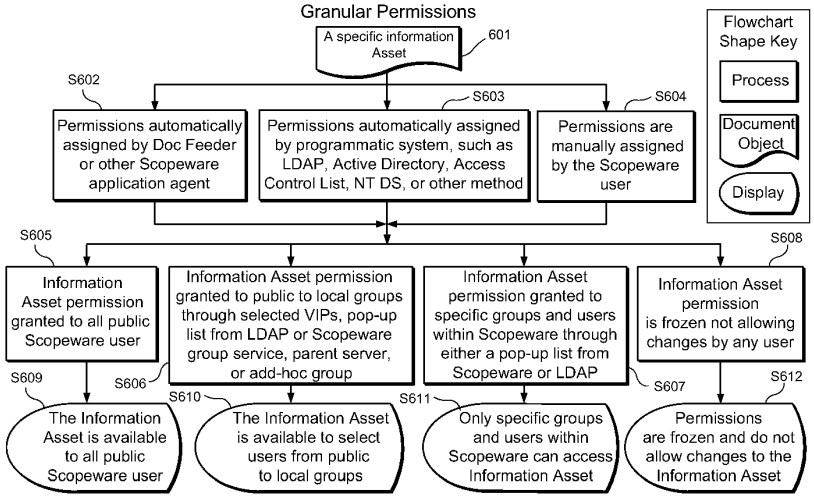
(Continued)

Primary Examiner — Kuen Lu
(74) *Attorney, Agent, or Firm* — Cooper & Dunham, LLP

(57) **ABSTRACT**

A steam-based document storage and retrieval system accepts documents that are in diverse formats and come from diverse application, automatically creates document model objects describing these documents in a consistent format and associating time stamps with the documents to automatically create a main stream in chronological order. The stream, or sub-streams meeting selected search criteria, are displayed in a variety of forms, including a receding, partly overlapping stack with aids that facilitate user interaction.

33 Claims, 11 Drawing Sheets



U.S. PATENT DOCUMENTS

5,287,448	A		2/1994	Nicol et al.	
5,297,032	A		3/1994	Trojan et al.	
5,299,122	A	*	3/1994	Wang et al.	707/1
5,303,361	A		4/1994	Colwell et al.	
5,355,602	A		10/1994	Menke et al.	
5,402,526	A		3/1995	Bauman et al.	
5,430,710	A		7/1995	Mueller et al.	
5,448,729	A		9/1995	Murdock	
5,479,602	A		12/1995	Baecker et al.	
5,499,330	A		3/1996	Lucas et al.	
5,504,852	A		4/1996	Thompson-Rohrlich	
5,528,739	A		6/1996	Lucas et al.	
5,530,859	A		6/1996	Tobias, II et al.	
5,535,063	A		7/1996	Lamming	
5,542,086	A	*	7/1996	Andrew et al.	707/6
5,543,088	A		8/1996	Halbirt	
5,586,237	A	*	12/1996	Baecker et al.	345/670
5,589,892	A		12/1996	Knee et al.	
5,600,833	A		2/1997	Senn et al.	
5,603,025	A	*	2/1997	Tabb et al.	707/2
5,613,134	A		3/1997	Lucus et al.	
5,616,876	A		4/1997	Cluts	
5,621,874	A		4/1997	Lucas et al.	
5,621,906	A		4/1997	O'Neill et al.	
5,625,818	A		4/1997	Zarmer et al.	
5,649,182	A		7/1997	Reitz	
5,649,188	A		7/1997	Nomura et al.	
5,701,582	A		12/1997	DeBey	
5,724,567	A		3/1998	Rose et al.	
5,729,730	A		3/1998	Wlaschin et al.	
5,737,737	A	*	4/1998	Hikida et al.	707/104.1
5,758,324	A		5/1998	Hartman et al.	
D395,297	S		6/1998	Cheng et al.	
5,764,972	A		6/1998	Crouse et al.	
5,778,364	A		7/1998	Nelson	
5,784,620	A	*	7/1998	Isham	719/316
D398,299	S		9/1998	Ballay et al.	
5,835,129	A		11/1998	Kumar	
5,835,789	A	*	11/1998	Ueda et al.	710/33
5,890,177	A		3/1999	Moody et al.	
5,905,992	A		5/1999	Lucas et al.	
5,912,668	A		6/1999	Sciammarella et al.	
5,937,417	A	*	8/1999	Nielsen	715/207
6,006,227	A	*	12/1999	Freeman et al.	707/7
6,012,072	A		1/2000	Lucas et al.	
6,012,074	A		1/2000	Lucas et al.	
6,151,610	A		11/2000	Senn et al.	
6,178,409	B1	*	1/2001	Weber et al.	705/79
6,199,082	B1	*	3/2001	Ferrel et al.	715/205
6,202,058	B1		3/2001	Rose et al.	
6,243,724	B1	*	6/2001	Mander et al.	715/273
6,262,732	B1		7/2001	Coleman et al.	
6,396,513	B1		5/2002	Helfman et al.	
6,401,097	B1		6/2002	McCotter et al.	
6,457,017	B2		9/2002	Watkins et al.	
6,466,237	B1		10/2002	Miyao et al.	
6,496,857	B1		12/2002	Dustin et al.	
6,523,048	B2		2/2003	DeStefano	
6,638,313	B1		10/2003	Freeman et al.	
6,639,876	B2		10/2003	Kalis et al.	
6,714,489	B2		3/2004	Kalis et al.	
6,725,427	B2		4/2004	Freeman et al.	
6,768,999	B2		7/2004	Prager et al.	

FOREIGN PATENT DOCUMENTS

WO	03/001345	A1	1/2003

OTHER PUBLICATIONS

Freeman, Lifestreams Project Home Page, 3 pages, 1994-1996, Jan. 1, 1994.

Freeman, Lifestreams for the Newton, Steve Mann's Developers Corner, Oct. 31, 1995, 5 pages.

Freeman et al., Lifestreams: Organizing your Electronic Life, AAAI Fall Symposium, AI Applications in Knowledge Navigation and Retrieval, Nov. 30, 1995, Cambridge, MA, 6 pages.

Landsdale, M., The psychology of Personal Information Management, Applied Ergonomics, (Mar. 1988): pp. 55-66.

Malone, Thomas W., How Do People Organize Their Desks? Implications For The Design Of Office Information Systems, ACM Transaction On Office Systems, vol. 1, No. 1 (Jan. 1983): pp. 99-112.

Nelson, Theodor H., The Right Way to Think About Software Design, In The Art of Human-Computer Interface Design, Brenda Laurel, (Ed.), (1990): pp. 235-243.

Steinberg, Lifestreams, WIRED 5.02, Feb. 28, 1997, 11 pages.

Getting Results with Microsoft Outlook, copyright Microsoft Corporation 1995-1996, pp. 28-29, Jan. 1, 1995.

User Guide; Mirror Worlds Technologies, Inc. (New Haven, Connecticut).

Gobel, David P., Using Lotus Magellan, Que Corporation (1989).

Retrospect User's Guide, version 3, first edition, Dantz Development Corporation (1989-1995).

Rose, Daniel E., et al., "Content Awareness in a File System Interface: Implementing the 'Pile' Metaphor for Organizing Information", ACM-SIGIR '93 (1993).

Mander, Richard, et al., "A 'Pile' Metaphor for Supporting Casual Organization of Information", ACM-CHI '92 (May 3-7, 1992).

HyperCard Basics and HyperCard Stack Design Guidelines, Apple Computer, Inc. (1990).

Carriero et al. "The 'Lifestreams' Approach to Reorganizing the Information World", YALEU/DCS/TR-1070 (1995).

Gifford, David K., et al., "Semantic File Systems", ACM-SFS '91 (1991).

"On Location 2,0.01" by ON Technology (1990-1991).

Bolour, A., et al., "The Role of Time in Information Processing: A Survey", ACM-SIGART Bulletin, Issue 80 (Apr. 1982).

Rahm, Erhard, "Recovery Concepts for Data Sharing Systems" IEEE (1991).

Inside Macintosh, Addison-Wesley Publishing Company (Aug. 1992).

O'Toole, James W., et al., "Names should mean What, not Where" (1992).

Staples, Loretta, "Representation in Virtual Space: Visual Convention in the Graphical User Interface", ACM-Interchi '93 (Apr. 24-29, 1993).

Goldberg, David., et al., "Using collaborative filtering to weave an information Tapestry", Assocation for Computing Machinery, Inc. (1992).

Gelernter, David, "The Cyber-Road Not Taken", The Washington Post (Apr. 3, 1994).

W3C: A Little History of the World Wide Web, http://www.w3.org/History.html, (Accessed Jul. 15, 2010).

Gelernter, David, "The Cyber-Road Not Taken", The Washington Post, Apr. 3, 1994.

Bolt, Richard A., "Spatial Data-Management", Massachusetts Institute of Technology, 1979.

Lansadale, M.W., et al., "MEMOIRS: A personal Multimedia Information System", The Proceedings of the Fifth Conference of the British Computer Society Human Computer Interaction Specialist Group University of Nottingham, Sep. 1989.

Pemberton, Steven, "Not With a Whimper", SIGCHI, Apr. 1998.

Miller, James R., "The Apple Advanced Technology Group—An Introduction to the Special SIGCHI Bulletin Issue", SIGCHI, Apr. 1998.

Russell, Daniel M., "The ATG Knowledge Management Technologies Laboratory", SIGCHI, Apr. 1998.

Miller, James R., "An Overview of the ATG Intelligent Systems Program", SIGCHI, Apr. 1998.

Miller, James R., et al., "From Documents to Objects: An Overview of LiveDoc", SIGCHI, Apr. 1998.

Bonura, Thomas, et al., "Drop Zones: An Extension to LiveDoc", SIGCHI, Apr. 1998.

Boguraev, Branimir, et al., "An Architecture for Content Analysis of Documents and Its Use in Information and Knowledge Tasks", SIGCHI, Apr. 1998.

Boguraev, Branimir, et al., "Dynamic Document Presentation", SIGCHI, Apr. 1998.

Nardi, Bonnie, et al., "An Online Digital Photography Course for High School Teachers", SIGCHI, Apr. 1998.

Shulka, Shilpa V., "Hit Squads & Bug Meisters", SIGCHI, Apr. 1998.

Rose, Daniel E., "Beyond Search: the Information Access Research Group at Apple", SIGCHI, Apr. 1998.
Russell, Daniel M., "User Experience Research Group", SIGCHI, Apr. 1998.
Walker, William F., Rapid Prototyping of Awareness Services Using a Shared Information Server, SIGCHI, Apr. 1998.
Bellegarda, Jerome R., "Interaction-Driven Speech Input", SIGCHI, Apr. 1998.
Machiraju, N. Rao, "The ATG Learning Communities Laboratory: An Overview", SIGCHI, Apr. 1998.
Bellamy, Rachel, et al., "Learning Conversations", SIGCHI, Apr. 1998.
Houde, Stephanie, et al., "In Search of Design Principles for Tools and Practices to Support Communication Within a Learning Community", SIGCHI, Apr. 1998.
Roddy, Brian J., et al., "Interface Issues in Text Based Chat Rooms", SIGCHI, Apr. 1998.
Spohrer, Jim, "ATG Education Research: The Authoring Tools Thread", SIGCHI, Apr. 1998.
Graves, Mike, et al., "Unfamiliar Ground: Designing Technology to Support Rural Healthcare in India", SIGCHI, Apr. 1998.
Mountford, S. Joy, "A History of the Apple Human Interface Group", SIGCHI, Apr. 1998.
Harris, Jed, et al., "Discourse Architecture", SIGCHI, Apr. 1998.
Singer, Samuel Weller, "Researches into the History of Playing Cards", Printed by T. Bensley and Son, 1816.
"Playing Poker at Egan's" (image), http://www.orww.org/Hamey_Cattle/Student_Reports/Images/11-Poker_1882jpg (ca. 1882).
"Commercial America", Philadelphia Commercial Museum, Jul. 1911.
"Eleventh Annual Report of the Trustees of the Boston State Hospital", Published by Wright & Potter Printing Co., 1920.
"System—The Magazine of Business", A.W. Shaw Company, 1921.
Ibberson, J.R., MD, "The Business Side of General Practice", Canad. M.A.J., vol. 72, Apr. 15, 1955.
Carlbom, Ingrid, et al., "Planar Geometric Projections and Viewing Transformations", Computing Surveys, vol. 10, No. 4, pp. 465-502, Dec. 1978.
Schmandt, Christopher Martin, "Pages Without Paper", Bachelor of Science Thesis at the Massachusetts Institute of Technology, Dec. 1978.
Donelson, William C., "Spatial Management of Information", Architecture Machine Group, Massachusetts Institute of Technology, Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques, 1978.
Bolt, Richard A., "Put-That-There": Voice and Gesture at the Graphics Interface, Architecture Machine Group, Massachusetts Institute of Technology, ACM, 1980.
Halasz, Frank, et al., "Analogy Considered Harmful", Association for Computing Machinery, 1981.
Bolt, Richard A., "Gaze-Orchestrated Dynamic Windows", Architecture Machine Group, Massachusetts Institute of Technology Cambridge Massachusetts, ACM, Computer Graphics, vol. 15, No. 3, Aug. 1981.
Williams, Gregg, "The Lisa Computer System", BYTE Publications, Inc., Feb. 1983.
Williams, Gregg, "The Apple Macintosh Computer", BYTE Publications, Inc., Feb. 1984.
Anderson, John J., "Atari 520ST: A Reborn Atari Once Again Points the Way to the Next Generation", Creative Computing, vol. 11, No. 10, Oct. 1985.
Greenberg, Saul, et al., "Issues and Experiences in the Design of a Window Management System", Canadian Information Processing Society—Edmonton Conference, 1986.
Henderson, D. Austin, Jr., et al., "Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface", ACM Trans. On Graphics, vol. 5, No. 3, pp. 211-243, Jul. 1986.
Robek Mary F., et al., "Information and Records Management", 3rd ed., Glencoe Publishing Co., pp. 160-163, 179-181, 192-193, 224-229, 248-251, (1987).
Lynch, Vincent, "American Jukebox—The Classic Years", Chronicle Books, pp. 36, 74-75, (1990).
Robertson, G. G., et al., "Cone Trees: Animated 3D Visualizations of Hierarchical Information", UIR-R-1991-06, (1991).
Card, S. K., et al., "The Information Visualizer: An Information Workspace", UIR-R1991-01, (1991).
Mackinlay, Jock D., et al., "The Perspective Wall: Detail and Context Smoothly Integrated", UIR-R-1991-03, (1991).
Kay, Alan C., "The Early History of Smalltalk", ACM SIGPLAN Notices, vol. 28, No. 3, Mar. 1993.
Adams, Michael, et al., "Jukeboxes", Schiffer Publishing, Ltd., pp. 85-86, 103-104, 105, (1996).
Card, Stuart K., et al., "The WebBook and the Web Forager: An Information Workspace for the World-Wide Web", CHI '96 Conference on Human Factors in Computing Systems, Apr. 13-18, 1996.
Meyers, Brad A., "A Brief History of Human-Computer Interaction Technology", Interactions, pp. 44-54, 1998.
Bardini, Thierry, "Bootstrapping", Stanford University Press, Chapter 6, pp. 143-181, (2000).
Sutherland, Ivan Edward, "Sketchpad: A Man-Machine Graphical Communication System", PhD Dissertation at the Massachusetts Institute of Technology, Jan. 1963.
Engelbart, Doug, "The Click Heard Round the World", Wired Magazine, 2004.
Reimer, Jeremy, "A History of the GUI", arstechnica.com, May 5, 2005.
"The Library of Congress Classification System", Aug. 2006.
Barnes, Susan B, "Alan Kay: Transforming the Computer into a Communication Medium", IEEE Annals of the History of Computing, 2007.
Calamar, Gary et al., "Record Store Days: From Vinyl to Digital and Back Again", Sterling Publishing Co., 2009.
Masinter, Larry, "Document Management, Digital Libraries and the Web", Jun. 9, 1995.
Herot, Christopher F., "Spatial Management of Data", Computer Corporation of America, ACM Trans. On. Database Systems, vol. 5, No. 4, pp. 493-514, Dec. 1980.
Newman, William M., "A System for Interactive Graphical Programming", Harvard University, ACM Spring Joint Computer Conference, 1968.
Schultz, Jan R., et al., "An Initial Operational Problem Oriented Medical Record System—for Storage, Manipulation and Retrieval of Medical Data", University of Vermont, ACM Spring Joint Computer Conference, 1971.
Apple Macintosh (screenshot image).
Carriero, Nicholas, et al., "Bauhaus Linda", Department of Computer Science, Yale University.
"Small Talk on the Alto" (image).
Donelson, William C., "Spatial Management of Data", Massachusetts Institute of Technology, 1977.
"Lotus Magellan Explorer's Guide", Lotus Development Corp., 1989.
Factor, Michael, et al., "Real-Time Data Fusion in the Intensive Care Unit", Yale University, Nov. 1991.
Gelernter, David, et al., "Graphics Interfaces & Sensor Activator Extensions for the Process Trellis Software", Yale University, Oct. 31, 1995.
"Lifestream" (image), 1995.
Harrison, Beverly L., et al., "Timelines: An Interactive System for the Collection and Visualization of Temporal Data", Proceedings of Graphic Interface '94, pp. 141-148, (1994).
"IBM OS2" (screenshot image).
Gerlenter, David, "Applications and Systems for Large Scale Adaptive Parallelism", Mar. 22, 1997.
Date, C. J., "An Introduction to Database Systems, 3rd ed., The Systems Programming Series, vol. 1", IBM Corporation, Addison-Wesley Publishing Company, 1981.
"The Billboard" (magazine), Jan. 26, 1946.
"Billboard" (magazine), Jul. 4, 1992.
"Library of Congress Classification" (accessed May 6, 2010).
Swineheart, Daniel C., et al., "A Structural View of the Cedar Programming Environment", ACM Trans. On Programming Languages and Systems, vol. 8, No. 4, Oct. 1986.
"Commodore Amiga 1000 Computer" (screenshot image).
"Atari 520ST Computer" (screenshot image).

Lucas, Peter, et al., "Workscape: A Scriptable Document Management Environment", Maya Design Group, Conference Companion CHI '94, Apr. 1994.
Bailay, Joseph M., "Designing Workscape: An Interdisciplinary Experience", Maya Design Group, Conference Companion CHI '94, Apr. 1994.
Perkins, Roderick, et al., "Inventing the Lisa User Interface", Interactions, pp. 40-53, (1997).
Feiner, Steven, et al., "An Experimental System for Creating and Presenting Interactive Graphical Documents", ACM Trans. On Graphics, vol. 1, No. 1, pp. 59-77, Jan. 1982.
Houde, Stephanie, "Iterative Design of an Interface for Easy 3-D Direct Manipulation", CHI '92, May 1992.
"Announcing Retrospect and Retrospect Remote 3.0", Business Wise, Aug. 8, 1995.
"Dantz Delivers Retrospect Archives", MacWEEK, Inc., Jun. 10, 1989.
"Archiving Easy on Retrospect: Uniform Interface to Storage Devices", MacWEEK, Inc., Feb. 21, 1989.
"Mac the Knife, GmbH Rumors", MacWEEK, Inc., Jan. 31, 1989.
"Dantz, ChrisMac drive over to CD-R", MacWEEK, Inc., Dec. 11, 1995.
"System 7.5.2 Fixes", MacWORLD, Oct. 1, 1995.
"Dantz to Waltz in with Friendlier Retrospect", MacWEEK, Inc., Jul. 31, 1995.
"Dantz Eases Network Backup", InfoWorld, Jan. 11, 1993.
"Retrospect User's Guide", 1st Ed., Dantz Development Corporation, 1989.
"Retrospect User's Guide", 2nd Ed., Dantz Development Corporation, 1993.
"Rowe CD100b Jukebox" (image).
Spence, Robert, et al., "Data Base Navigation: An Office Environment for the Professional", Behavior and Information Technology, vol. 1, No. 1, pp. 43-54, (1982).
Johnson, Jeff, et al., "The Xerox Star: A Retrospective", IEEE Computer, Sep. 1989.
Shneiderman, Ben, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations", University of Maryland, Proc. Visual Languages, Sep. 1996.
Rittsteiger, Andreas, "AMI Jukeboxes" (images).
Apple Computer Inc. v. Microsoft Corporation, 35 F.3d 1435 (9th Cir. 1994).
Windows 1.0 (screenshot image).
Windows 2.0 (screenshot image).
Windows 3.0 (screenshot image).
Windows 95 (screenshot image).
Windows NT3.1 (screenshot image).
Plaisant, Catherine et al. "Lifelines: Visualizing Personal Histories", ACM CHI '96 Conference Proc., pp. 221-227, color plate 518, Apr. 1996.
Musthaler, Linda, "A Tall Order for Document Managers", Network World, pp. 3540, Jul. 18, 1994.
Seiden, Peggy, et al., "Information Retrieval Systems for Microcomputers", Library Hi Tech, vol. 3, Issue 1, pp. 41-54, (1985).
"Zylab Includes Auto-Indexing in Zyindex 3.0", InfoWorld, vol. 10, Issue 18, p. 16, May 2, 1988.
Kappes, Sandra, et al., "Document Management for the Knowledge Worker System", US Army Corps of Engineers, USACERL ADP Report 95/38, Nov. 15, 1995.
"Verity K2 Toolkit Search System Administration Guide V2.2", Verity Incorporated, Jul. 20, 2000.
Cleveland, Gary, "Overview of Document Management Technology", International Federation of Library Associations and Institutions Universal Dataflow and Telecommunications Core Programme, Jun. 1995.
Factor, Michael, et al., "The Trellis Architecture for Intelligent Monitors", Department of Computer Science, Yale University, 1990.
Bush, Vannevar, "As We May Think", Life Magazine, Sep. 10, 1945.
Lansadale, M.W., et al., "Memoirs: A personal Multimedia Information System", Elseiver Science Publishers B.V., 1989.
Lansadale, M.W., et al., "Memoirs: A personal Multimedia Information System", 1995.
Bush, Vannevar, "As We May Think", The Atlantic Monthly, Jul. 1945.
"On Location Manual, v. 1.0", On Technology, Inc., 1990.
Coolidge, C. M., "Dogs Playing Poker (His Station and Four Aces)" (image), 1903.
Musthaler, Linda, "DMS's getting mix-and-match wardrobe," Network World, pp. 38-41, Jan. 8, 1996.
Kullberg, Robin Lee, "Dynamic Timelines: Visualizing Historical Information in Three Dimensions", Massachusetts Institute of Technology, 1995.
USDC-EDTX 6:08-cv-88 LED *Mirror Worlds* v. *Apple Inc.*; Apple Inc's Motion for Summary Judgement (partially redacted).
USDC-EDTX 6:08-cv-88 LED *Mirror Worlds* v. *Apple Inc.*; Expert Rebuttal Report of John Levy, Ph.D. (partially redacted).
USDC-EDTX 6:08-cv-88 LED *Mirror Worlds* v. *Apple Inc.*; Expert Report of Steven K. Feiner, Ph.D. (confidential).
Solomon, Gitta, "Piles Metaphor ATG" (video), Apple Computer, Inc., Mar. 1992.
"Spatial Data Management System" (video), MIT, Jan. 1980.
Milash, Brett, et al., "Lifelines: Visualizing Personal Histories" (video), University of Maryland, Apr. 1995.
Lucas, Peter, "CHI '94 Video" (video), Maya Design Group, CHI '94, 1994.
Lucas, Peter, "Workscape Video" (video), http://www.youtube.com/watch?v=H5-T_S50Sr4 (screen capture video), (1993).
"200 Points of Light" (video) http://vvww.youtube.com/watch?v=H9F17JrG-SE (screen capture video), (1990).
Mirror Worlds, LLC v. *Apple, Inc.* Memorandum Opinion and Orderdated Aug. 11, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Mirror Worlds, LLC Notice of Dismissal with Prejudice of Certain Infringement Claims dated Sep. 1, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Mirror Worlds, LLC Defendant and Counterclaim Plaintiff Apple Inc's Notice of Asserted Prior Art for Trial dated Sep. 6, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Memorandum Opinion and Order dated Sep. 16, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Verdict form dated Oct. 1, 2010.
Mirror Worlds, LLC v. *Apple, Inc.* Memorandum Opinion and Order dated Apr. 4, 2011.
Carriero et al. "The 'Lifestreams' Approach to Reorganizing the Information World", YALEU/DCS/TR-1070v2 (Apr. 1995).

* cited by examiner

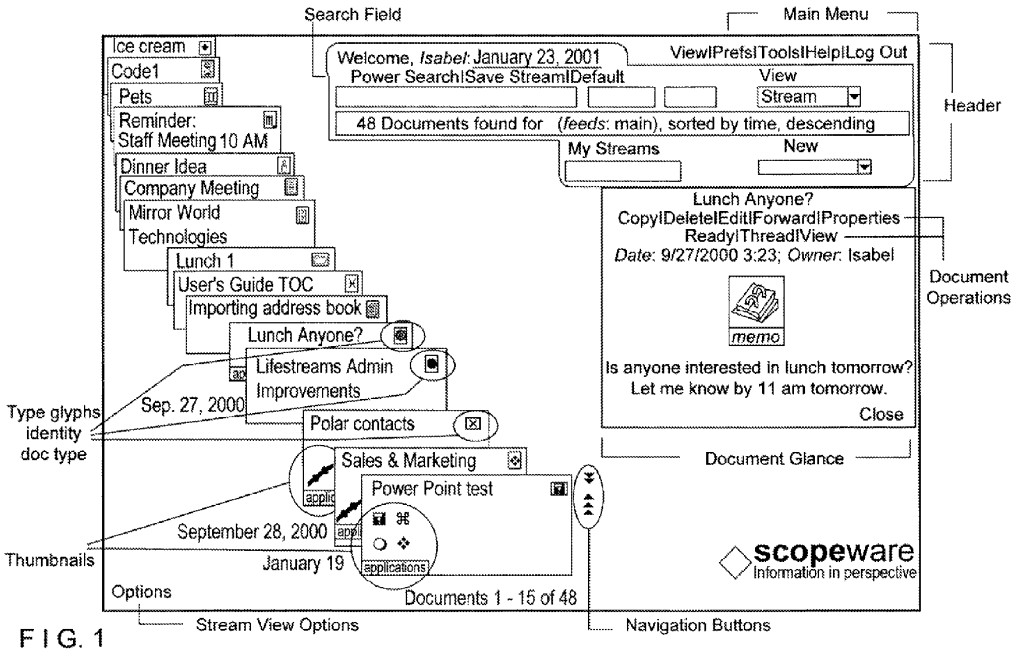
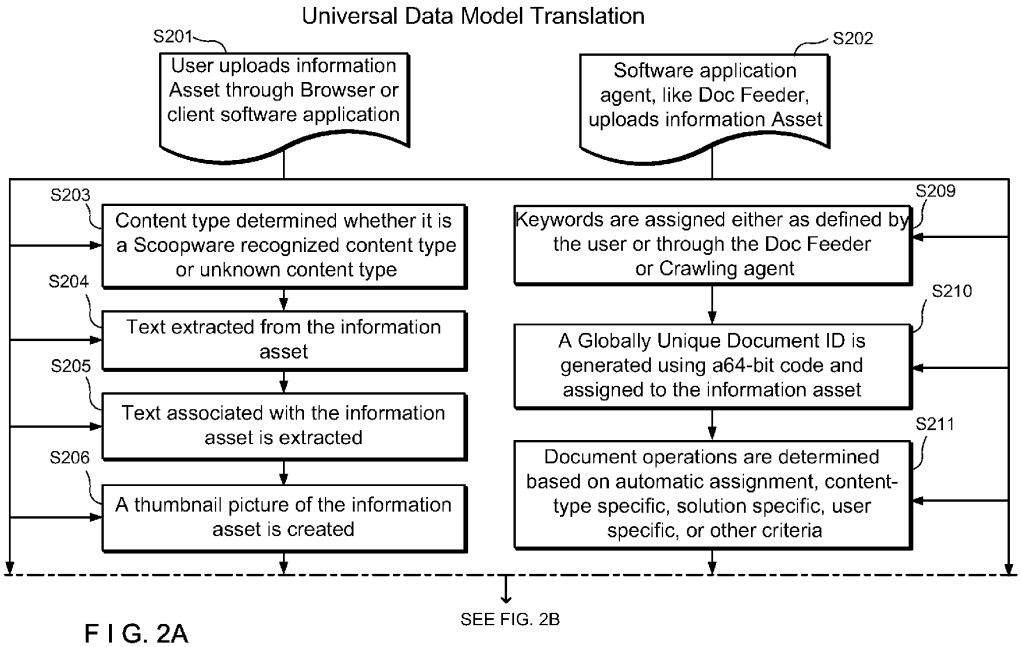
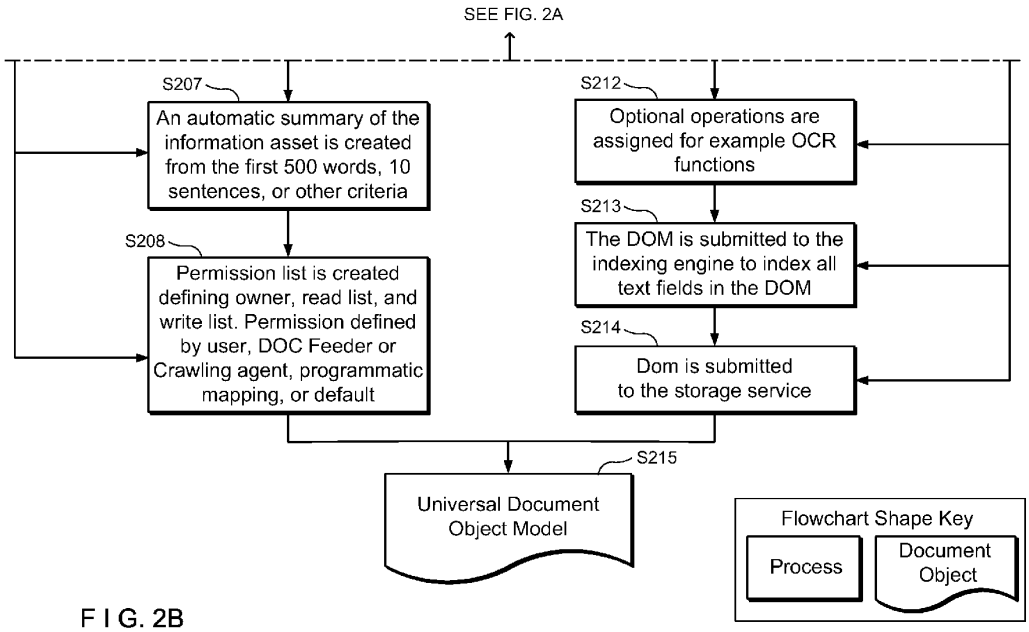

Search Field
Main Menu
Ice cream
Code1
Pets
Reminder:
Staff Meeting 10 AM
Dinner Idea
Company Meeting
Mirror World
Technologies
Lunch 1
User's Guide TOC
Importing address book
Lunch Anyone?
Lifestreams Admin
Improvements
Polar contacts
Sales & Marketing
Power Point test
Welcome, Isabel: January 23, 2001
ViewIPrefsIToolsIHelpILog Out
Power SearchISave StreamIDefault
View
Stream
48 Documents found for (feeds: main), sorted by time, descending
My Streams
New
Header
Lunch Anyone?
CopyIDeleteIEditIForwardIProperties
ReadyIThreadIView
Date: 9/27/2000 3:23; Owner: Isabel
memo
Is anyone interested in lunch tomorrow?
Let me know by 11 am tomorrow.
Close
Document Operations
Document Glance
Type glyphs
identity
doc type
Sep. 27, 2000
September 28, 2000
January 19
Thumbnails
applications
scopeware
Information in perspective
Options
Documents 1 - 15 of 48
Stream View Options
Navigation Buttons

FIG. 1


Universal Data Model Translation
S201
User uploads information Asset through Browser or client software application
S202
Software application agent, like Doc Feeder, uploads information Asset
S203
Content type determined whether it is a Scoopware recognized content type or unknown content type
S204
Text extracted from the information asset
S205
Text associated with the information asset is extracted
S206
A thumbnail picture of the information asset is created
S209
Keywords are assigned either as defined by the user or through the Doc Feeder or Crawling agent
S210
A Globally Unique Document ID is generated using a64-bit code and assigned to the information asset
S211
Document operations are determined based on automatic assignment, content-type specific, solution specific, user specific, or other criteria
SEE FIG. 2B

F I G. 2A


SEE FIG. 2A
S207
An automatic summary of the information asset is created from the first 500 words, 10 sentences, or other criteria
S208
Permission list is created defining owner, read list, and write list. Permission defined by user, DOC Feeder or Crawling agent, programmatic mapping, or default
S212
Optional operations are assigned for example OCR functions
S213
The DOM is submitted to the indexing engine to index all text fields in the DOM
S214
Dom is submitted to the storage service
S215
Universal Document Object Model
Flowchart Shape Key
Process
Document Object

F I G. 2B

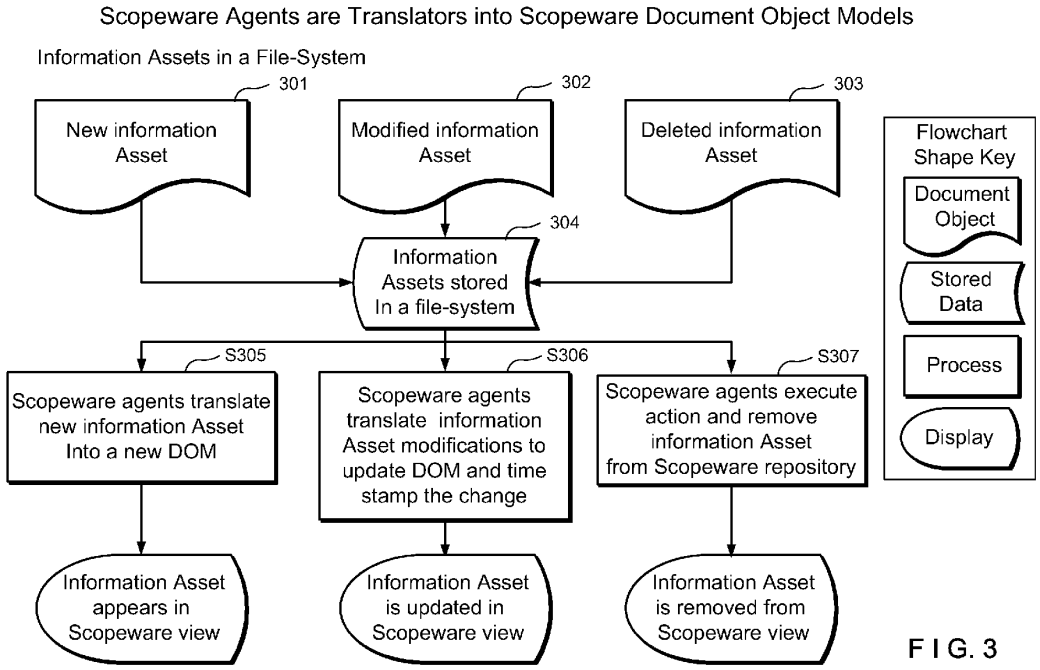
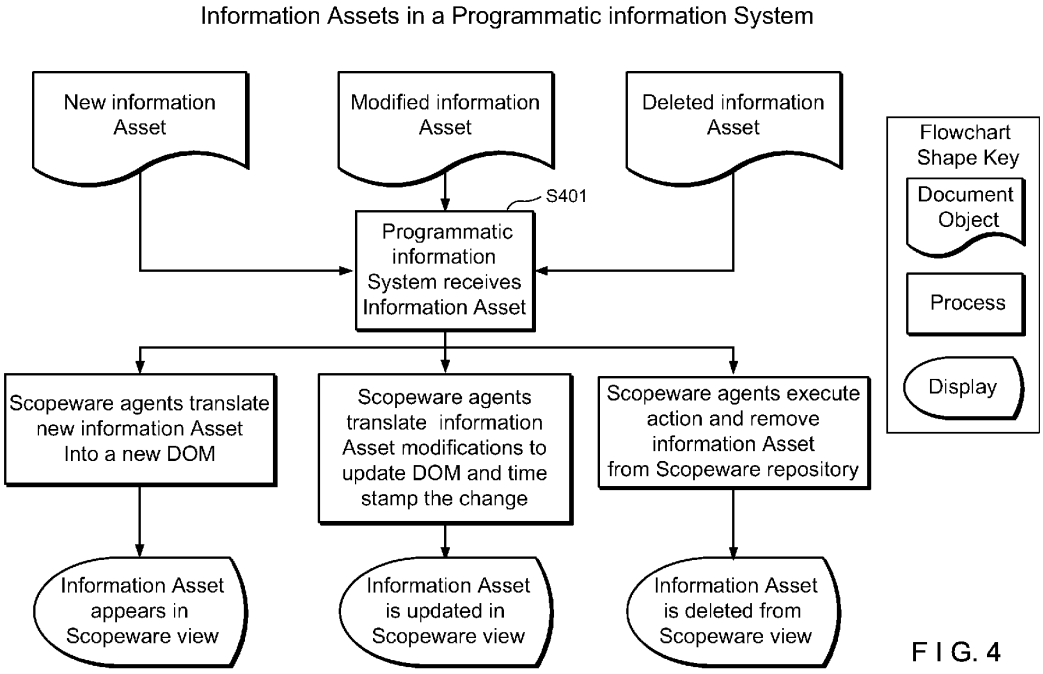
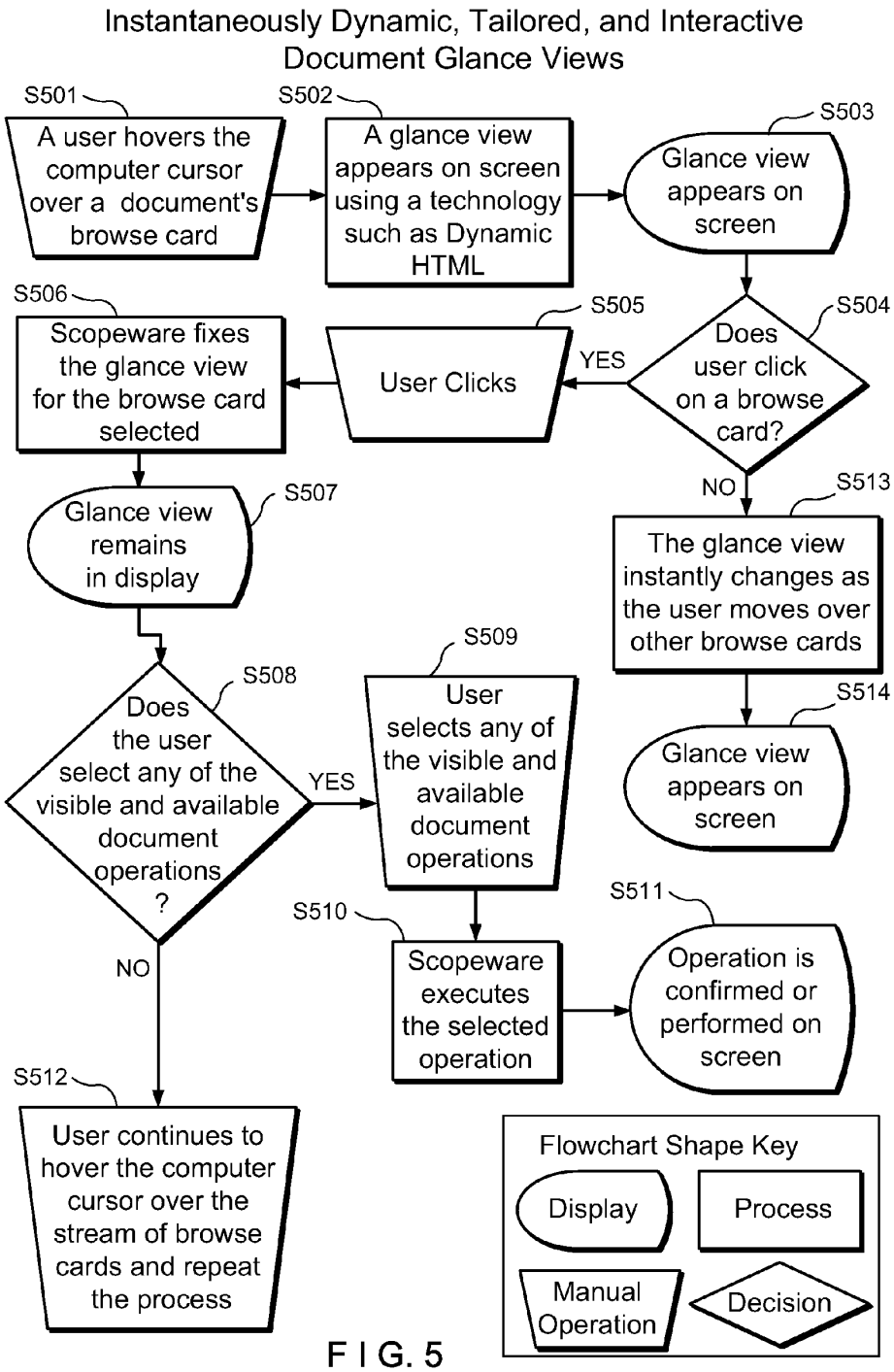

Scopeware Agents are Translators into Scopeware Document Object Models
Information Assets in a File-System
301
New information Asset
302
Modified information Asset
303
Deleted information Asset
304
Information Assets stored In a file-system
S305
Scopeware agents translate new information Asset Into a new DOM
S306
Scopeware agents translate information Asset modifications to update DOM and time stamp the change
S307
Scopeware agents execute action and remove information Asset from Scopeware repository
Information Asset appears in Scopeware view
Information Asset is updated in Scopeware view
Information Asset is removed from Scopeware view
Flowchart Shape Key
Document Object
Stored Data
Process
Display

FIG. 3


Information Assets in a Programmatic information System
New information Asset
Modified information Asset
Deleted information Asset
S401
Programmatic information System receives Information Asset
Scopeware agents translate new information Asset Into a new DOM
Scopeware agents translate information Asset modifications to update DOM and time stamp the change
Scopeware agents execute action and remove information Asset from Scopeware repository
Information Asset appears in Scopeware view
Information Asset is updated in Scopeware view
Information Asset is deleted from Scopeware view
Flowchart Shape Key
Document Object
Process
Display

F I G. 4


Instantaneously Dynamic, Tailored, and Interactive Document Glance Views
S501
A user hovers the computer cursor over a document's browse card
S502
A glance view appears on screen using a technology such as Dynamic HTML
S503
Glance view appears on screen
S504
Does user click on a browse card?
YES
S505
User Clicks
S506
Scopeware fixes the glance view for the browse card selected
NO
S513
The glance view instantly changes as the user moves over other browse cards
S514
Glance view appears on screen
S507
Glance view remains in display
S508
Does the user select any of the visible and available document operations ?
YES
S509
User selects any of the visible and available document operations
S510
Scopeware executes the selected operation
S511
Operation is confirmed or performed on screen
NO
S512
User continues to hover the computer cursor over the stream of browse cards and repeat the process
Flowchart Shape Key
Display
Process
Manual Operation
Decision

F I G. 5

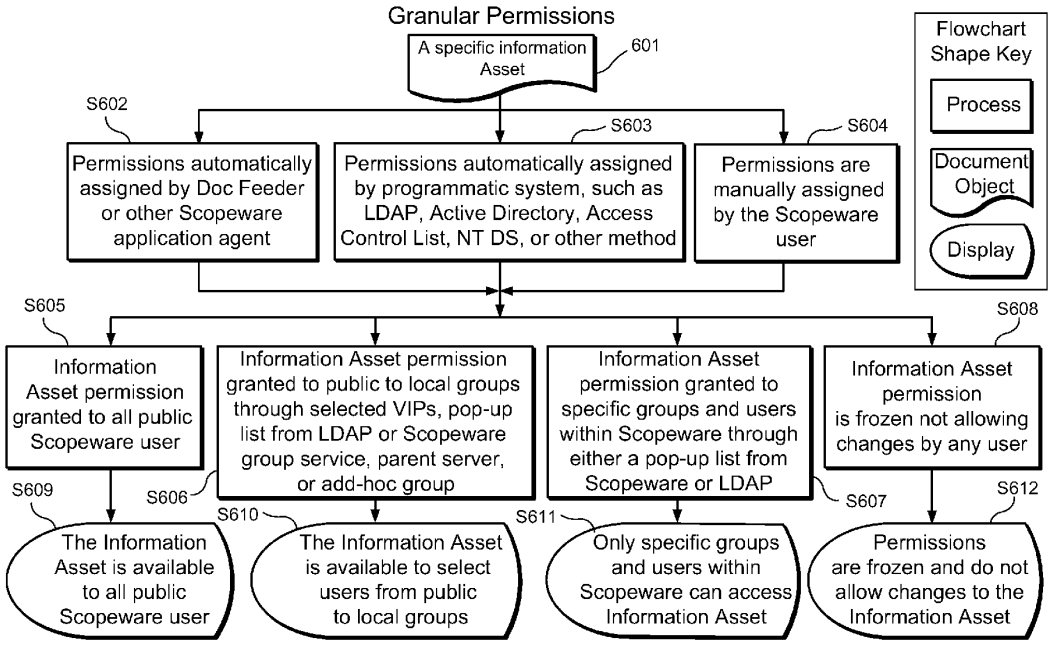

Granular Permissions
A specific information Asset
601
S602
S603
S604
Permissions automatically assigned by Doc Feeder or other Scopeware application agent
Permissions automatically assigned by programmatic system, such as LDAP, Active Directory, Access Control List, NT DS, or other method
Permissions are manually assigned by the Scopeware user
Flowchart Shape Key
Process
Document Object
Display
S605
S608
Information Asset permission granted to all public Scopeware user
Information Asset permission granted to public to local groups through selected VIPs, pop-up list from LDAP or Scopeware group service, parent server, or add-hoc group
Information Asset permission granted to specific groups and users within Scopeware through either a pop-up list from Scopeware or LDAP
Information Asset permission is frozen not allowing changes by any user
S609
S606
S610
S611
S607
S612
The Information Asset is available to all public Scopeware user
The Information Asset is available to select users from public to local groups
Only specific groups and users within Scopeware can access Information Asset
Permissions are frozen and do not allow changes to the Information Asset

FIG. 6

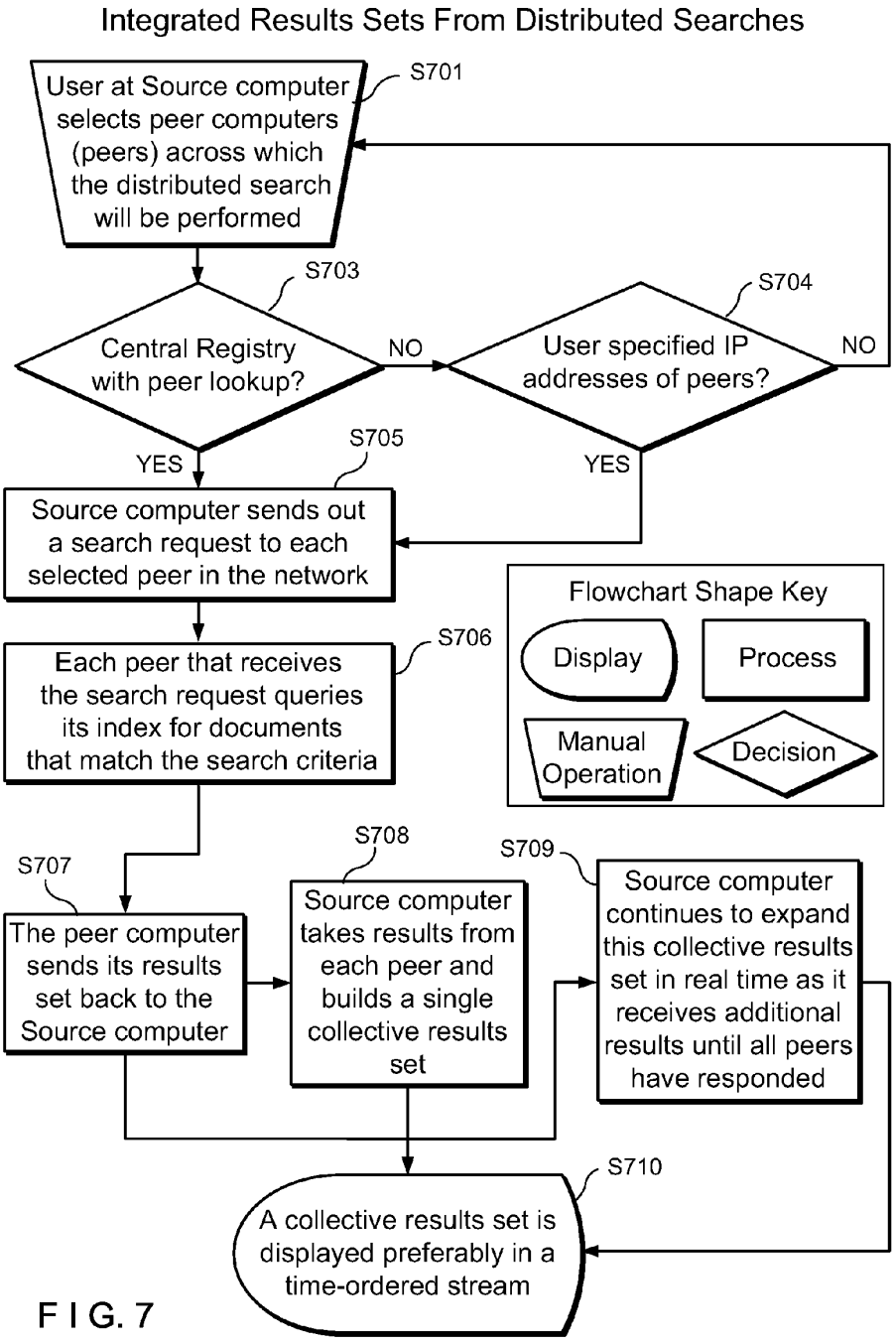

Integrated Results Sets From Distributed Searches
User at Source computer selects peer computers (peers) across which the distributed search will be performed
S701
S703
Central Registry with peer lookup?
NO
S704
User specified IP addresses of peers?
NO
YES
YES
S705
Source computer sends out a search request to each selected peer in the network
Each peer that receives the search request queries its index for documents that match the search criteria
S706
Flowchart Shape Key
Display
Process
Manual Operation
Decision
S707
The peer computer sends its results set back to the Source computer
S708
Source computer takes results from each peer and builds a single collective results set
S709
Source computer continues to expand this collective results set in real time as it receives additional results until all peers have responded
S710
A collective results set is displayed preferably in a time-ordered stream

FIG. 7

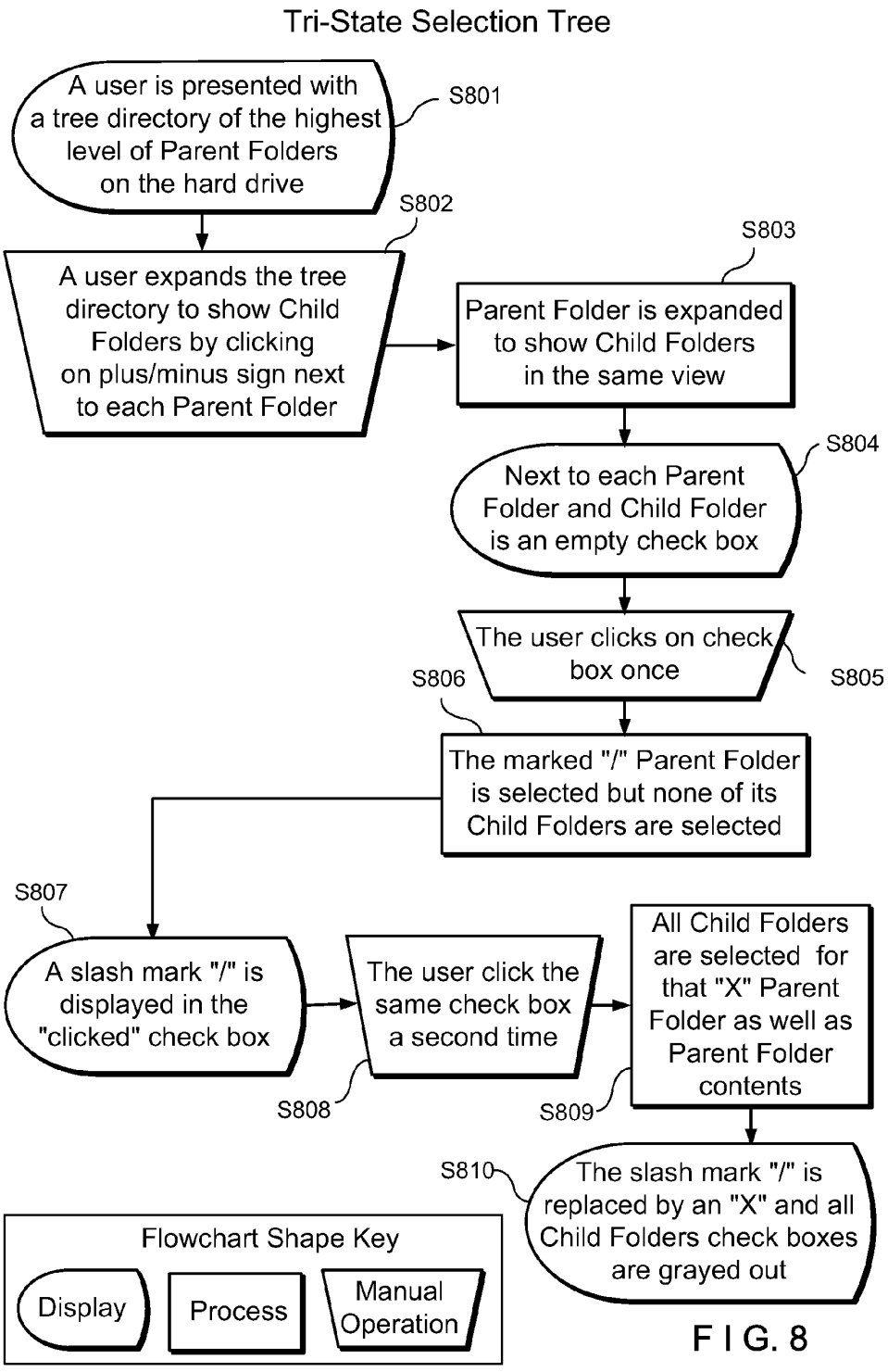

Tri-State Selection Tree
A user is presented with a tree directory of the highest level of Parent Folders on the hard drive
S801
S802
A user expands the tree directory to show Child Folders by clicking on plus/minus sign next to each Parent Folder
S803
Parent Folder is expanded to show Child Folders in the same view
S804
Next to each Parent Folder and Child Folder is an empty check box
The user clicks on check box once
S805
S806
The marked "/" Parent Folder is selected but none of its Child Folders are selected
S807
A slash mark "/" is displayed in the "clicked" check box
The user click the same check box a second time
S808
All Child Folders are selected for that "X" Parent Folder as well as Parent Folder contents
S809
S810
The slash mark "/" is replaced by an "X" and all Child Folders check boxes are grayed out
Flowchart Shape Key
Display
Process
Manual Operation

FIG. 8

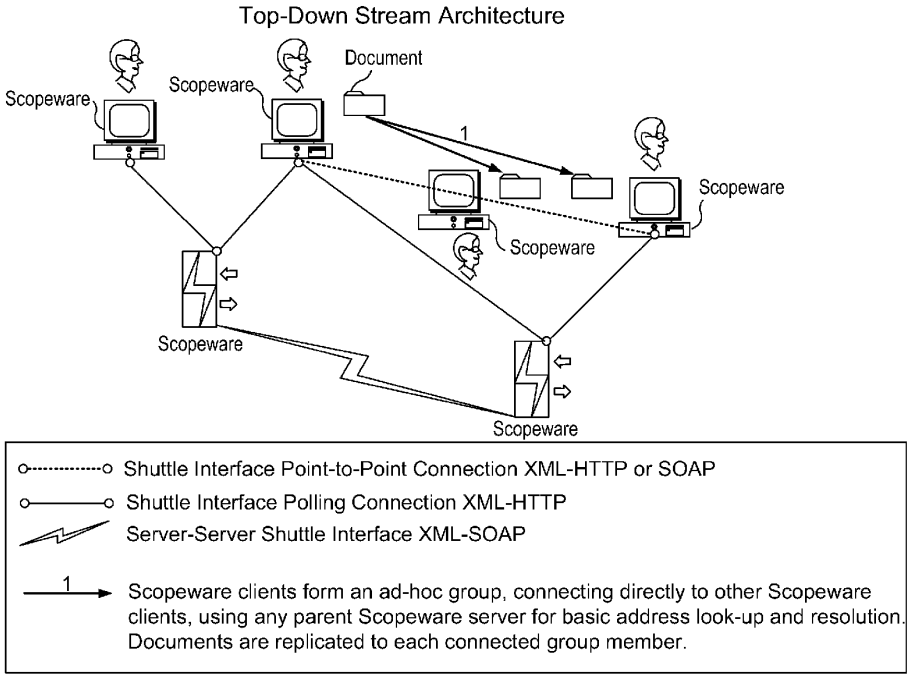

Top-Down Stream Architecture
Scopeware
Scopeware
Document
1
Scopeware
Scopeware
Scopeware
Scopeware
Shuttle Interface Point-to-Point Connection XML-HTTP or SOAP
Shuttle Interface Polling Connection XML-HTTP
Server-Server Shuttle Interface XML-SOAP
1
Scopeware clients form an ad-hoc group, connecting directly to other Scopeware clients, using any parent Scopeware server for basic address look-up and resolution. Documents are replicated to each connected group member.

FIG. 9

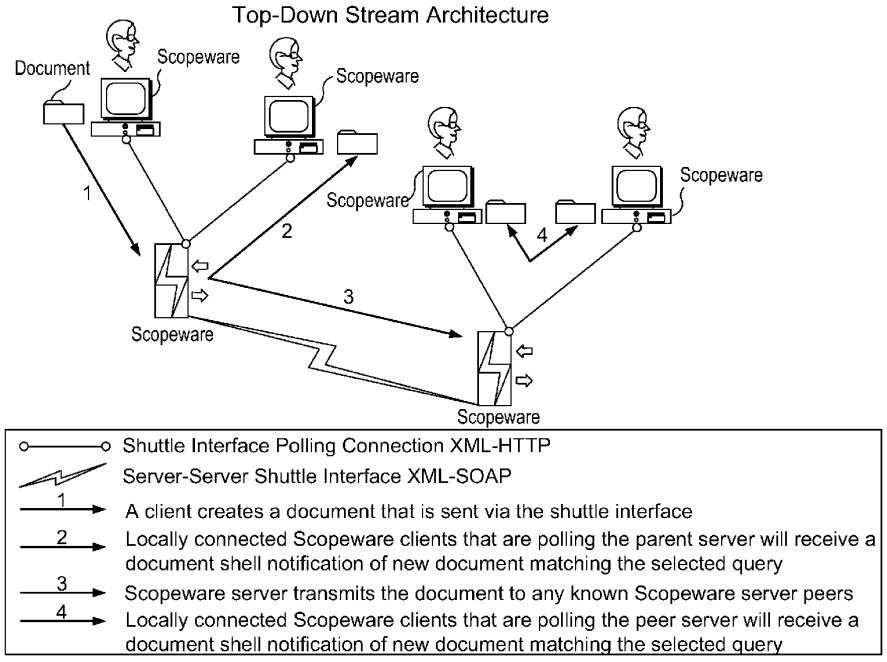

Top-Down Stream Architecture
Document
Scopeware
Scopeware
Scopeware
Scopeware
Scopeware
Scopeware
Scopeware
1
2
3
4
Shuttle Interface Polling Connection XML-HTTP
Server-Server Shuttle Interface XML-SOAP
1 A client creates a document that is sent via the shuttle interface
2 Locally connected Scopeware clients that are polling the parent server will receive a document shell notification of new document matching the selected query
3 Scopeware server transmits the document to any known Scopeware server peers
4 Locally connected Scopeware clients that are polling the peer server will receive a document shell notification of new document matching the selected query

FIG. 10

DESK-TOP, STREAM-BASED, INFORMATION MANAGEMENT SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a Rule 1.53 (b) continuation of U.S. patent application Ser. No. 11/528,070 filed Sep. 26, 2006, which in turn is a Rule 1.53 (b) continuation application of U.S. patent application Ser. No. 09/892,385 filed Jun. 26, 2001, which in turn is a continuation-in-part of U.S. patent application Ser. No. 09/398,611 filed Sep. 17, 1999 and now U.S. Pat. No. 6,638,313, which in turn is a Rule 1.53 continuation of U.S. patent application Ser. No. 08/673,255 filed Jun. 28, 1996 and now U.S. Pat. No. 6,006,227, and hereby incorporates by reference said prior applications in their entireties, as though fully set forth herein.

INCORPORATION BY REFERENCE OF MATERIAL ON COMPACT DISC

This patent specification incorporates by reference the contents of the compact disc attached hereto in duplicate (Copy 1 and Copy 2). Each disc is labeled in accordance with Rule 1.53(e)(6), with the collective names Scopeware 2.0 and Vision 1.0. The date of creation of the files on the disc is Jun. 25, 2001. The computer code on the compact disc was generated from correspondingly named source code. The names of individual files on the disc within these collective names, as well as the size of the individual files, are identified in the list of files attached to the Transmission Letter In Accordance with 37 C.F.R. §1.52(e)(ii). The contents of the compact disc submitted herewith in duplicate and the contents of the list of files attached to said Transmission Letter are hereby incorporated by reference in this application as though fully set forth herein.

FIELD

This patent specification is in the field of systems for handling information by computer and more specifically relates to an enhanced system for handling heterogeneous items of information to store, manage, customize, organize and/or deliver such information regardless of its source and type in particularly efficient, easy-to-use, and intuitively understood.

BACKGROUND AND SUMMARY

Traditional information management systems store and retrieve documents on the basis of attributes such as the name and storage location of a document. This, however, can get very unwieldy in typical usage, as more and more names and locations of documents become a part of the storage and retrieval scheme. Although it is possible in some cases to search or order documents by other attributes, such as content and time of creation or revision, it may still be necessary to specify which file folders, directories, or storage devices to search. If a user no longer remembers how a particular item of information was stored in a traditional system, it may be difficult or impractical to retrieve it efficiently.

In an effort to alleviate these and other concerns with traditional storage and retrieval systems, and to provide a more effective and natural approach that better fits the way people tend to work with and think of items of information, a new system described herein uses approaches that rely primarily on an intuitive, time-associated way of dealing with information. The system is stream-based in that it creates

time-ordered streams of information items or assets, beginning with the oldest and continuing through current and on to future items. An information item or asset in this system can be any type—a file, an email message, bookmark, IRL, memo, draft, scanned image, calendar note, photo, shopping list, voicemail, rolodex or business card, a video clip, etc. When a user tunes in a stream, ordinarily a receding parade of documents appears on the screen. The closest are nearest in time. When a new document arrives, for example when a new email message comes in, it appears at the head of the stream, at the front of the parade. (When a newer message arrives, it steps in front of the parade.) Further-away documents are older.

Ordinarily, a user stands at the line current in time and looks into the past, but the stream also extends into the future. If the user has a meeting next Tuesday at 10 AM, a note to that effect goes into the stream's future, and a note about a meeting Wednesday goes in the stream in front of the note about next Tuesday. Documents in the stream flow steadily onward, as time does. Documents in the future part of the stream flow toward the present; documents in the present flow toward the past. Newly arriving documents push older documents further into the past.

The receding parade of documents is an efficient way to present information on a computer screen. The display uses foreshortening for a perspective effect to pack more information into limited space. For easy browsing, when the user touches a document on the screen with the cursor, a summary of that document with a thumbnail vies appears immediately, without requiring clicking or other user action, as a browse card—a dedicated small window besides the receding parade of time ordered documents. The user controls the displayed stream with VCR-type controls, to move forward or back, to go toward or to the beginning or the end of time in the stream, to now, or to any date or time, past or future.

An item of information in a stream need not be given a name, or a designation of storage location. In a traditional system, a requirement that all documents have names can have implications beyond the necessity of inventing and remembering names. For example, emails may not have names of their own but may need to be stashed inside some other file; to search for an email the user may need to go to this special mail file and search that file. In the system disclosed here, items of information such as emails do not need to be named and can be searched along with any other types of information items.

Searches in the disclosed system can be by a combination of three methods, search, browse, and time-order.

Time-order in itself often makes it possible to locate documents. Often the user needs a document that showed up recently, this morning, or two days ago, or at some time that can be pinned down with some degree of accuracy. Time-order together with browsing through the stream (and its glance views) makes it possible to glance quickly through the documents that are from the approximate time of interest and quickly pull out the right one. (While traditional systems can time-order documents it often is difficult to intersperse in the list all recent emails, news updates, bulletin-board postings, URLs and other documents, let alone voicemail messages. Without a browse feature for a stream as disclosed herein, such a list can be of little value, whereas with browse and an all-encompassing stream that gets updated promptly with new material, one can sweep over large numbers of documents, get instance glances (summary, thumbnail, etc.) of each and find the right one fast.)

When searching in a stream in the disclosed system, the user gets a new stream—a substream. One can search on any

word or phrase, as every word in every document is indexed, on document types and metadata, and on time-related data (e.g., show me all email from last March). If the user searches for an entity called Schwartz Bottling, the new substream will the narrative or documentary history of all dealings with that entity—first contacts, subsequent internal documents or communications, reports, calendar items, and so on.

A substream in the disclosed system is in some ways similar to a folder or directory in a traditional system. Instead of a "Schwartz Bottling" folder in which the user has put documents by so naming them, he/she has created a substream with those document, and can save it for later use or create it again as needed. The substream can do all a folder can but is much more powerful than a folder. A substream collects documents automatically; the user has to put documents in a folder by hand, one by one. A substream can persist in that it continues to trap newly created or received documents that match it. If a user looks at the "Schwartz Bottling" substream tomorrow, she/he may find it has grown to include a new email or other documents that were interspersed automatically. A substream can tell a story, and include the future. A substream is non-exclusive, in that a document can belong to many substreams. A folder in a traditional system imposes on computers many of the obsolete, irrelevant limitations of a physical filing cabinet drawer or folder. A substream is an organizational tool that can make more efficient use of computer characteristics than an analog of filing an retrieving physical documents.

One reason for the efficiency of the disclosed system is that it handles all types of different documents, or items of information, in essentially the same way, even if the document is of a type or format unknown to the system. Each document when created, received or otherwise encountered is treated consistently according to a universal Document Object Model (DOM). As described below in more detail, the system processes the document to create its Document Object Modes that includes various aids such as significant information about the document including items such as summary, type of document, thumbnail of the document, who is the document' owner, who has permission to access the document, keywords, command options, time stamp, index, etc. This creation of a document's DOM is done automatically, although the user can aid the process. It can be done by a translator agent or programmatically.

The system creates a glance view or browse card of each document that has the same overall format to make searching for and working with a document more intuitive but also is specific to the documents in many ways. One important difference from traditional systems is that the browse card has command buttons that match the type of documents. While the command set for traditional systems may use the same command button set for different types of documents, in the disclosed system the command set that shows in the displayed browse card is specific to the document—it has the unique combination of command buttons that make sense for that document. The command buttons unique to the browse card can be shown on the card itself or separately.

The browse card comes on the screen automatically when the cursor is over the corresponding document in the displayed stream; the user need not take any other action such as clicking on the document or taking an action calling a program that can open or work with the document.

The universal DOM of a document is created automatically as a new document of any type is added to the basic stream of information items. It is done for any existing, legacy documents, when the system is first installed on a computer, and is done as any additional documents are created or otherwise come in. Metadata such as owner, date, access permission and keywords are created as part of this automatic process.

Access permission is a part of a document's metadata, so permission levels need have the constraints of traditional information handling systems where a group or an individual typically has access to all documents in a particular folder or directory, or has a particular type of access to a folder.

Search results are integrated into a substream, at the right place, when and as they become available. The user can start using an incomplete substream and watch it build up. If the search must extend over a number of computers or even servers, and some are unavailable at the time, the results that come in when any become available are integrated into the substream at the right places.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a screen that can serve as a default view when a software product according to a preferred embodiment is opened on a computer; the labels that are added are not normally a part of the displayed screen.

FIGS. **2-8** are flowcharts illustrating processes in an example of a preferred embodiment.

FIGS. **9** and **10** are examples of configurations in a preferred embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

FIG. **1** illustrates a default screen seen on a PC or other equipment working with the disclosed system. It can show up upon turning on the computer, or upon calling the disclosed system. As seen in FIG. **1**, the screen illustrates a receding stream of documents, with the most recent documents at the front. Passing the cursor over a document in the stream causes that document's "glance view" or "browse card" to appear on the screen. The glance view of a document is so labeled in FIG. **1**. The screen also includes the following features appropriately labeled in FIG. **1**: (a) the Search Field is an area in which the user can type one or more words for which the system will search in documents (information assets) in the displayed part of the stream and/or in additional information assets that might not be displayed; (b) the Main Menu is where the user sets preferences, finds help information, logs out, and/or performs other operations; (c) the Header contains information such as links, command buttons and choice boxes used to navigate; (d) the Stream View Options allow the user to configure the presentation of the stream of information assets; (e) the Document Glance allows quick scanning of information assets that are visible on the screen, and presentation of more detailed information on the selected information asset; (f) the Type Glyphs identify the nature of an information asset at a glance (e.g., a Word document); and (g) the Thumbnails is a graphic representation of the type of document (e.g., an audio file, an email, an event, etc.). The User Guide published by the assignee hereof (a copy is submitted concurrently with the filing of this application with an IDS form) further describes the operation of a relevant example and, together with the programs contained in the compact disc submitted herewith, provides a more detailed disclosure of a preferred embodiment.

Certain particularly novel features of the disclosed system are described below by reference to flowcharts and block diagrams. More detailed information on a particular example of implementation of these and other features of the system

are evident from the software on the attached compact disc, which is the best mode known to the inventors at the time of filing this patent application.

FIG. **2** illustrates creation of a universal data object model of a documents in accordance with a preferred embodiment. This is an important part of the disclosed system that helps make possible the efficient handling of heterogeneous document types in a manner that users find easy and intuitive. A document object model (DOM) can be thought of as a document shell of the information asset (IA) that contains, anon other items, a thumbnails of the information asset, permission rights, and metadata. The DOM is created from the IA and is stored in a desktop computer and/or a server, either independently of the IA itself or with a replica (copy) of the IA. From there, the system makes the DOM (with a pointer to its IA or replicated IA) to the desktop user or to users that have access to the document through some computer connection.

As seen in FIG. **2**, the process of creating a DOM starts with the uploading at step S**201** of information assets (documents) through a browser or a client software application, or step S**202** with uploading using a software application agent called Doc Feeder in a specific embodiment of the disclosed system. At the following steps, which need not be performed in the order of their description below, a DOM of the IA is created. The IA uploaded at step S**201** or S**202** can comprise structured or unstructured data. At step S**203** the process determines the content type of the IA, e.g., if it is a type that the system recognizes. If it is, the system includes content-type specific metadata in the document's DOM: MIME/content type information, a glyph of the application that creates/views the content-type, and/or the system assigns other content-type data to the DOM shell. If step S**203** determines that the IA is an unknown content type, it assigns to the DOM a content-type for "unknown content-type.". Step S**204** extracts text from the information asset, for example, in a text document, this step extracts the text of the document. Step S**205** extracts text that may not be within but may be associated with the information asset, for example, the time stamp of the document, the owner of the document, and possibly other textual information that is or can be associated with the document. Other possible examples are attributes of the IA such as file reference path, database/repository path, file metrics such as size, encryption, other identification information, etc. Step S**206** generates a thumbnail picture of the IA. The thumbnail can be a reduced-size picture of the document, for example of the first page, and can be converted to a graphic image format. Other examples of thumbnails are JPEG, MPEG, BMP, GIF, AVI, or other still or moving image files representative of some aspect of the IA. Step S**207** produces an automatic summary of the IA, e.g., a replica of its first 500 words, or first 10 sentences, or some other information copied or otherwise derived from the IA. Step S**208** creates a permission list unique to the IA that defines the owner of the IA (e.g., its creator), and lists of people or entities and groups that can access the IA or the DOM of that IA for reading and/or writing purposes. This permission list can be defined by the user for the particular IA or for a class of IAs, or can be created automatically, e.g., by software agents called Doc Feeder or Crawling agent in a particular embodiment of the described system, or by programmatic mapping such as LDAP, Active Directory, NTDS or some other mapping. Alternatively, at least for some documents, the permission list can be default setting.

Step S**209** assigns keywords to the information asset. The software agents Doc Feeder or Crawler can assign keywords, and the user can manually assign or add keywords. Step S**210** generates and assigns to the IA a Globally Unique Document ID, e.g. as 64 bit code unique to the IA. Step S**211** determines and assigns to the IA document operations that are unique to the IA. Depending on the IA, these operations or command buttons can be basic, such as "View" and "Reply." They can be content-specific, such as "Play" for multimedia information assets. They can be solution-specific, such as "Fax of Purchase." They can be user-specific, such as "Delete" allowed to only certain users. An important point is that the operations or command buttons assigned to a particular IA match the IA and need not be the same for different information assets, as is the typical case with traditional information management systems. Step S**212** assigns optional operations or command buttons to the IA. They include, for example, commands to send the IA to an optical character recognition (OCR) service that can be a separate service, IP, HTTP-based or an asynchronous operation. Alternatively, the optional operation can be another OCR operation that can perform OCR on a selected part of the IA, or on digital graphic portions or can involve multi-part associations. At step S**213**, the information asset is submitted to an indexing engine (asynchronous service) Again, this can be a separate service, IP, HTTP-based. This step can index all or selected fields of the IA, including but not limited to the IA summary, title, permissions, IA text, keywords, time, metadata, and content-type. At step S**214** the DOM created as described above is submitted to a storage service. This can be a database that is a file reference with a pointer to the actual location of the IA on a network or a local file system, or it can be a database that contains the actual IA in a repository such as a user's computer or a centralized repository. The document object model so generated is made available for use in step S**215**.

FIGS. **3** and **4** illustrate methods of creating document object models from information assets. As seen in FIG. **3**, three type of information assets are involved—new information assets **301**, modified information assets **301**, and deleted information assets **303**. All come to a file system **304**. At step S**305**, agents specific to the disclosed embodiment of the system known as Scopeware 2.0 translate the IA into a DOM, i.e., create a DOM shell for the IA, with attributes as discussed in connection with FIG. **2**. At step S**306**, Scopeware agents translate the IA modifications into an updated DOM and time-stamp the change so the new time-stamp becomes a part of the DOM and the modified IA can be places in the stream of documents at a place reflecting the new time-stamp. At step S**307**, Scopeware agents execute actions for removing the deleted IA from the repository of documents. The display, such as that seen in FIG. **1** reflects the actions takes at steps S**305**, S**306** and S**307**. As a result of step S**305**, the stream on the display shows at **308** the new IA (provided the time period where the new IA fits is being displayed). As a result of step S**306**, the modified IA appears at **309** in its correct place in the displayed receding stream of documents. As a result of step S**307**, the deleted documents is removed at **310** from the displayed stream, and the remaining In FIG. **4**, a programmatic information system received new, modified and deleted information assets for storage and distribution to appropriate translation agents as illustrated. In other respects, the FIG. **4** arrangement corresponds to that of FIG. **3**, so the description of corresponding portions will not be repeated.

At least some of the document object model created as described above becomes a part of a glance view or browse card of the type illustrated in FIG. **1**. An important feature of the system disclosed here is to conveniently display such a glance view in a natural and intuitively accepted way to facilitate operations.

Traditional user interfaces for computers typically present lists or graphical icons of "documents" (including but not

limited to computer files, emails, web pages, images and other types of electronic information). These lists and icon displays provide only a limited amount of information about the document—typically, title and application type only, although additional information as well in some cases. This can make it difficult for users to identify the document without downloading and/or opening the document with its associated application. For example, in Windows 2000, the user interface displays a small temporary pop-up window of the document's title, application type, author and size when the user hovers his cursor on the document icon; however, the pop-up window appears only after a brief delay, usually 1-2 seconds and is for documents that are on the screen at the time, which tend to be a small part of the many documents typically stored in or accessible through a user's computer.

In contrast, the disclosed system creates a pop-up window for heterogeneous documents of known and unknown application types that appears instantly, as perceived by the user, as he/she hovers the cursor over the document's representation in the user interface. In the example of FIG. **1**, this representation is an index card in a cascading flow of overlapping index cards (called "browse cards"), and the pop-up window is called a "glance view". This glance view not only contains the document's title, application type and owner, but also may contain rich multimedia cues (such as a thumbnail image of the first page of the document, a WAV or MP3 preview of an audio file, or an animated GIF preview of a video file), text summaries and document operations specific to the document's application type and access permissions. For example, if the user has write permission for a document, the "Edit" operation will be visible and available; however, if not, the Edit operation will not be visible or available. These document operations are interactive, allowing users to select available operations directly.

Referring to FIG. **5** for an illustration of the instantaneously dynamic, tailored, and interactive document glance view feature of the disclosed system, at S**501** a user hovers his or her computer cursor over a document's browse card. Essentially instantly, at least as perceived by the user, and without any mouse clicking or other action on the part of the user, step S**502** processes the information needed for a glance view to appear on the screen, and at S**503** the glance view appears next to the browse card, using a technology such as Dynamic HTML. If the user clicks on a document's browse card, as detected by the test at step S**504**, and as executed by the user at S**505**, step S**506** causes the glance view to become fixed and step S**507** causes it to remain in the display. The glance view does not change until the user clicks on another document's browse card. If the user does not click on any browse card, as determined by the test of step S**504**, the glance view will instantly change as the user moves his cursor over other browse cards, to reflect the glance view of the underlying browse card. If the user has clicked on a browse card to fix the glance view as a stationary window, the user can then select any of the visible and available document operations, by taking the "yes" branch of step S**508** and selecting at S**509** an available operation (as earlier described, the operations or command buttons that show are specific to the document). At step S**510** the system executes the selected operation (command) and the display reflects this at S**511**. If at step S**508** the user takes the "no" branch, she can continue to hover the cursor over the stream of browse cards and repeat the process, at step S**512**. If at S**504** the system determines that the user has not clicked to fix a glance view, the glance view information essentially instantly changes at S**513** as the user moves the cursor over other browse cards, and the new glance views appear on the screen at S**514**.

FIG. **6** illustrates a process involving another important feature of the disclosed system—granular permissions for access to information assets that allows clients to receive seamless and uniform access to contents without necessitating changes to existing network security and access rights. In traditional systems, a network administrators typically would grant access to specific network drives and file folders. The permission typically would allow a user to access the entire folder or drive, or would deny access to an entire folder or drive, rather than to a particular information asset or document.

In the disclosed system, each information asset is accessible through specific access permission for each client or designated group of clients. Examples of access stage permissions are read, write, and aware. Read permissions allow a client to view the full information asset. Write permissions allow the client to view and edit the'document. Aware permission alerts the client that an information asset exists, for example by providing a document shell in the client's stream of documents, but does not allow the client to view or edit the document. A group of clients who want to collaborate on a project or event can establish a designated group that can be assigned permissions to relevant documents for the project or event. Thus, each member can receive real-time additions to his or her stream of documents and information assets are posted. The clients can assign permission to the other group members themselves, by so designating the appropriate documents to be shared, without involving a network administrator. Some documents, such as personal to-do lists, can be accessible only to a specified user, but the user can change this at any time to allow access, full or partial, to other designated persons. Assignments of permissions for access can be done as granularly as an individual client level or individual document, or as diffuse as a departmental or enterprise level.

As seen in FIG. **6**, an information asset **601** can have permission levels assigned to it in several ways. At step S**602**, a software agent such as Doc Feeder can automatically assign permissions; at step S**603** a programmatic system such as SDAP, Active Directory, Access Control Lists, NT DS, of some other system assigns permissions to the document; and/or at step S**604** the user manually assigns permissions to the document. Examples of processes relevant to different types of permissions are: step S**605** grants access to all public users of the system; step S**606** assigns permissions to groups as illustrated; step S**607** assigns permissions to specific groups as illustrated, and step S**608** freezes permissions and does not allow the document to be changed. The display, of the type illustrated in FIG. **1**, can provide information representative of the permissions, as illustrated at steps S**609** thorugh S**612** in FIG. **6**.

Another important feature of the disclosed system is illustrated in FIG. **7** and pertains to integrating search results from distributed searches. In traditional systems, search requests in a client/server model with a central index usually return a single, well-defined results set. In a peer-to-peer network, however, search results may come back to the "Source" computer (the computer that issues the search query) in a haphazard manner because of network latency (variable traffic speed and bandwidth across a distributed network) and variable peer presence (peer computers can be turned on and off, or removed from network at times).

The disclosed system asynchronous responses to a distributed query across a peer-to-peer network of computers to integrate the results from diverse sources, arriving at different times, and comprising diverse types of documents, into a single unified results set. One preferred embodiment leverages the time-ordered presentation interface earlier described

in so that search results are integrated into a time-ordered stream according to each document's original time-stamp, regardless of when the document's search results set was received by the Source computer.

As seen in FIG. **7**, at step **701** a user at a Source computer selects peer computers ("Peers") across which the distributed search will be performed. If the test at S**703** determines that there is no central registry with peer hookup, and the test at S**704** determines there is no user-specified IP address of peers, the process returns to S**701**, where the user can specify addresses or they can be provided in some other way. The central registry with lookup of Peers can involve Online/offline status, IP/DNS resolution servicejiand Optional public/private key authentication. When the test at S**703** or at S**704** leads to the "yes" branch, at step S**705** the Source computer sends out a search request that travels to each selected Peer in the network. At S**706**, each Peer that receives the search request queries its index for documents that match the search criteria, and at S**707** the peer computer then sends its results set back to the Source computer. The response can be XML-based, a binary byte stream, or an in-band and out-of-band transfer. At S**708** the Source computer takes the results set from each Peer and builds a single collective results set. In a preferred embodiment, this collective results set is organized as a time-ordered stream of documents, as seen in FIG. **1**. This can involves an on-the-fly browser combination with XML & XSL with time-sort algorithm, XML to presentation layer with time-sort algorithm, and in-band and out-of-band transfer. Importantly, at S**709**, the Source computer continues to expand this collective results set, essentially in real time as it receives additional results sets from Peers until all Peers have responded or some other relevant event has taken place. At S**710**, the collective results are displayed as soon as results have come in at the Source computer, and the display is updated as additional results come in, even when a Peer that was off-line comes on line and sends results at a later time.

Yet another feature of the disclosed system is a particularly convenient tri-state tree. In a single scrolling tree directory of the contents of a hard drive (or hard drives in a network), a user may want to select "Parent Folders" (folders containing subfolders) and "Child Folders" (subfolders contained within a folder) that can be further operated on. This feature allows users to select folders in one or more of the following combinations:

1. All Parent Folders and all Child Folders
2. Some Parent Folders and all their Child Folders
3. Some Parent Folders and some of their Child Folders
4. No Parent Folders and no Child Folders (the do nothing option)

This selection tree has useful application beyond the particular example of information handling disclosed here; it can be used to select folders for any computer operation. For example, it can enable users to discretely select software application or operating system components to install or remove.

A single scrolling tree directory of Parent and Child Folders that can expand and contract to show the contents of Parent and Child Folders is known—Microsoft Windows Explorer is an example of one. A Tri-State Selection mechanism also is known—Microsoft Add/Remove Windows Components is an example of another way of selecting various Parent and Child Folders. However, the Microsoft Add/Remove Windows Components feature does not display all Parent and Child Folders within a single scrolling tree directory; Child Folder and other contents of a Parent Folder are displayed in a separate window only after the user clicks on a Details button. In addition, only the contents of one Parent Folder can be displayed at a time.

The Tri-State Selection Tree described here combines the elements of a single scrolling tree directory with a tri-state selection mechanism in a new and unique way to enable users to discretely select specific Parent and/or Child Folders all in one single view.

Referring to FIG. **8** for an illustration, at step S**801** a user is first presented with a tree directory of the highest level of Parent Folders on a hard drive or network. At S**802** the user can expand the tree directory to show Child Folders by clicking on a plus/minus sign next to each Parent Folder, and the directory so expands at S**803**. At S**804**, the display shows a check box next to each Parent Folder (e.g., to the right of the plus/minus sign). By default, all check boxes are empty, indicating that no Parent or Child Folders are selected. If at step S**805** the user clicks on a check box once, the process at step S**806** selects the marked"/" Parent Folder but none of its Child Folders are selected, and step S**807** shows this on the display. If at step S**808** the user clicks the check box a second time, the slash mark is replaced by an "X" and all the Child Folders' check boxes are then selected and grayed out at S**809**, indicating that all Child Folders are selected for that Parent Folder, and this is displayed at S**810**.

Thus, by expanding the tree and clicking on check boxes, the user can systematically and efficiently select a discrete number of folders on which to perform an operation.

Yet another feature of the disclosed system is an arrangement of a redundant array of inexpensive servers (RAIS). Processing of a large set of information or document requires benefits of a centralized architecture—reliability and scalability, and RAIS is a novel approach to provide benefits of a centralized architecture—namely reliability and scalability with numerous inexpensive computers. Thus, RAIS can deliver essentially infinite scalability, can allow inexpensive smaller computers to be used to solve enterprise computational problems rather then expensive larger platforms, cheaper/faster.

For example, consider:

Set of Information, D, with specific documents D1, D2, D3; D{D1,D2,D3}

RAIS of N×N size here with N=3; RowN,ColN

Replication factor is number of columns

Scalability factor is number of rows

1. Here N=3, with 9 computers

	Col1	Col2	Col3
Row1	A	A	A
Row2	B	B	B
Row3	C	C	C

2. To post a Document, Dn, one copy is sent to a sub-server in each ColN, so

	Col1	Col2	Col3
Row1	A(Dn)	A(Dn)	A(Dn)
Row2	B	B	B
Row3	C	C	C

3. Thus Dn is replicated N times (N=3) and thus if Col1: Row1 computer is unavailable there are two other computers with the same Dn. This is RAIS replication.

4. To post a universe, or set of documents, D{D1,D2,D3}, can use simple (round-robin) or complex (latency, closest path, spanning tree) routing, sending each document to a different RowN.

	Col1	Co12	Col3
Row1	A(D1)	A(D1)	A(D1)
Row2	B(D2)	B(D2)	B(D2)
Row3	C(D3)	C(D3)	C(D3)

5. Thus to reassemble the entire universe or set of documents, D, need to send a request to each RowN. To reconstruct, D, for an N×N RAIS requires N request/responses.
6. Multiple smaller requests can be used instead of one mammoth request. This reduces latency, bandwidth and process constraints. This is RAIS scalability.
7. Note that any one of the computers in Row1 can be used to re-construct the total set D found in Col1. For example, if Row1:Col1 computer is unavailable, then Row1:Col3 computer has a copy of the data. In fact, D is can be constructed from any arrangement that completes a COIN.
8. To increase either replication or scalability simply increase N.

Scopeware Software Agents, either desktops or servers, can be installed on each computer in a RAIS matrix to achieve this functionality.

The disclosed system can be implemented in a variety of ways in terms of physical information storage—for example, physical information storage can be centralized or decentralized. Decentralized storage, physical storage of information with multiple servers and/or clients, is possible through network agents called Doc Feeders, which may be located at a server or client level. The Doc Feeder allows a storage location of a client, for example a file folder on a desktop hard drive, to be included in the system level data repository for use throughout an organization or enterprise. Depending upon implementation, the Doc Feeders can replicate the information asset (IA) to a server or maintain a constant pointer to the physical storage location while populating the system with the document object model (DOM). As earlier described, a DOM is a document shell of the IA that contains, among other items, a thumbnail of the IA, permission rights, and metadata. A DOM is created from the IA and placed on the Scopeware server, either independent of the IA or with a replication of the IA. From there, the Scopeware server will share the DOM (with constant pointer to the IA or replicated IA) with other connected system servers and clients in order to make the IA available to all clients connected to the network. Thus, the system servers and network agents (Doc Feeders) act as document proxies for both storage and retrieval of IAs.

In addition, the system servers within the network need not be physically close in proximity. For example, a client in a truly global organization with locations and system servers on several continents can query and retrieve sales results across all system servers and clients through a federated search. In essence, the disclosed system creates a virtual store from all documents accessible to any system server or client either centralized or decentralized.

The physical information storage of the disclosed system follows three models: duplication, replication, and document reference. The duplication model physically stores a duplicate IA on the parent Scopeware server that was created by the client. Other clients polling the parent Scopeware server have full access to the IA, depending upon permissions, whether or not the original document is available from its native storage location (i.e. client PC is turned off). The replication model replicates the IA from the parent Scopeware server to the peer Scopeware servers within a federated network. All clients within the federated network have full access to the IA, depending upon permissions, whether or not the original document is available from its native storage location (i.e. client PC is turned off). An example of the replication model is the concept of a redundant array of inexpensive servers. This concept, which is described in detail in the distributed enterprise model, utilizes client machines in place of a singe server. The document reference model "parks" only a DOM of the IA on all Scopeware servers and maintains a constant pointer to the actual physical location of the IA rather than storing a full copy of the IA on the Scopeware server. Other clients will only be able to gain access to the IA when the physical location of the IA is connected to the network (i.e. client PC is turned on).

There are to primary types of streams in accordance with the disclosed system: Bottom-Up and Top-Down. Through the use of both Bottom-Up and Top-Down methodologies, Scopeware creates a living stream for the client with new DOMs appearing automatically as content arrives. The Scopeware distributed enterprise model can make use of both server-based resources and client-based resources where appropriate. Both types of streams can be used simultaneously and interchangeably.

Bottom-Up streams are comprised of information collaboration formed by ad-hoc groups of Scopeware clients. A bottom-up stream is composed of information created by the clients of a transitory group. Information shared and created by this group is be replicated via point-to-point connections (i.e. from client PC to client PC). In this way, bottom-up groups can form and disperse frequently, and without notification, while its members will still have access to the shared information. FIG. **9** illustrates this configuration.

Top-Down streams are more permanent, generally more administrative streams or collections of information, such as company-wide distribution lists, or groups like 'Accounting' and 'Development'. In these groups, information is "parked" to the server from the desktop. The server then sends the information to other known servers. Each client maintains a polling connection to the server to retrieve "parked" documents that have recently arrived from other remote servers or from local clients. FIG. **10** illustrates this configuration.

As earlier described, the user interface within the Scopeware product portfolio has unique characteristics. The DOM provides certain information that allows quick perusal of the information retrieval results via a proprietary "browse card" or "glance view" which is similar to an index card that contains data on the underlying IA. A unique "browse card" or "glance view" is created for each IA. The "browse card" or "glance view" includes metadata for the document, which is comprised of a title, identification number unique to Scopeware document referencing, date/time stamp, and owner information. The "browse card" or "glance view" also presents a thumbnail image of the IA and a summary of the IA contents. Finally, the "browse card" or "glance view" contains a list of operations appropriate for the IA's application that include, but are not limited to, copy, forward, reply, view, and properties.

The "browse card" or "glance view" arrives in the stream of those clients that have permission to view the IA. The owner can grant access to other clients or groups by granting read, write, or aware permissions through the properties of the

"browse card" or "glance view." Permission can be granted as granular as an individual-by-individual basis from the DOM, or through predetermined administrative groups via the Scopeware server.

The "browse card" or "glance view" is presented in a time-ordered sequence starting in the present going back into the past. The "browse card" or "glance view" is available in a number of views. The primary view is the stream. Other formats include a grid, Q, list, and thumbnails. The various views address the client's personal preferences for accessing time-ordered content in their most logical way. These views all contain the information presented in a "browse card" or "glance view" but are organized in a different method. Other specialized views include the address book and calendar.

An advantage of the "browse card" or "glance view" approach is the ease of browsing, searching, and retrieving IAs. In the stream view, the "browse card" or "glance view" of each IA are aligned much like cards in a recipe box. For each item, the title and application icon are viewable on the "browse card" or "glance view" in the stream. When the client passes over the "browse card" or "glance view" in the stream with the mouse pointer, the full "browse card" or "glance view" is presented to the client for easy viewing. From the "browse card" or "glance view," the client can perform any of the aforementioned actions available to the IA, subject to permission access.

The disclosed system is suitable for a number of computing models servicing multiple clients including a single departmental server model, an enterprise server model, a distributed enterprise model, and a peer-to-peer model (absent a dedicated Scopeware server or common server). In addition, the software enables wireless computing independent of or in conjunction with any or all of the aforementioned models. Wireless clients include WAP enabled phones, PDAs, Pocket PCs, and other similarly capable devices capable of receiving and transmitting data across a network. All of the Scopeware Implementation Models make use of the components previously discussed, providing consistent interface available across different computing topologies, from monolithic single servers to peer-to-peer collaboration.

Access to the IA contained in the Scopeware repository can be achieved through two methods. The first method of access is through the thin-client method. The thin-client method utilizes a web browser, such as Microsoft's Internet Explorer or Netscape's Navigator, on the client device to gain access to the Scopeware repository residing on the Scopeware server. The second method of access is the desktop-client method. The desktop-client method involves a local installation of Scopeware on the client device. The client device is then capable of performing the storage, retrieval, extraction, and processing of IAs as they are introduced to the Scopeware repository. All the models below can utilize either method of access to the Scopeware repository, however the distributed enterprise and peer-to-peer models are optimized with the desktop-client method.

Single Server Model. A single server model makes content on one Scopeware server available to any client connected to the departmental server. The Scopeware software creates a unique DOM that represents to the user interface the relevant details of the IA physically stored by the server or client. Thus, when a client connected to the network requests access to and retrieval of IAs through Scopeware, the client can view all documents contained within the network that satisfy the query parameters and access restrictions regardless of the document's native application. The documents available include those stored locally by the client, those saved to a central storage location, and those stored by peer clients with Doc Feeders connected to the shared server.

Enterprise Server Model. In an enterprise server model, where multiple Scopeware servers are installed, federated access to and retrieval of IAs across the network is enabled. In federated information sharing, a client asks one Scopeware server for IAs that may reside on it or one of many connected peer Scopeware servers. In this model, the actual IA may reside on any network-connected client, the Scopeware server, or a centralized data storage location. Transparent to the client, the Scopeware servers shuffle the retrieval request and access restrictions to present a single, coherent stream to the client via the presentation architecture previously discussed (within the original patent document).

Distributed Enterprise Model. A distributed enterprise model utilizes the clients for storage, retrieval, and processing of IAs. Through the use of directory monitoring agents, similar to network agents, the physical location of an IA need not be on the Scopeware server, but rather can reside with any client. The Scopeware servers take on a secondary role as administration servers and content parking lots. This model pushes the processing tasks to the clients while using the servers to shuttle IAs throughout the enterprise. The indexing engine, thumbnailing engine, lightweight storage database will be based at the clients.

Taking Scopeware beyond distributed networking and the federated architecture—into a more distributed approach will be straightforward, given the way that the system has been designed. Key elements of the next stage of deployment are distributed document processing and scalable server arrays.

Distributed document processing consists of two different approaches. First, when information was created physically on a desktop machine, but was part of a larger application and intended for storage on a server (rather than on the desktop), the Desktop facilities could do the document extraction, indexing, thumbnailing, etc., and post the results to the Scopeware Server. Second, a Scopeware Server that was handed a document (perhaps from an OCR process or from a central email application) could hand the document off to an available Scopeware Desktop for the same processing. These strategies relieve the processing load on the Scopeware Server and leave it free to focus on handling searches and stream integration, allowing a given Scopeware Server to handle a much larger user load.

When an organization needs to support central processing of large document bases—and needs the reliability, accessibility and security of a centralized architecture—Scopeware Servers will support deployment in a novel architecture we have named RAIS—a redundant array of inexpensive servers.

In this architecture, imagine a square array of desktop machines—call each one a "sub-server." The array as a whole comprises the Scopeware Server. (This does not require wiring together an actual array or cluster; any interconnect such as a Ethernet sub-net or even HTTP over a broader network will work.) In these arrays, columns of servers provide redundancy for storage, while rows (within columns) provide redundant points of distribution.

To post document D, one copy of D is sent to a sub-server in each column of the array. To replicate everything five times such that losing any data requires the loss of five sub-servers, five columns are used. The number of columns in the array is managed to support exactly the degree of replication (and redundancy) desired. The write processes can be managed in a number of ways to ensure that the different rows in the columns are balanced.

To send a polling message or search request ("give me all the latest stuff"), a request is sent to each sub-server in one

column (note that the means to do this transparently to the user is an extension of the federated search technology). Each column of sub-servers absorbs one copy of every posting (because any write has gone into at least one row of the column); therefore, all the sub-servers in any one column collectively have copies of everything. Just a "replication factor," is chosen for data redundancy, a "distribution factor" is chosen for responsiveness and for data management, representing the number of rows in any column. To get ten small responses to a search request instead of one big response, or to distribute the total data-storage burden over ten machines instead of one, the array is implemented with ten sub-servers in every column.

The entire "Server" can be run with only one row (resulting in replication, but no distribution) or with only one column (resulting in distribution but no replication). In the limit, row size=column size=1, and the effect is to have a single conventional server.

This approach to distributed processing, scalability and reliability for large applications allows arbitrary sets of "smaller" computers (single/dual processor, inexpensive memory and disk storage) to be used in place of very large, expensive machines. This allows the application platform to be designed to the reliability and access requirements of the particular application, and then scaled incrementally (by adding more small machines into the array) as the actual application grows in terms of users served or information managed.

Distributed document processing and server arrays will give Scopeware almost infinite scalability while maintaining compatibility with early solutions or architectures. In addition to adding greater reliability, this architecture will support very large information processing applications. This will allow enterprise-scale, top-down applications—inbound support/sales email handling, customer service or even IRS-scale tax document processing.

Distributed document processing (with Scopeware Desktop) could be combined with either a "conventional" (1 processor array) Scopeware Server or with a more powerful array. This will allow organizations to create departmental or workgroup level solutions that can grow into enterprise applications if necessary.

At the same time, the system will allow users themselves to create self-organizing applications based on their specific and current needs. Ad hoc teams can create collaborative spaces that cross organizational boundaries if necessary. These applications can leverage either Scopeware Desktops or departmental-level Scopeware Servers.

Because the system has the architecture and capacity to support any level of centralization or decentralization concurrently, applications and their platforms can be engineered centrally or grown organically, and they can be tailored to the needs of their users and the organization on an ongoing basis.

Peer-to-Peer Model. The peer-to-peer (P2P) model allows multiple clients to share IA directly without the use of a dedicated Scopeware server. The P2P model allows for pure ad hoc collaboration among Scopeware clients. For example, a client can share IA via the Internet with identified Scopeware clients that have permission to access IA from the client, and vice versa. This is similar to the distributed enterprise environment except the dedicated Scopeware server has been removed as a storage, retrieval, and connection mechanism. Instead, Scopeware clients will connect point-to-point with other Scopeware clients through a general network connection such as the Internet.

Using P2P, a client can create a virtual shared stream that looks as though it is stored on a server but is in fact stored only by many clients. Historically, all clients would need access to a shared file folder on a common server in order to share information. With Scopeware, clients can share information that is located on each other's device and are not restricted to a common server or single physical storage location. To illustrate, five clients of Scopeware want to create a shared virtual stream to support a project. They call their group "Team One." Then, when any member of "Team One" posts a document to his or her stream, and marks it "readable by Team One," the system automatically sends a copy to every Scopeware client on the "Team One" list. Each Scopeware client receiving this document pops it into its client's local stream. Thus information created by a client who is a member of "Team One" (and flagged for Team One by the owner) winds up in the local stream of every member of Team One, whether the post is a document, an event (team meeting), task, or contact. It's as if he had sent his posting to a "client" server, and then everyone had polled the server, but in fact there's no server.

The invention claimed is:

1. A method of operating a computer system comprising:

providing the computer system with documents in respective formats according to respective different applications through which the provided documents are generated or modified, which formats differ from one of the document to another for at least some of said provided documents, said provided documents being delivered to the computer system or generated by the computer system;

storing at least some of the documents provided to the computer system in computer storage;

said computer system being configured to automatically generate and store in computer storage respective representations related to the documents provided thereto, thereby forming a main collection of document representations;

said computer automatically generating and storing said main collection of document representations without requiring a user to designate a directory structure, a physical location for storage of document representations or corresponding documents, or another pre-imposed document categorization structure for each of said document representations or documents;

said automatically generated and stored document representations being in a consistent format despite differences in format from one to another of the documents corresponding thereto;

said automatically generated and stored representations of said documents including respective automatically generated time indicators associated with the documents corresponding to said representations;

said automatically generated and stored representations of said documents further including respective automatically generated information relating the document representation to the respective documents provided to and stored in said computer system;

said automatically generated and stored main collection of document representations being unbounded in time and size and being configured to include documents associated with time indicators related to future times as well as to past and present times;

said automatically generated and stored main collection of document representations requiring no fixed beginning or end and being non-transitory and selectively searchable by the computer system;

providing selected search criteria;

causing said computer system to perform a first search of at least said main collection of document representations

according to selected first search criteria, to provide first search results, and to utilize said first search results to generate a first sub-collection of document representations related to a respective sub-collection of the documents provided to the computer system;

selectively causing the computer system to display on a computer screen graphical depictions of only a first portion of said first sub-collection of document representations generated by utilizing said first search results, said first portion corresponding to only a portion of the documents provided to and stored in the computer system;

said first portion of said first document sub-collection of document representations corresponding to a multi-document portion of the documents provided to and stored in said computer system;

said computer system being configured to maintain at least one of said the main collection of document representations and said first sub-collection of document automatically responsive to events subsequent to the providing of said first search results such that additional document representations corresponding to additional documents provided to the computer system subsequent to an initial display of said first portion of the first sub-collection of document representations and meeting said selected search criteria are automatically included in a subsequent display of graphical depictions of one or more portions of said first sub-collection of document representations;

said additional document representations also including automatically generated respective time indicators associated with the documents subsequently provided to the computer system;

automatically showing on the computer screen a display of a glance view of a displayed document depiction while continuing to show on the screen plural displayed graphical depictions of respective plural document representations;

said glance view being an abbreviated version of the document corresponding to the graphical depiction and being indicative of content thereof; and

said showing of the glance view occurring in response to a user designating a displayed document representation by interacting with a screen area currently associated with the graphical depiction, without requiring the user to click on the designated screen area in order to enable such showing of the glance view.

2. The method of claim **1**, in which said interacting with a screen area comprises user interaction with a scroll bar user interface.

3. The method of claim **2**, in which said user interaction with a scroll bar interface causes a selection of which document representations to include in said first portion of document representations.

4. The method of claim **1**, in which the step of causing the computer system to display on a computer screen graphical depictions of said first portion of document representation comprises displaying said graphical depictions using perspective to create the illusion of increasing distance from a viewpoint implied by the displayed depictions.

5. The method of claim **4**, in which said step of causing the computer system to display said graphical depictions comprises displaying the graphical depictions as a partly overlapping stack of document depictions creating the illusions that the depictions gradually recede from the viewpoint.

6. The method of claim **5**, in which said step of causing the computer system to display said graphical depictions comprises displaying the graphical depictions as a foreshortened and receding stack of document representations.

7. The method of claim **1**, in which said step of causing the computer system to display said graphical depictions comprises displaying at least one graphical depiction of a document representations associated with a future time.

8. The method of claim **1**, in which said representations of said documents further include an index of the content of said documents for use in performing said first search.

9. A process of providing a computer system with document management facilities, comprising:

providing the computer system with a collection facility configured to collect a multiplicity of documents from diverse applications, said documents being in respective different formats unique to the respective applications;

providing the computer system with an indicator generating facility configured to automatically generate respective identifying indicators that are associated with said documents and comprise time information;

providing the computer system with a maintaining facility configured to maintain in non-transitory form a main collection of the documents collected by the collection facility and the indicators associated therewith;

said maintaining facility being configured to maintain said main collection of documents unbounded in practical terms consistent with the computer system extent to thereby accommodate many documents and the respective indicators associated therewith in a form that is selectively searchable and retrievable by the computer system;

providing the computer system with a search facility configured to respond to search criteria provided to the computer system by generating, from said main collection of documents, a subcollection of a multiplicity of documents that are a subset of the main collection documents;

providing said computer system with a display facility configured to display on a computer display at one time representations of plural documents that comprise only a segment rather than all of the documents of one of the main collection or the subcollection of documents;

said display facility being further configured to display said representations in an arrangement consistent with the indicators associated with the documents corresponding to the representations that are displayed at the time, wherein at least some of the displayed representations show at least some information indicative of a content of the respective documents, which content-indicative information is in addition to any display of hierarchical data structure or time of creation of modification of documents;

said display facility being further configured to display an enhanced representation of at least one of the documents of the displayed segment while continuing to display other representation of the documents in the segment;

providing the computer system with a live main collection updating facility configured to maintain live the main collection of documents by automatically responding to subsequent events to include new documents in the main collection in response to said new documents being collected by the collection facility in response to said subsequent events, and to include in said indicators stored in the maintaining facility new indicators associated with said new data units;

providing the computer system with an automatic subcollection updating facility configured to respond to said search criteria by automatically including in a subcol-

lection generated in response to said search criteria or new search criteria those of said new documents that meet said criteria; and

providing said computer system with a display updating facility configured to cause said display facility to display a segment that includes representations of at least some of said new documents.

10. The process of claim **9** in which said computer system comprises plural computer platforms at different locations, as least some of said platforms being configured to serve as viewpoints through which to access said main collection of documents through the Internet and to provide the computer system with said search criteria.

11. The process of claim **10** including providing the computer system with an access restriction facility configured to cause the collection or at least some of the subcollections to be accessible to different extents in response to respective different sources of request for access.

12. The process of claim **9** in which said search facility is further configured to generate two or more subcollections of documents in response to respective search criteria, wherein the same document may be included in two or more of said subcollections.

13. The process of claim **9** including providing said display facility with a segment moving facility responsive to inputs to the computer system to change a time span related to the selection of the documents to include in the displayed segment.

14. The process of claim **13** in which said segment moving facility is configured to respond to user input provided through a scrolling arrangement.

15. The process of claim **9** in which said display facility is configured to respond to touching a displayed document representation through a user input by highlighting the representation and providing a glance view of the documents corresponding to the representation.

16. The process of claim **9** further comprising providing the computer facility with a summarize or compress facility configured to automatically summarize or compress at least some of the documents in said subcollection into a single document included in the subcollection or into several documents that are less in number that the documents in the subcollection.

17. The process of claim **9** in which said collection or subcollection of documents comprises respective image frames of video.

18. The process of claim **9** in which said display facility is configured to display said document representations as upright rectangles that partly overlap and give the illusion of receding from a selected viewpoint.

19. An article of manufacture comprising a storage device storing a computer program in a computer-readable, non-transitory form, said program when executed by a computer system causing the computer system to carry out a process comprising:

collecting a multiplicity of documents from diverse applications, said documents being in respective different formats unique to the respective applications;

automatically generating respective identifying indicators that are associated with said documents and comprise time information;

maintaining in non-transitory form a main collection of the collected documents and the indicators associated therewith;

said main collection of documents unbounded in practical terms consistent with the computer system extent to thereby accommodate many documents and the respective indicators associated therewith in a form that is selectively searchable and retrievable by the computer system;

causing the computer system to respond to search criteria provided to the computer system by generating, from said main collection of documents, a subcollection of a multiplicity of documents that are a subset of the main collection documents;

causing the computer system to display on a computer display at one time representations of plural documents that comprise only a segment rather than all of the documents of one of the main collection or the subcollection of documents;

said displaying comprising displaying the representations in an arrangement consistent with the indicators associated with the documents corresponding to the representations that are displayed at the time, wherein at least some of the displayed representations show at least some information indicative of a content of the respective documents, which content-indicative information is in addition to any display of hierarchical data structure or time of creation of modification of documents;

said display further including a display an enhanced representation of at least one of the documents of the displayed segment while continuing to display other representation of the documents in the segment;

maintaining live at least the main collection of documents by automatically responding to subsequent events to include new documents in the main collection in response to said new documents being collected in the collected step in response to said subsequent events, and automatically generating and storing new indicators associated with said new data units;

causing the computer system to respond to said search criteria by automatically including in a subcollection generated in response to said search criteria or new search criteria those of said new documents that meet said criteria; and

causing said display facility to display a segment that includes representations of at least some of said new documents.

20. The article of manufacture of claim **19** in which said main collection of documents is in the form of a stream comprising a time-ordered collection of documents that functions as a diary of a person or an entity's electronic live.

21. The article of manufacture of claim **19** in which said main collection of documents comprises a catalog of data items and said subcollection comprises at least a subset of the data items re-formatted for a particular user.

22. The article of manufacture of claim **19** in which said indicators are chronological indicators.

23. The article of manufacture of claim **19** in which said search criteria are provided to the computer system through a voice interface.

24. The article of manufacture of claim **19** in which said segment is related to a time interval.

25. The article of manufacture of claim **19** in which said causing said display facility to display representations of the document in a segment that includes representations of at least some of the new documents comprises causing the display to so include new documents by updating a current display of at least some of the representations of the documents in said segment.

26. The article of manufacture of claim **19** in which said computer system comprises plural computer platforms, and wherein said computer system is configured to provide access

to some but not all of the documents in a collection or subcollection in response to requests to access by selected ones of said computer platforms.

27. The article of manufacture of claim **19** in which said program is executed in plural computer platforms connected via the Internet.

28. The article of manufacture of claim **19** in which said collection of documents comprises at least one software agent in a groupware system, said software agent visiting other collections or subcollections of documents to set up a new collection or subcollection related to a group event.

29. The article of manufacture of claim **19** in which said display facility displays said representations of the documents in a segment in time order without requiring a designation from a user for time-ordering the specific display of the representations.

30. A computer system with document management facilities, comprising:

a collection facility configured to collect a multiplicity of documents from diverse applications, said documents being in respective different formats unique to the respective applications;

an indicator generating facility configured to automatically generate respective identifying indicators that are associated with said documents and comprise time information;

a maintaining facility configured to maintain in non-transitory form a main collection of the documents collected by the collection facility and the indicators associated therewith;

said maintaining facility being configured to maintain said main collection of documents unbounded in practical terms consistent with the computer system extent to thereby accommodate many documents and the respective indicators associated therewith in a form that is selectively searchable and retrievable by the computer system;

a search facility configured to respond to search criteria provided to the computer system by generating, from said main collection of documents, a subcollection of a multiplicity of documents that are a subset of the main collection documents;

a display facility configured to display on a computer display at one time representations of plural documents that comprise only a segment rather than all of the documents of one of the main collection or the subcollection of documents;

said display facility being further configured to display said representations in an arrangement consistent with the indicators associated with the documents corresponding to the representations that are displayed at the time, wherein at least some of the displayed representations show at least some information indicative of a content of the respective documents, which content-indicative information is in addition to any display of hierarchical data structure or time of creation of modification of documents;

said display facility being further configured to display an enhanced representation of at least one of the documents of the displayed segment while continuing to display other representation of the documents in the segment;

a live main collection updating facility configured to maintain live the main collection of documents by automatically responding to subsequent events to include new documents in the main collection in response to said new documents being collected by the collection facility in response to said subsequent events, and to include in said indicators stored in the maintaining facility new indicators associated with said new data units;

an automatic subcollection updating facility configured to respond to said search criteria by automatically including in a subcollection generated in response to said search criteria or new search criteria those of said new documents that meet said criteria; and

a display updating facility configured to cause said display facility to display a segment that includes representations of at least some of said new documents.

31. The computer system of claim **30** in which said identifying indicators comprise time information associated with the respective documents.

32. The computer system of claim **30** in which said display facility is further configured to display the representations of documents in a segment in time order without requiring a command from a user for time-ordering each display.

33. The computer system of claim **30** in which said display updating facility is configured to include said representations of at least some of the new documents during a continuing current display of representations of at least some of the documents in a segment.

* * * * *

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION

The undersigned hereby certifies that the foregoing brief complies with the relevant type-volume limitations of the Federal Rules of Appellate Procedure and Federal Circuit Rules because the brief has been prepared using a proportionally spaced typeface using Microsoft Word in 14-point or larger Times New Roman Font or another proportionally spaced typeface with serifs and includes 13,958 words, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(f), Federal Circuit Rule 32(b)(2), and Federal Circuit Rule 28(a)(12).

October 6, 2022

/s/ Benjamin T. Wang

Marc A. Fenster
mfenster@raklaw.com
Benjamin T. Wang
bwang@raklaw.com
James Tsuei
jtsuei@raklaw.com
Russ, August & Kabat
12424 Wilshire Blvd., 12th Floor
Los Angeles, CA 90025
Tel: (310) 826-7474

Counsel for Plaintiff-Appellant
Mirror Worlds Technologies, LLC