# EXHIBIT 2

**IN THE UNITED STATES DISTRICT COURT**

**FOR THE EASTERN DISTRICT OF TEXAS**

**TYLER DIVISION**

| | | |
|---|---|---|
| **MIRROR WORLDS, LLC** | § | |
| | § | **Civil Action No. 6:08-CV-88** |
| **Plaintiff,** | § | |
| | § | |
| **v.** | § | **JURY TRIAL DEMANDED** |
| | § | |
| **APPLE INC.** | § | |
| | § | |
| **Defendant.** | § | |
| | § | |

**DEFENDANT'S AMENDED INVALIDITY CONTENTIONS**

Pursuant to P.R. 3-3, Defendant Apple Inc. ("Apple") respectfully submits these Invalidity Contentions. These disclosures employ plaintiff Mirror Worlds LLC's ("Mirror Worlds") interpretations of U.S. Patent No. 6,006,227 ("the '227 patent"), U.S. Patent No. 6,638,313 ("the '313 patent"), U.S. Patent No. 6,725,427 ("the '427 patent"), and U.S. Patent No. 6,768,999 ("the '999 patent"), collectively the "patents-in-suit," to the extent they can be discerned from Mirror Worlds' P.R. 3-1 disclosures and positions taken during prosecution of the patents-in-suit and related patents. These P.R. 3-3 disclosures are not, and nothing in these disclosures should be seen as, an endorsement or acceptance of any of Mirror Worlds' claim constructions, nor an assertion of particular constructions by Apple. Apple expressly reserves the right to propose alternative constructions to those advocated by Mirror Worlds and to challenge and contest Mirror Worlds' claim construction positions.

Prior art not included in this disclosure, whether or not now known to Apple, may become relevant depending on the positions Mirror Worlds asserts and the claim constructions the Court adopts. Apple's ongoing investigations may also uncover additional prior art. Apple reserves the right to modify these disclosures, including without limitation, by adding or

withdrawing prior art to or from this disclosure and/or modifying the charts herein in light of the Court's claim construction ruling, any revised or supplemented infringement contentions by Mirror Worlds, and positions taken by Mirror Worlds in related litigation, reexamination or other prosecution, or as otherwise appropriate. To the extent that Apple obtains additional or further information, Apple reserves the right to supplement these Invalidity Contentions.

The obviousness combinations of references provided in Section I below under 35 U.S.C. § 103 are merely exemplary and are not intended to be exhaustive. Additional obviousness combinations of the references identified below are possible, and Apple reserves the right to use any such combination(s) in this litigation. In particular, Apple is currently unaware of the extent, if any, to which Mirror Worlds will contend that limitations of the claims at issue are not disclosed in the art identified by Apple as anticipatory, and the extent to which Mirror Worlds will contend that elements not disclosed in the specifications of the patents-in-suit and related applications would have been known to persons of skill in the art. To the extent that an issue arises with any such limitations, Apple reserves the right to identify other references that would have made such limitations obvious in view of the relevant disclosures.

Accordingly, Apple reserves the right to supplement or modify these Invalidity Contentions based on further discovery and in a manner consistent with the Federal Rules of Civil Procedures and the Court's rules, including the Patent Rules.

Additionally, as addressed below in Section II, on information and belief, the claims of the '227 patent, '313 patent, '427 patent, and '999 patent are invalid under 35 U.S.C. § 102(f) because the named inventors did not invent the subject matter of those patents. The facts pertaining to the inventor(s)' derivations of the claimed subject matter are being further investigated, and Apple expects that discovery taken in this litigation will further reveal facts concerning this defense. Accordingly, Apple reserves the right to supplement or modify these Invalidity Contentions with respect to derivation as further discovery occurs.

Further, as addressed below in Section III, claims of the patents-in-suit are invalid under 35 U.S.C. § 112 because the claims are indefinite, lack a proper written description, and/or

do not enable one of ordinary skill in the art at the time of the invention was made to make or use the claimed invention. The parties have not yet taken claim construction positions, and the facts pertaining to indefiniteness, written description, and/or enablement are being further investigated, and Apple expects that discovery taken in this litigation will further reveal facts concerning this defense. Accordingly, Apple reserves the right to supplement or modify these Invalidity Contentions with respect to 35 U.S.C. § 112 as claim construction and further discovery occurs.

Finally, Mirror Worlds states in its 3-1(e) disclosure that the earliest priority date to which all claims of the '227, '313, '427, and '999 patents are entitled is June 28, 1996, the filing date of the '227 patent application to which the '313, '427, and '999 patents claim priority. The claims recited in the '313, '427, and '999 patent, however, contain limitations that are not supported by the applications to which to which the '313, '427, and '999 claims priority. Therefore, the claims of the '313, '427, and '999 patents are not entitled to a June 28, 1996 priority date.

## I. PRIOR ART: ANTICIPATION AND OBVIOUSNESS

Pursuant to P.R. 3-3(a) and 3-3(b), and in light of the infringement contentions set forth in Mirror Worlds 3-1 contentions and accompanying claim charts, Apple identifies herein the prior art now known to Apple that anticipates and/or renders obvious the asserted claims of the patents-in-suit. In these invalidity contentions, including the appendices and exhibits, any citations to a printed publication or other reference describing a prior art system also should be construed to be a reference to the prior art system itself. Thus, for example, these contentions refer to Hypercard, the HFS file system, Lotus' Magellan, Retrospect, and On Location. The citations in these contentions are to manuals, books, or screenshots describing the functionality of those systems. Apple intends to reply on both the system (*i.e.*, a computer running the identified software) that was sold and/or in public use, and the manual or book describing the system as prior art in this case. However, all citations are to the manual, book, or screenshots describing the system.

## A. Base References For Anticipation And Obviousness

1. United States Patent No. 6,243,724 (Mander *et al.*) – Method and Apparatus for Organizing Information in a Computer System (piles) (APMW0000001-APMW0000049) ("hereinafter "the '724 patent" or "1")

2. The Lotus Magellan product, as described in, *e.g.*, Using Lotus Magellan (1989); as well as the book Using Lotus Magellan (1989) (APMW0000050-APMW0000366) and United States Patent No. 5,303,361 (APMW0018307-APMW0018326) (hereinafter "Lotus Magellan" or "2")

3. The Retrospect software product, as described in, *e.g.*, Retrospect User's Guide (1995), as well as the book Retrospect User's Guide (1995) (APMW0000367-APMW0000704) (hereinafter "Retrospect User's Guide" or "3")

4. United States Patent No. 5,499,330 (Peter Lucas, DEC) – Document Display System for Organizing and Displaying Documents as Screen Objects Organized Along Strand (APMW0000705-APMW0000732) (hereinafter "the '330 patent" or "4")

5. English translation of Japanese Patent No. 6-180661 (Yumiko *et al.*) (APMW0000733-APMW0000751) (hereinafter "the JP '661 patent" or "5")

6. United States Patent No. 5,504,852 (Thompson-Rohrlich) – Method for Creating a Collection of Aliases Representing Computer System Files (Smart Folders) (APMW0000752-APMW0000759) (hereinafter "the '852 patent" or "6")

7. United States Patent Number 5,621,906 (O'Neill *et al.*) – Perspective-Based Interface Using An Extended Masthead (APMW0000760-APMW0000769) (hereinafter "the '906 patent" or "7")

8. United States Patent No. 5,758,324 (Hartman *et al.*) – Resume Storage and Retrieval System (APMW0000770-APMW0000796) (hereinafter "the '324" or "8")

9. United States Patent No. 6,396,513 (Helfman *et al.*) – Electronic Message Sorting and Notification System (APMW0000797-APMW0000811) (hereinafter "the '513 patent" or "9")

10. SIGIR '93 – "Content Awareness in a File System Interface: Implementing the 'Pile' Metaphor for Organizing Information" by Rose, Mander, Oren, Ponceleon, Salomon & Wong (APMW0000812-APMW0000821) (hereinafter "the SIGIR '93 article" or "10")

11. United States Patent No. 5,724,567 (Rose *et al.*) – System for Directing Relevance-Ranked Data Objects to Computer Users (APMW0000822-APMW0000834) (hereinafter "the '567 patent" or "11")

12. United States Patent No. 6,202,058 (Rose *et al.*) – System for Directing Relevance-Ranked Data Objects to Computer Users (APMW0000835-APMW0000845) (hereinafter "the '058 patent" or "12")

13. "A 'Pile' Metaphor for Supporting Casual Organization of Information," by Mander, Salomon and Wong (CHI '92) (APMW0000846- APMW0000862) (hereinafter "the CHI '92 article" or "13")

14. United States Patent No. 5,649,188 (Nomura *et al.*) – Electronic Filing Apparatus Which Allows Information to be Retrieved Based on a Box, a Date, or a Card Associated with the Information (APMW0000863-APMW0000978) ) (hereinafter "the '188 patent" or "14")

15. The HyperCard Basics (Apply Computer, 1990) and HyperCard Stack Design Guidelines (Addison-Wesley, 1989) (APMW0000979- APMW0001019)

16. United States Patent No. 6,00,227 (Freeman *et al.*) – Document Stream Operating System (APMW0014222 - APMW0014237) (hereinafter "the '227 patent" or "16")

17. TR-1070 – "The 'Lifestreams' Approach to Reorganizing the Information World," YALEU/DCS/TR-1070 (1995) (YALE000430 - YALE000441) (hereinafter "TR-1070" or "17")

18. "Semantic File Systems," by Gifford, Jouvelot, Sheldon and O'Toole (ACM'91) (APMW0018268 - APMW0018277) (hereinafter "the SFS article" or "18")

19. On Location 2.0.1, by ON Technology, Inc. (1990-91) (APMW0018278 - APMW0018306) (hereinafter "On Location" or "19")

Pursuant to P.R. 3-3(c), the claim charts attached hereto as Exhibits 1 to 19 identify specifically where, in each of these base references, each element of each asserted claim is found. In addition, the table in Appendix A states for each reference and each claim whether Apple contends the reference anticipates the claim or renders it obvious.

**B. Obviousness Combinations**

In general, the problem purportedly addressed by the Mirror Worlds patents was already well known to those of skill in the art by 1996. The '227 patent describes disadvantages of conventional operating systems that it seeks to address. These are "(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." '227 patent at 1:40-45; *see also* Deposition of N. Carriero at 55:5-15 ("Q. Sitting here today, give me your best recollection as to what his [Dr. Gelernter's] Lifestreams idea was. A. Ah. Okay. At that

point, I think what he was trying to get at was a system that organized data through a default attribute of chronology and avoided extraneous compartmentalization and differentiation that's induced by deciding what to name something, what file to put it in, what folder, you know, it goes in, that kind of thing. So he was trying to distinguish -- create an environment that went away from that?"), 58:8-14 ("Q. Would it be fair to say that you recall, sitting here today, that at least one important part of the Lifestreams idea that Dr. Gelernter came up with was organization of information based on time as opposed to the name and folder hierarchy that existed at the time? A. That would be one -- yes, one of the aspects of it."). The '227 patent further explains that "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." *Id,* at 1:52-58. The '227 patent then goes on to explain that the "document stream operating system and method" it describes is intended to "solve many, if not all" of these disadvantages. *Id.* at 2:13-16.

However, these problems were already well known by 1996. For example, the '361 patent (Lotus Magellan) explains at 1:20-51 that "the storage capacity and access speed of today's hard disk drives is increasing rapidly. At the same time the price of hard disk drives is decreasing rapidly. As a result there is a proliferation of hard disk drives installed in PCs and users of varying levels of expertise are storing more and more data on the drives. Many users, however, encounter difficulties in searching and retrieving the data they have stored. For example, users sometimes cannot remember the name of the file that contains the data they seek or even where the file is located within a maze of directories and subdirectories of files. Further, users who store vast amounts of data in files created with a growing diversity of software applications, *e.g.*, spreadsheets, personal information managers, word processors, database managers, and electronic mail exchanges, often find that they cannot consolidate the data." To

address this problem, the ‘361 patent describes indexing and searching capabilities to allow convenient and effective searching and retrieving.

Similarly, in the 1994 Washington Post article “The Cyber-Road Not Taken,” the author explains that he does not want to organize his information into files, or to name those files. Instead, the author wants a “lifestream,” which “captures your whole life, in terms of chunks of information,” and can be searched/filtered in order to only display certain kinds of documents. The same is true of TR-1070, which explains that “names and directories should be junked as organizing devices,” and that instead, computers “should provide sophisticated logic for finding” information.

In yet another example, the SFS article also describes these problems with the traditional hierarchical or “tree structured” file system, and proposes as an alternative a “semantic file system,” wherein files are located by searching an index of their contents or attributes, instead of by their file name and location. The SFS article describes an implemented semantic file system. Through this file system, users seeking files obtain them by entering search criteria and browsing through “virtual directories” containing the results of those searches, rather than browsing through traditional static directories.

As this shows, the problem being addressed by the Mirror Worlds patents was well known in the art prior to 1996. Moreover, as discussed in more detail below, each of the elements of the various solutions proposed by Mirror Worlds to this problem was also known as a solution to these problems.

Apple contends that each of the base references identified in Section I.A above renders each of Mirror Worlds’ claims obvious on the basis of the disclosure in the claim charts attached hereto as Exhibits 1 to 19 in combination with the knowledge of a person of ordinary skill in the art. The charts attached hereto as Exhibits 1 to 19 identify specifically where, in each of these base references, each element of each asserted claim is found. Apple further contends each of Mirror Worlds’ claims is rendered obvious by the additional references identified in the “Additional Obviousness References” list below (Section I.C). The references in Section I.C

show knowledge in the relevant art during the relevant time period. Apple has provided a description, but not claim charts, for the references listed in Section I.C because those references being used as supporting references in an obviousness combination to show the state of the art at the relevant time.

Apple further contends that each of Mirror Worlds' claims is obvious on the basis of the combinations of references set forth in the following table, as well as for the reasons set forth in the numbered paragraphs below, which provide explanation of the basis for the obviousness combinations shown in the table below.

| **Base Reference** | **In Combination With** |
|---|---|
| 1 ('724 patent) | 2-7, 10, 13, 15, 17-20, 22, 26, 27, 31, 32, and/or 35; and/or for the '999 patent, 16. |
| 2 (Lotus Magellan) | 1, 3-7, 10, 13, 15, 17-20, 23-26, 31, 32, and/or 35; and/or for the '999 patent, 16. |
| 3 (Retrospect) | 1, 4, 5, 6, 7, 9, 10, 13, 15, 17-20, 26, 31, 32 and/or 35; and/or for the '999 patent, 16. |
| 4 ('330 patent) | 1-3, 6, 9, 10-13, 15, 17-20, 23, 26, 31, and/or 35; and/or for the '999 patent, 16. |
| 5 (JP '661 patent) | 1-3, 6-7, 9-13, 15, 17-20, 23, 26, 31, 32 and/or 35; and/or for the '999 patent, 16. |
| 6 ('852 patent) | 1, 3-5, 7, 10-13, 15, 17-20, 23, 26, 31, 32 and/or 35; and/or for the '999 patent, 16. |
| 7 ('906 patent) | 1-3, 6, 8-13, 15, 17-20, 23, and/or 35; and/or for the '999 patent, 16. |
| 8 ('324 patent) | Will not be used as a base reference |
| 9 ('513 patent) | 1, 3-5, 7, 10, 13, 15, 17-20, 23, 32, and/or 35; and/or for the '999 patent, 16. |
| 10 (SIGIR '93) | 1-7, 13, 15, 17-20, 22, 26, 27, 31, 32, and/or 35; and/or for the '999 patent, 16. |
| 11 ('567 patent) | 1, 3-7, 10, 13, 15, 17-20, 32, and/or 35; and/or for the '999 patent, 16. |
| 12 ('058 patent) | 1, 3-7, 10, 13, 15, 17-20, 32, and/or 35; and/or for the '999 patent, 16. |
| 13 (CHI '92) | 1-7, 10, 15, 22, 26, 31, 32, and/or 35; and/or for the '999 patent, 16. |
| 14 ('188 patent) | 1-7, 10, 13, 17-20, 32, and/or 35; and/or for the '999 patent, 16. |
| 15 (Hypercard) | 1-7, 10, 13, 17-20, 23, 26, 31, 32 , and/or 35; and/or for the '999 patent, 16. |
| 16 (227 patent) | (for the '999 patent): 1, 2, 4, 17, 18, 20, and/or 26. |

| | |
|---|---|
| 17 (TR-1070) | 1-7, 9-15, 17-20, 23, 25-27, 31-32, and/or 35; and/or for the '999 patent, 16. |
| 18 (SFS article) | 1-7, 10-13, 15, 17, 19, 20, 22, 23, 30-32, and/or 35; and/or for the '999 patent, 16. |
| 19 (On Location) | 1-7, 10-13, 15, 17, 18, 20, 22, 23, 31, 32, and/or 35; and/or for the '999 patent, 16. |

### 1.0 Organizing Each Data Unit Into A Main Stream

Many of the Mirror Worlds' asserted claims—'227:1-6 and 9-12; '227:13-17, 20, and 22; '227:25-29; '313:1-4; '427:1-2, 5, and 7; '427:8-10, 13, and 15; '427:16-19, 22, and 24; '427:25-26, 29, and 31; '999:1—require "a computer system which organizes each data unit received or generated by the computer system" into "a main stream" of data units, or something similar, such as a "document stream organizing system," "document organizing facility," or "time-ordered stream." The concept of organizing each data unit (*i.e.*, document) received or generated by the computer system into a main stream of data units was well known, commonly used, and routine to those of skill in the art before 1996.

For example, the HFS file system used on Macintosh computers organized every file received by the computer on a volume-by-volume basis. *See, e.g.,* Inside Macintosh: Files at 2-53. There were many other file systems available at the time, such as FAT and NTFS, that also did this. Another example is Lotus' Magellan, which "serves as an information pilot" that "creates an index of every word in every file on your disk, so finding information is a simple and fast process." *See, e.g.*, Using Lotus Magellan at 1-2. Another example is the '724 patent, which describes indexing "every document" in a computer system to allow searching and automated sorting and organization of the documents in the computer system. *See, e.g.*, '724 patent at 24:8-26:19. Another example is the '852 patent, which "provides a secondary and parallel organization of files stored on a computer system" that allows searching of those files, for example to identify all files "having 'progress report' in their names" or that were "modified

today." *See, e.g.,* '852 patent at 2:1-10; 2:54-68; 4:44-68. Another example is the Retrospect software, which archives files and generates an index of the files on the archival media. *See, e.g.,* Retrospect User's Guide at 21. Another example is On Location, which indexes the data on a user's computer to allow the user to search for it later. A further example is the SFS article, which describes personal computer indexing systems and information retrieval systems in combination with distributed file systems and describes the use of attributes / metadata in lieu of typical hierarchical folder structures and naming of files in order to organize all the documents in a system into a "semantic file system." Another example is TR-1070, which describes a "lifestream" or "a stream of information chunks, typically intended to include every information chunk of interest to its owner." TR-1070 at 2. A further example is the Cyber-Road Not Taken article.

The result of organizing each data unit received or generated by a computer system into a main stream of data units was predictable to those of skill in the art before 1996. Each data unit (*i.e.,* file) would be organized, allowing more efficient operations, such as searching or sorting, on those data units.

If it is determined that any of the base references listed above do not disclose organizing each data unit received or generated by the computer system into a main stream of data units, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the HFS file system, Lotus' Magellan, the '724 patent, the '852 patent, On Location, SFS article, TR 1080, Cyber-Road Not Taken article, or the Retrospect references. Design incentives would have prompted such modification, because it was well known to be desirable to perform efficient searching and sorting of the files on a computer. Market forces would also have prompted such modification, for example because it was known before 1996 that "the average manager spends almost four 40-hour work weeks each year looking for information that is misplaced or mislabeled." *See, e.g.,* Using Lotus Magellan at 1. In the combination, each of the original

elements of the base reference would be performing the same known and predictable functions described in the base reference, and the organizing each data unit received or generated by the computer system into a main stream of data units would perform the same known and predictable functions described in the HFS file system, Lotus' Magellan, the '724 patent, the '852 patent, On Location, SFS article, TR 1080, Cyber-Road Not Taken article, or the Retrospect software references. The results of such combination would also be predictable to a person of skill in the art before 1996. With each data unit organized into a main stream, more efficient searches and sorting of the data units would be enabled.

### 1.1 Organizing Data Units Generated In The Local Computer System

Many of Mirror Worlds' asserted claims—'227:1-6 and 9-12; '227:13-17, 20, and 22; '227:25-29; '313:1-4; '313:9-11; '427:1-2, 5, and 7; '427:8-10, 13, and 15; '427:16-19, 22, and 24; '427:25-26, 29, and 31; '427:32-34, 37, and 39; and '999:1—require "generating data units in the computer system" and organizing them into the "main stream" described above, or something similar, such as "receiving documents from diverse applications." The concept of generating data units in the computer systems was well known, widely used, and routine to those of skill in the art before 1996.

Essentially all computer systems generated data. For example, a Macintosh computer using the HFS file system could create new file, such as text or image or sound files, and organize those files with the file system. Similarly, the computer systems described in the '724 patent, TR-1070, and the SFS article and used with On Location, Lotus Magellan and Retrospect generated new files that were organized by the respective references. The result of generating new files with a computer system was predictable to those of skill in the art before 1996. New data units to be organized as described in section 1.0 would be created, and would then be organized as described to allow for more efficient operations, such as searching or sorting, on those data units.

If it is determined that any of the base references listed above do not disclose organizing each data unit received or generated by the computer system into a main stream of data units, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the HFS file system, Lotus' Magellan, the '724 patent, '852 patent, SFS article, On Location, TR-1070, or the Retrospect software references. The reasons for this obviousness are the same as those described above in section 1.0.

### 1.2 Organizing Data Units Received From Other Computer Systems, Such As "Client" Computers Or The World-Wide-Web

Some of Mirror Worlds' asserted claims—'227:1-6 and 9-12; '227:13-17, 20, and 22; '227:25-29; and '999:1—require "receiving data units from other computer systems" and organizing into the "main stream" described above, or something similar. The concept of receiving data units from other computer systems and organizing them into a main stream was well known, widely used, and routine to those of skill in the art before 1996.

Computer networking was widely known, and email was in widespread use, as was the "World Wide Web." The HFS file system, for example, organized all data units received by a Macintosh computer regardless of whether they were generated by the Macintosh or received from another computer, for example through the World Wide Web. The same is true of other well-known file systems, like FAT and NTFS. Moreover, organizing data units on other computer systems to allow searching through those other systems was also known. For example, the '852 patent describes allowing users to drag "servers" into "a folder where users store aliases of places to look [i.e. search]." '852 patent at 4:44-53. Lotus Magellan "can search for and list files across directories, on separate drives, and even across local area networks." *See, e.g.,* Using Lotus Magellan at 2. The Crouse '972 patent discloses an archiving file system which operates on top of the native file system and allows for "completely transparent" storage and retrieval of files located on a remote network data server or on a distributed network. '972 patent at 4:22-42. Other references, such as the '724 patent**,** SFS article, TR-1070, and Retrospect, also provide this

disclosure. The result of receiving data units from other computer systems and organizing them into a main stream was predictable to those of skill in the art before 1996. With data unit from various different systems organized into a main stream, more efficient searches and sorting of the data units across systems would be enabled. A search would be performed and it would identify those data units that satisfied the search criteria.

If it is determined that any of the base references listed above do not disclose receiving data units from other computer systems and organizing them into a main stream, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the HFS file system, Lotus' Magellan, the '724 patent, '852 patent, '972 patent**,** SFS article, TR-1070, or the Retrospect software references. The reasons for this obviousness are the same as those described above in section 1.0.

### 1.3 Receiving Documents From Diverse Applications / Document Object Models

Many of the Mirror Worlds asserted claims—'227:1-6 and 9-12; '227:13-17, 20, and 22; '227:25-29; '313:1-4; '313:9-11; '427:1-2, 5, and 7; '427:8-10, 13, and 15; '427:16-19, 22, and 24; '427:25-26, 29, and 31; '427:32-34, 37, and 39; and '999:1—require "receiving documents from diverse applications in formats that are specific to the respective applications and differ as between at least some of the applications," or something similar. Some of Mirror Worlds' asserted claims, *e.g.*, '999:1, further require creating "document object models having a consistent structure" that include "selected information" from these diverse files "created by diverse software." The concept of receiving data units in diverse formats, and organizing them by creating document object models that have a consistent structure across the diverse documents was well known, widely used, and routine to those of skill in the art before 1996.

The HFS file system, for example, organized all data units received by a Macintosh computer regardless of the application that created them or the nature of the data that they contained. The same is true of other well-known file systems, like FAT and NTFS. The HFS

file system also populated its catalog file with metadata about each file (*i.e.*, a document object model) regardless of the nature of the data or the identity of the file that created it. This allowed the pbCatSearch function to search for any file, regardless of what created it or what type it was. Another example is Lotus Magellan, which "creates an index of every word in every file on your disk, so finding information is a simple and fast process," regardless of what type of file is being indexed, or what program created that file. *See, e.g.*, Using Lotus Magellan at 1-2. The same is true of the '724 patent, which describes indexing "every document" in a computer system to allow searching and automated sorting and organization of the documents in the computer system regardless of their file type. *See, e.g.,* '724 patent at 24:8-26:19. Another example is the '852 patent, which "provides a secondary and parallel organization of files stored on a computer system" that allows searching of those files, for example to identify all files "having 'progress report' in their names" or that were "modified today," regardless of the file type or what application created the file. *See, e.g.*, '852 patent at 2:1-10; 2:54-68; 4:44-68. Another example is the '330 patent, which "supports multiple renderers, and which renderer is used for a particular document is determined by an attribute of the document." '330 patent at 5:46-48; Other references, such as the '972 patent, On Location, TR-1070, the SFS article, the Cyber-Road Not Taken article, and Retrospect, also provide this disclosure. The result of receiving data units in diverse formats, and organizing them by creating document object models that have a consistent structure across the diverse documents was predictable to those of skill in the art before 1996. With data unit from various different formats and applications organized through consistent document object models (typically consolidated in an index), efficient searches and sorting of the data units would be enabled, regardless of the specific type of file or data. A search would be performed and it would predictably identify those data units that satisfied the search criteria based on the information in their document object model, regardless of the specific type or byte-content of the corresponding data unit.

If it is determined that any of the base references listed above do not disclose receiving data units from other computer systems and organizing them into a main stream, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the HFS file system, Lotus' Magellan, the '724 patent, '852 patent, the '330 patent, the '972 patent, the SFS article, On Location, TR-1070, the Cyber-Road Not Taken article, or the Retrospect software references. The reasons for this obviousness are the same as those described above in section 1.0, with the additional point that it was both well known, and more desirable, to be able to organize each data unit into a main stream even where the data units were of different types and from different applications.

**2.0 Generate Substream(s)**

Many of Mirror Worlds' asserted claims—'227:1-6 and 9-12; '227:13-17, 20, and 22; '227:25-29; '313:2-4; '313:11—require generating "at least one substream" of data units, or something similar. The concept of searching or sorting a main stream of data units to generate a substream of data units was well known, commonly used, and routine to those of skill in the art before 1996.

In general, computers have had searching and sorting capabilities for decades. The ability to search or sort through a large and growing stream of data units is one of the natural uses of a computer. Databases were widely known and used long before 1996, and had the ability to search or sort large streams of data units. *See generally,* C. J. Date, An Introduction to Database Systems, 3rd ed. (Addison-Wesley, 1981). There are many other examples of searching a stream of data units, such as the HFS file system used on Macintosh computers included a catalog file containing metadata about each file, that could be searched using the function "pbCatSearch" in order to find files based on their metadata. *See, e.g.,* Inside Macintosh: Files at 2-53. Another example is Lotus' Magellan, which "serves as an information pilot" that "creates an index of every word in every file on your disk, so finding information is a simple and fast

process." *See, e.g.*, Using Lotus Magellan at 1-2. Another example is the '724 patent, which describes indexing "every document" in a computer system to allow searching and automated sorting and organization of the documents in the computer system. *See, e.g.*, '724 patent at 24:8-26:19. Another example is the '852 patent, which identifies files on a computer system that meet "defined search criteria" and presents aliases to those files to the user in a folder representing the search results. Another example is the '330 patent, which describes how substreams are generated from a mainstream. *See, e.g.*, '330 patent at 9:65-10:7; 13:65-14:36. Another example is Retrospect, which describes creating a "catalog" of all files from which the user can find files according to selection criteria. *See, e.g.,* Retrospect User's Guide at 58-68. The result of searching or sorting a main stream of data units to generate a substream of data units was predictable to those of skill in the art before 1996. A search would be performed and it would identify those data units that satisfied the search criteria. In addition, the SFS article, On Location, and TR-1070 all disclose the ability to search or sort through a large and growing stream of data units.

Thus, if it is determined that any of the base references listed above do not disclose searching or sorting a main stream of data units to generate a substream of data units, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the HFS file system, Lotus' Magellan, the '724 patent, the SFS article, On Location, TR-1070, or the '852 patent references. Design incentives would have prompted such modification, because it was well known to be desirable to perform efficient searching and sorting of the files on a computer, particularly as an alternative to traditional hierarchy and name-based file organization techiniques. *See, e.g.*, SFS article, TR-1070. Market forces would also have prompted such modification, for example because it was known before 1996 that "the average manager spends almost four 40-hour work weeks each year looking for information that is misplaced or mislabeled." *See, e.g.,* Using Lotus Magellan at 1. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions

described in the base reference, and the searching or sorting a main stream of data units to generate a substream of data units would perform the same known and predictable functions described in the HFS file system, Lotus' Magellan, the '724 patent, the SFS article, On Location, TR-1070, or the '852 patent references. The results of such combination would also be predictable to a person of skill in the art before 1996. With the ability to generate substreams, the results of searches for data units satisfying the search criteria would be enabled, allowing users to identify the subset of data units from the mainstream that satisfy search criteria.

**2.1 Generating Persistent Substream(s), Including Generating "Live" Substreams**

Some of Mirror Worlds' asserted claims—'227:1-6 and 9-12; '227:13-17, 20, and 22; '227:25-29; '313:3-4—require generating "maintaining the main stream and the substreams as persistent streams" of data units, or something similar. The concept of maintaining a stream of data as a "live" and "persistent" stream, *i.e.*, one that persists and is regularly updated until it is removed by a user, was well known, commonly used, and routine to those of skill in the art before 1996.

For example, the '724 patent, which describes indexing "every document" in a computer system to allow searching and automated sorting and organization of the documents in the computer system, and then creating "scripts" for "piles" of documents which automatically collect documents that meet specific criteria into a particular pile. *See, e.g.*, '724 patent at 24:8-26:19; Fig. 14; 21:66-22:61. These criteria-based piles are persistent and can be set to automatically sort "any new or modified document" in the system. *Id.* at 28:3-12. Another example is Lotus' Magellan, which allowed a user to search the computer system and save the search for future uses. *See, e.g.*, Using Lotus Magellan at 1-2. Another example is the '852 patent, which identifies files on a computer system that meet "defined search criteria" and presents aliases to those files to the user in a folder representing the search results. These folders persist and are automatically updated when the system has available resources. Another example

is the ’330 patent, which describes how substreams are generated from a mainstream. *See, e.g.*, ’330 patent at 9:65-10:7; 13:65-14:36. Another example is Retrospect, which describes creating a “catalog” of all files from which the user can find files according to selection criteria, and using scripts to automate backup or archiving of the files that satisfy the criteria. *See, e.g.*, Retrospect User’s Guide at 58-68. The result of adding persistence to a search or sort of a stream of data units was predictable to those of skill in the art before 1996. The search criteria would be saved, and the results of the search would be periodically updated as new data units were added, and existing data unites were deleted or modified. In addition, the SFS article, On Location, and TR-1070 all disclose the ability to search or sort through a “live” stream of data units.

If it is determined that any of the base references listed above do not disclose maintaining a stream of data as a “live” and/or “persistent” stream, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of Lotus’ Magellan, the ’724 patent, ’852 patent, the ’330 patent, the ’972 patent, the SFS article, On Location, TR-1070, or the Retrospect references. The reasons for this obviousness are the same as those described above in Section 2.0, with the additional point that it was predictable, well known, and desirable, to be able to save searches and or automate the process of updating them, in order to save users time and effort and produce a more efficient computer system.

### 2.2 Generating Substream(s) From Other Substreams

Some of Mirror Worlds’ asserted claims—’227:10 and ’227:11—require generating “substreams from existing substreams” of data units, or something similar such as operating on a substream “using a set of operations selected by the user.” The concept of searching or sorting a stream of data units to generate a substream of data units was well known, commonly used, and routine to those of skill in the art before 1996, as described above in Section 2.0. In addition, the concept of performing operations such as searching or sorting on a substream of data units, in

order to generate a substream from an existing substream, was well known, commonly used, and routine to those of skill in the art before 1996.

In general, generating a substream from an existing substream simply involves combining two sets of search criteria, and as such is simply a more specific or narrow search, and is obvious for all the same reasons described above in Section 2.0. Moreover, the references described herein show that this was well known. For example, the ’330 patent describes how to generate “substrands” of documents and further explains how documents received in a particular time period can be group together. *See, e.g.*, ’330 patent at 13:65-14:36; 14:29-336 (“ . . . the user requests that all mail messages received after a specified date be grouped in the foreground, and all others in the background.”) Another example is the ’724 patent, which describes multiple different methods of creating subpiles from existing piles of documents using search criteria. *See, e.g.*, ’724 patent at 29:37-33:24. Similarly, Retrospect allows the use of stacked search criterion, or “selectors.” The same is true of the ’852 patent, which explains that a “viewer can be placed inside a viewer, knowing that the search domain of the ‘inner’ viewer is already restricted to files found by the ‘outer’ viewer.” *See e.g.* ’852 patent at 4:37-42. In addition, the SFS article, On Location, and TR-1070 all disclose the ability to search or sort through a “substream” in order to generate what is, in effect, a sub-substream.

If it is determined that any of the base references listed above do not disclose generating “substreams from existing substreams” of data units, or operating on a substream “using a set of operations selected by the user,” it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of Lotus’ Magellan, the ’724 patent, the ’852 patent, the ’330 patent, the SFS article, On Location, TR-1070, or the Retrospect references. The reasons for this obviousness are the same as those described above in Section 2.0, with the additional point that it was predictable, well known, and desirable, to be able to perform nested searches and in order to allow generation of more sophisticated queries and produce a more efficient computer system.

### 3.0 A Timestamp To Identify Each Data Unit

Many of Mirror Worlds' asserted claims—'227:1-6 and 9-12; '227:13-17, 20, and 22; '227:25-29; '313:1-4; '313:9-11; '427:1-2, 5, and 7; '427:8-10, 13, and 15; '427:16-19, 22, and 24; '427:25-26, 29, and 31; '427:32-34, 37, and 39—require using a "timestamp" to "identify each data unit," or something similar, such as "automatically associating time-based indicators with the documents received." The concept of associating a timestamp with a data unit to identify it, such as for the purpose of identifying documents edited in the "last week," was well known, commonly used, and routine to those of skill in the art before 1996.

For example, the HFS file system used on Macintosh computers uses a file's creation date, modification date, and/or backup date as types of criteria that it can use to identify data units. *See, e.g.*, Inside Macintosh: Files at 2-38. Other file systems available at the time, such as FAT and NTFS, also used a file's various timestamps—such as create time and date, last access date, last modification time and date—to identify documents. Another example is Lotus' Magellan, which stores as part of its index of each file the file's date and time so that a document may be found during a search. *See, e.g.*, Using Lotus Magellan at *xii*, and 13. Another example is the '724 patent, which describes indexing documents in the computer system so that searching and sorting may be done, including assigning identifying colors to documents based on their dates. *See, e.g.*, '724 patent at 20:14-43, Figs. 13a, 13b, 14. Another example is Retrospect, which describes indexing remotely stored documents so that storage, search and retrieval may be done according to timestamps. *See, e.g.*, Retrospect User's Guide at 151-157. Another example is the '852 patent, which stores date and time information for files to enables searching for documents based on date. *See, e.g.*, '852 patent at Fig. 4, Fig. 2; 1:55-2:11 (describing searches such as "files modified today" and "files not accessed in the past 12 months."). Similarly, the '330 patent describes how to generate "substrands" of documents that group together documents received in a particular time period. *See, e.g.*, '330 patent at 13:65-14:36; 14:29-36 (" . . . the user requests that all mail messages received after a specified date be grouped in the foreground, and all others in the background.") Another example is On Location, which indexes each file,

including the file's date and time, so that a document may be found and categorized based on date and/or time. Similarly, the SFS article describes computer indexing systems including Magellan and On Location that store date information, as well as information retrieval systems that store and categorize documents using date information. A further example is TR-1070, which describes a "lifestream" or "a stream of information chunks, typically intended to include every information chunk of interest to its owner" and explains that "chunks are stored in the stream chronologically by the date and time at which they were created." The result of associating a timestamp with a date unit to identify it was predictable to those of skill in the art before 1996. The various timestamps associated with a file make searching and sorting files more efficient.

If it is determined that any of the base references listed above to not disclose using a timestamp to identify each data unit, it would have been obvious to a person of skill in the art by 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the HFS file system, Lotus' Magellan, On Location, the SFS article, TR-1070, the '852 patent, the '330 patent or the '724 patent. Design incentives would have prompted such modification, because it was well known that efficient searching and sorting of documents included the ability to find and sort documents by date and time, and well known that timestamps could be used to provide this functionality. Market forces would also have prompted such modification, for example because it was known before 1996 that "the average manager spends almost four 40-hour work weeks each year looking for information that is misplaced or mislabeled," and it was known that time-based searches are often an efficient way to find misplaced information. *See, e.g.*, Using Lotus Magellan at 1. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions described in the base reference, and using a timestamp to identify each data unit would perform the same known and predictable functions described in the HFS file system, Lotus' Magellan, Retrospect, On Location, the SFS article, TR-1070, the '852 patent, the '330 patent,

and the ’724 patent. Additionally, the results of such combination would also be predictable to a person of skill in the art before 1996. By associating each data unit with a timestamp to identify it for purposes of searching and sorting, useful searches and sorts would be enabled, such as date-based searches.

### 4.0 A Chronological Indicator Having The Respective Timestamp

Some of Mirror Worlds’ asserted claims—’227:1-6 and 9-12; ’227:13-17, 20, and 22; ’227:25-29; ’313:1-4; ’313:9-11; ’427:1-2, 5, and 7; ’427:8-10, 13, and 15; ’427:16-19, 22, and 24; ’427:25-26, 29, and 31; ’427:32-34, 37, and 39—require “associating each data unit with at least one chronological indicator having the respective timestamp,” or something similar, such as “automatically associating time-based indicators with the documents received.” The concept of associating each data unit with at least one chronological indicator having the respective timestamp was well known, commonly used, and routine to those of skill in the art before 1996.

For example, the ’852 patent describes how folders can be created by searching documents using metadata criteria such as the “Last Modified” date. *See, e.g.,* the ’852 patent at 3:8-15. Another example is the SIGIR ’93 article, which describes how documents can be organized and searched through use of an internal representation of metadata attributes that depend on, among other things, the documents’ timestamps. *See, e.g.,* SIGIR ’93 article at 261-262. Another example is Retrospect, document archival software which creates a catalog listing of metadata attributes containing, among other attributes, a document’s timestamp information. *See, e.g.,* Retrospect User’s Guide at 21. Another example is the ’330 patent, which describes how documents are organized and displayed according to time-based categories such as old documents that have been read, and new documents that have not been read. *See, e.g.,* ’330 patent at 13:65-14:36; 14:29-336 (“ . . . the user requests that all mail messages received after a specified date be grouped in the foreground, and all others in the background.”) Another example is the ’724 patent, which describes indexing documents in the computer system so that searching, sorting, and categorizing may be done, such as by assigning identifying colors to

documents based on their dates. *See, e.g.,* '724 patent at 20:14-43, Figs. 13a, 13b, 14. Another example is Lotus Magellan, which stores, as part of its index of each file, the file's date and time so that a document may be found and categorized based on date and/or time. *See, e.g.*, Using Lotus Magellan at *xii*, and 13. Another example is On Location, which indexes each file, including the file's date and time, so that a document may be found and categorized based on date and/or time. Similarly, the SFS article describes computer indexing systems including Magellan and On Location that store date information, as well as information retrieval systems that store and categorize documents using date information. A further example is TR-1070, which describes a "lifestream" or "a stream of information chunks, typically intended to include every information chunk of interest to its owner" and explains that "chunks are stored in the stream chronologically by the date and time at which they were created." The result of associating each data unit with at least one chronological indicator having the respective timestamp was predictable to those of skill in the art before 1996. Being able to organize and search documents according to their metadata attributes improves the efficiency of searching and storing documents.

If it is determined that any of the base references listed above do not disclose associating each data unit with at least one chronological indicator having the respective timestamp, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the '852 patent, the SIGIR' 93 article, Lotus Magellan, the '724 patent, the '330 patent, On Location, the SFS article, TR-1070, and Retrospect. The use of chronological indicators in combination with a file system was a well-known method yielding predictable results because the use of metadata as a means to search and sort documents improves the efficiency of document organization and because use of chronologically-based categories was a well known form of document organization, *e.g.*, "email from last week" or "documents edited yesterday." In each of the combinations, each of the original elements of the base references would be performing the same

known and predictable functions described in the base reference, and associating each data unit with at least one chronological indicator having the respective timestamp would perform the same known and predictable functions described in the ’852 patent, the SIGIR ’93 article, Magellan, the ’330 patent, On Location, the SFS article, TR-1070, and Retrospect. The results of such combinations would also be predictable to a person of skill in the art before 1996. With the ability to associate each data unit with at least one chronological indicator having the respective timestamp, users can efficiently organize and search their documents.

### 4.1 Include Each Data Unit in the Mainstream According to Timestamp in Chronological Indicator

Some of Mirror Worlds’ asserted claims—’227:1-6 and 9-12; ’227:13-17, 20, and 22; ’227:25-29;—require “including each data unit according to the timestamp in the respective chronological indicator in the main stream,” or something similar. As explained above in Section 4.0, the concept of associating each data unit with at least one chronological indicator having the respective timestamp was well known, commonly used, and routine to those of skill in the art before 1996. The concept of a main stream was also well known, as explained above in Section 1.0. In addition, the concept of including the data units in the main stream according to its timestamp and chronological indicator was well known, commonly used, and routine to those of skill in the art before 1996. Particularly, as described above in Section 4.0, each of the ’852 patent, the SIGIR ’93 article, Retrospect, Magellan, On Location, the SFS article, the ‘330 patent, and the ‘724 patent describe organizing and/or categorizing streams of documents based on timestamps. Another example is TR-1070, which describes a “lifestream” or “a stream of information chunks, typically intended to include every information chunk of interest to its owner” and explains that “chunks are stored in the stream chronologically by the date and time at which they were created.”

If it is determined that any of the base references listed above do not disclose associating each data unit with at least one chronological indicator having the respective timestamp, it would

have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the ’852 patent, the SIGIR’ 93 article, Lotus Magellan, the ’724 patent, the ’330 patent, On Location, the SFS article, TR-1070, and Retrospect. The reasons for this obviousness are the same as those described above in Sections 3.0 and 4.0.

### 5.0 Archiving

Some of Mirror Worlds’ asserted claims—’227:9; ’227:22; ’313:1-4; ’313:9-11; ’427:1-2, 5, and 7; ’427:8-10, 13, and 15—require “archiving data units,” or something similar. The concept of archiving data units was well known, commonly used, and routine to those of skill in the art before 1996.

For example, Retrospect software that ran on Macintosh computers was a document archival program. *See, e.g.*, Retrospect User’s Guide at *v*. There were many other backup and archiving systems available of the time, such as the classic UNIX file archiver *tar*, Stuffit, and WinZip. Also, the ’972 patent discloses an archiving file system “specifically designed to support the storage of, and access, to remote files stored on high speed, large capacity network data servers.” *See*, ’972 patent at Abstract. Another example is TR-1070, which describes the desirability of automatic computer-controlled archival. Another example is Lotus’ Magellan, which describes copying files and entire disk structures to remote locations for backup storage. *See, e.g.*, Using Lotus Magellan at 88-89. Another example is the Rahm reference, which describes the expansive state of the art in data sharing, but focuses primarily on how to retrieve data archived from shared resources. *See, e.g.*, Rahm reference at 368. The result of using archiving software and systems was predictable to those of skill in the art before 1996. Data units (*e.g.*, documents) would be stored remotely, saving local disk space and allowing for retrieval of documents in the event of local system crashes. “Archiving allows you to remove seldom-used files from a hard disk without permanently getting rid of them.” Retrospect User’s Guide at 104.

If it is determined that any of the base references listed above do not disclose archiving data units, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the Retrospect, Magellan, the ’972 patent, TR-1070, or Rahm references. The use of archiving software in combination with a file system was a well-known method yielding predictable results because archival is the purpose of the archiving software, and the purposes, techniques and results of archiving were well known. Design incentives would have prompted such modification, because it was well known that being able archive documents for later retrieval was desirable such as for backup/restore purposes, and also because it was well known that it was desirable to free expensive and scarce disk or memory space by archiving older files. In the combination, each of the original elements of the base references would be performing the same known and predictable functions described in the base reference, and the archiving feature would perform the same known and predictable functions described in Retrospect, the ’972 patent, and the Rahm reference. The results of such combinations would also be predictable to a person of skill in the art before 1996. With the ability to archive documents, users can keep seldom-used files at the same time as local hard drive or memory space is freed up; furthermore, reliability is increased because archived documents can be retrieved in the event of a crash or failure.

### 5.1 Automatically Archiving Data Units

Some of Mirror Worlds’ asserted claims—’227:9; ’313:1-4; ’313:9-11; ’427:1-2, 5, and 7; ’427:8-10, 13, and 15—require “automatically archiving the received documents,” and sometimes also archiving the documents’ time-based indicators. The concept of automatically archiving documents was well known, widely used, and routine to those of skill in the art before 1996. For example, Retrospect software uses a menu-driven user interface to create scripts which automatically archive documents. *See, e.g.*, Retrospect User’s Guide at Chapter 14. Retrospect can also be further automated by integrating its own scripting with AppleScript. *See,*

*e.g.*, Retrospect User's Guide at Chapter 26. . Another example is TR-1070, which describes the desirability of automatic computer-controlled archival. A further example is the '972 patent. The result of automated archiving was predictable to those of skill in the art before 1996. By automating the archival process, the user could ensure consistency, and avoid having to have a person perform a routine task. *See,* Retrospect User's Guide at 80.

If it is determined that any of the base references listed above do not disclose automatically archiving data units, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or TR-1070, the '972 patent, or the Retrospect reference. The reasons for this obviousness are the same as those described above in section 5.0, with the additional point that automating a routine task such as running a backup or archive process was commonly used and predictable, was well known to be efficient, desirable, and cost-effective for businesses and users, and produced predictable results.

### 5.2 Archiving Data Units with Timestamps Older Than A Specified Time Point

Some of Mirror Worlds' asserted claims—'227:9 and '227:22—require archiving data units with timestamps older than a specified time point. The concept of archiving data units with timestamps older than a specified time point was well known, widely used, and routine to those of skill in the art before 1996. For example, Retrospect allows the user to search and select which files to archive, including using the document creation or modification date as conditions. *See*, Retrospect User's Guide Chapter 23. The result of archiving data units with timestamps older than a specified time point was predictable to those of skill in the art before 1996. By archiving older data units, the user could keep seldom-used files while freeing up local hard drive space.

If it is determined that any of the base references listed above do not disclose archiving data units with timestamps older than a specified time point, it would have been obvious to a

person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or the Retrospect reference. The reasons for this obviousness are the same as those described above in section 5.0, with the additional point that selectively archiving older documents was commonly used and predictable, was well known to be efficient and desirable way of moving files less likely to be needed or used, and was known to have predictable results.

### 5.3 Archiving Data Units while Retaining Chronological Indicators

Some of Mirror Worlds' asserted claims—'227:9; '227:22; '313:1-4; '313:9-11; '427:1-2, 5, and 7; '427:8-10, 13, and 15 —require archiving data units / documents "while retaining the respective chronological indicator and/or a data unit having a respective alternative version of the content of the archived data unit," or "archiving the documents and indicators in consistent format for selective retrieval," or something similar. The concept of archiving data units while retaining the indicator it was associated with was well known, widely used, and routine to those of skill in the art before 1996. For example, Retrospect and Magellan both archive the timestamp of a file along with a file itself. The same is true of the system describe in TR-1070. Moreover, Retrospect creates a catalog or index of the archived data units which allows the user to view the contents of the archived data units without accessing the archive. *See,* Retrospect User's Guide at 21. The catalog includes each data unit's metadata attributes, including timestamp information. *See,* Retrospect User's Guide at 22. This enables various functionality, including the ability to perform differential updating of archives (*i.e.*, where only new files and files that have been modified since the last archival are added to an archive).

If it is determined that any of the base references listed above do not disclose archiving data units while retaining chronological indicators and/or a data unit having a respective alternative version of the content of the archived data unit, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or the Retrospect, TR-1070, and Magellan

references. The reasons for this obviousness are the same as those described above in section 5.0, with the additional point that retaining the timestamp and/or chronological indicator associated with a file when that file was archived was commonly used and predictable, was well known to be efficient and desirable for example in order to enable differential updating of an archive, and was known to have predictable results.

## 6.0 Using Subsystems From Another Operating System, Including For Writing Documents, Interrupt Handling, And Input/Output

Many of Mirror Worlds' asserted claims—'313:1-4; '427:1-2, 5, and 7; '427:8-10, 13, and 15; '427:16-19, 22, and 24; '427:25-26, 29, and 31—require using "subsystems from at least one other operating system," and in some of these claims the "other operating system" must be used "for operations including writing documents to storage media, interrupt handling, and input/output." The concept of using the subsystems from another operating system, such as the underlying operating system on a computer, or from another networked computer, was well known, commonly used, and routine to those of skill in the art before 1996.

For example, Lotus' Magellan, "serves as an information pilot" that "creates an index of every word in every file on your disk, so finding information is a simple and fast process," and runs on top of the existing DOS operating system. It uses the functionality of DOS for storing data, interrupt handling, and input/output, as well as to access and index files located on other computers in a local area network. *See, e.g.*, Using Lotus Magellan at 1-2, 15-17 (describes installing Magellan onto a computer running DOS); 85 ("Magellan tried to remove the already existing file . . . because this filed is actually a directory, and DOS doesn't easily allow the removal of directories, an error resulted . . . "). Another example is the '724 patent, which describes indexing "every document" in a computer system to allow searching and automated sorting and organization of the documents in the computer system. The system of the '724 patent is designed to run on top of an existing Macintosh operating system, which controls the actual storing of data, interrupt handling, and input/output. *See, e.g.* '724 patent at 5:32-33 ("A system and method for organizing information stored in a file system of a computer system");

6:27-30 ("in a preferred embodiment of the present invention, the file system operates on a Macintosh computer"); 5:42-6:23 (describing input/output components of underlying computer system). Another example is the Retrospect software, which runs on top of the Macintosh operating system (or other operating systems). *See, e.g.,* Retrospect User's Guide at 3 ("Retrospect requires System 7.0 or later"). A further example is the '852 patent, which provides methods for a computer system to identify files that meet "defined search criteria" and presents aliases to those files to the user in a folder representing the search results. This functionality is performed based on the underlying functionality of a computer operating system—the Macintosh OS in the illustrated embodiment—which handles the underlying functions of storing data, interrupt handling, and input/output. *See, e.g.,* '852 patent at 5:11-6:63 ("the following notes have been determined from an implementation of Viewers for the Apple Macintosh family of computers. The Viewers were build as a System 8 Finder extension"). A further example is the '330 patent, which provides access to data stored in networked data repositories. A further example is TR-1070, which describes building "viewports" on top of other operating systems in order to provide access to the user's "lifestream," and explains that while "so far" only a Unix-based viewport has been implemented, a "Macintosh-based viewport is under development." A further example is the SFS article, which describes implementing its semantic file system using an underlying Unix system. The result of using the subsystems from another operating system, such as the underlying operating system on the local computer or in a networked computer, was predictable to those of skill in the art before 1996. It allowed a programmer implementing higher-level functionality to take advantage of existing lower-level code in an operating system without having to recreate that functionality from scratch, and/or without access to the software running on the networked computer. This basic principle of building on existing software functionality is well known and widely used by those of skill in the art.

If it is determined that any of the base references listed above do not disclose using the subsystems from another operating system, such as the underlying operating system on the local

computer or in a networked computer, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the Lotus Magellan, the ’724 patent, the ’852 patent, TR-1070, the SFS article, or the Retrospect references. Design incentives would have prompted such modification, because it was well known to be desirable to take advantage of existing functionality in an operating system to avoid the cost of recreating that functionality—virtually all software is developed based on this design incentive. Market forces would also have prompted such modification, to avoid the wasted cost of recreating functionality that already existed. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions described in the base reference, and the the subsystems from the other operating system, such as the underlying operating system on the local computer or in a networked computer, would perform the same known and predictable functions known in the art as described above and in the Lotus Magellan, the ’724 patent, the ’852 patent, the ’330 patent, TR-1070, the SFS article, or the Retrospect references. The results of such combination would also be predictable to a person of skill in the art before 1996. With the ability to use the subsystems from another operating system, such as the underlying operating system on the local computer or in a networked computer, that functionality would not have to be developed from scratch.

**7.0 Receding, Foreshortened Stack**

Many of Mirror Worlds’ asserted claims—’227:15-16; ’227:25-28; ’313:1-4; ’313:9-11; ’427:1-2, 5, and 7; ’427:10; ’427:18; ’427:25-26, 29, and 31; ’427:32-34, 37, and 39—require “displaying at least some of said documents as a receding, foreshortened stack,” or something similar, such as “wherein the document representations form a visual stream having a three-dimensional effect.” The concept of displaying a set of documents as a receding foreshortened stack or pile was well known and routine to those of skill in the art before 1996.

To begin with, the concept of using a "stack" or "pile" metaphor to represent a set of documents in a user interface was well known, in part because it is an organization technique that has existed for a long time and thus provides an organizational scheme that is intuitively appealing to users. *See e.g.*, Malone, T. W., How do people organize their desks? Implications for the design of office information systems, ACM Transactions on Office Information Systems, Volume 1, Number 1, January 1983, Pages 99-112. Moreover, the use of a three dimensional representation of a stack or pile (*i.e.*, a receding foreshortened stack) to represent a set of documents was also well known. For example, the '330 patent discloses a system for "position screen objects in a three-dimensional workspace," allowing "grouping of documents, so that they can be manipulated in groups." *See, e.g.*, the '330 patent at 8:33-36, see also "Representation in Virtual Space: Visual Convention in the Graphical User Interface" by L. Staples (1993). The positioning of screen objects includes displays of a receding, foreshortened stack. *See, e.g.*, the '330 patent Figs. 3 & 5. Another example is the JP '661 patent which discloses a method for displaying files chronologically in a 3D perspective view. *See, e.g.*, the JP '661 patent, Figs. 10-14. Another example is the '906 patent, which discloses a user interface for arranging information in a perspective view. *See, e.g.*, the '906 patent at 2:29-34; Figs. 4 & 6. Another example is the '724 patent, which discloses a user interface in which document representations are stacked with a three-dimensional effect. *See, e.g.*, the '724 patent at Fig. 2. Another example is the Staples article, which explains the use of perspective more generally, and describes existing interfaces that use perspective, including the Ark Workspace software, in addition to showing receding, foreshortened stacks of partly overlapping document representations. *See, e.g.* Staples at Figs. 9-12. Another example is the Cyber-Road Not Taken article, in which the author describes the visual effect as "a caravan of shoeboxes, the most recent addition being the closest to you, receding into the far distance." Another example is TR-1070, which describes "a 3D stream receding from the present into the past," and explains that "3D is helpful because it allows us to use visual cues to communicate important information about chunks—namely, their relative ages." The result of displaying documents in a receding, foreshortened stack was

predictable to those of skill in the art before 1996. The receding, foreshortened stack to represent a stack of documents creates an intuitive user interface reflecting a user's physical desktop environment, and thereby providing useful visual cues about the organization of the documents.

If it is determined that any of the base references listed above do not disclose displaying a set of documents as a receding, foreshortened stack, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the '330 patent, the JP '661 patent, the '906 patent, or the '724 patent, the Cyber-Road Not Taken article, TR-1070, or the Staples article. Applying the known techniques of using stacks to represent sets of documents and of creating three-dimensional and perspective view effects to a known product--the computer user interface of an application or operating system--would yield the predictable result of displaying a receding, foreshortened stack of documents in that user interface. Design incentives would have prompted such modification, because it was well known to be desirable to have a more intuitive and user-friendly interface, and because it was well known that the stack or pile metaphor provided such an intuitive interface. It was also well known that computers were increasing in processing power very rapidly (*e.g.*, Moore's Law), and as a result it was entirely predictable that the processing power needed to compute and display a receding foreshortened stack of documents would be widely available. Furthermore, the teachings, suggestions, and motivations provided by T.W. Malone's article, "How Do People Organize Their Desks?" would have led one or ordinary skill in the art to modify the computer user interface to reflect how people organize their documents, and particularly, to adopt the use of the "stack" metaphor to represent a set of documents. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions described in the base reference, and the displaying documents in a receding, foreshortened stack would perform the same known and predictable functions described in the '330 patent, the JP '661 patent, the '906 patent, the Staples article, the Cyber-Road Not Taken article, TR-1070 or the '724 patent. The results of such combination would also be predictable to a person of skill in the art before 1996 – the base

reference would have a user interface that included the use of a receding foreshortened stack. User interfaces providing more intuitive reflections of a user's physical desktop will aid in efficient workflow and will increase the appeal of computer programs to users. Thus, market forces would also have prompted this combination.

**8.0 Document Representation(s) And Glance View(s)**

Many of Mirror Worlds' asserted claims—'227:6; '227:12; '227:15-16; '227:20; '227:25-29; '313:1-4; '313:9-11; '427:1-2, 5, and 7; '427:8-10, 13, and 15; '427:16-19, 22, and 24; '427:25-26, 29, and 31; '427:32-34, 37, and 39; and '999:1—require generating "document representations" of data units/documents, or something similar, such as "browse cards" or "alternative versions of the content of the data units." Many of Mirror Worlds' asserted claims—'227:25-29; '313:1-4; '313:9-11; '427:1-2, 5, and 7; '427:8-10, 13, and 15; '427:16-19, 22, and 24; '427:25-26, 29, and 31; '427:32-34, 37, and 39; and '999:1—further require generating an "information specifying glance view" that is a different "alternative version" of the content of the data unit/document that displays additional information. This "glance view" may "comprise an abbreviated version of the respective document," or something similar. The concept of representing a set of data units/documents with a set of corresponding document representations, and then further generating glance views to display additional information about the represented document, was well known and routine to those of skill in the art before 1996.

For example, the CHI '92 article shows document representations that can be stacked into "piles" in various ways. *See, e.g.*, CHI '92 article at Figs. 1, 2, 5, 7. It also discloses various techniques for displaying glance views of particular documents from within a stack, including "gesturing vertically" in order to generate a "viewing cone" that "contains a miniature version of the first page of the item under the pointer" when the pointer is over the representation of a document in the stack. *See, e.g.*, CHI '92 article at Figs. 5, 7. Another example is the '724 patent, which describes piles of document representations, and displaying a glance view after sliding a cursor over a document representation in the pile. '724 patent at Figs. 4a-f, 10a-b, 12a-

b. Another example is the JP ’661 reference, which shows stacks of document representations, and moving a cursor over a document representation in order to display additional information about the document across the bottom of the screen. *See* JP ’661, at Figs. 10, 12. A further example is TR-1070, which describes and illustrates clicking on a document representation in order to “display key attributes above.” TR-1070 at YALE000440 - YALE000441. Another example is the ’330 patent, which discloses organizing document representations (“screen objects”) into three-dimensional stacks. *See, e.g.,* ’330 patent at 3:1-3, Fig. 3. It discloses scrolling through this stack in order to bring onto the screen a larger, alternative visual representation that specifies additional information about the data unit, and it shows tiling several such glance views across the screen. *See, e.g.,* ’330 patent at Fig. 3; 7:12-25 (“When [a visual representation of a document] is as big as it can get, it is plastered against the workspace window and cannot be moved any closer.”) As these references show, it was well known in the art that it was useful to display glance views after sliding a cursor over a document representation, so that a user could readily access additional information about the document that was not ascertainable from the document representation. This use of glance views was routine to a person of skill in the art by 1996. The results of representing a set of data units with a set of corresponding document representations, and then generating glance views of particular documents to specify additional information about the represented data unit were predictable to those of skill in the art before 1996. It allowed simultaneous and integrated use of both a more compact representation of a set of documents (“document representation”), so that many documents could be seen and interacted with at once, and a more detailed representation of a particular document (“glance view”), to provide more detailed information about a particular document.

If it is determined that any of the base references listed above do not disclose representing a set of data units with a set of corresponding document representations, and then generating glance views of particular documents to specify additional information about the represented data unit, it would have been obvious to a person of skill in the art before 1996 to

combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the CHI ’92 article, the ’724 patent, the ’330 patent, TR-1070, or the JP ’661 references. For example, design incentives would have prompted such modification, because it was well known to be desirable to be able to have visual and intuitive representations of collections of documents, and the concept of working with documents organized in piles was known to be intuitively appealing to users. *See, e.g.*, ’724 patent at 2:42-60. Furthermore, it was known that in general, users working with documents needed to perform tasks that required visualizing larger sets of documents, as well as tasks that required obtaining more detailed information about particular documents. The use of document representations for larger sets in conjunction with glance views for more detailed information was a known and predictable solution to this requirement, and thus there was a design incentive to use it where users were working with sets of documents, as is the case with the base references. Market forces would also have prompted such modification, for example because it was known before 1996 as described above that users could be more efficient when using intuitive, well-designed interfaces. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions described in the base reference, and the representation of a set of data units with a set of corresponding document representations, and the generation of glance views of particular documents to specify additional information about the represented data unit, would perform the same known and predictable functions known in the art as described above, and/or in the CHI ’92 article, the ’724 patent, the ’330 patent, TR-1070, or the JP ’661 references. The results of such combination would also be predictable to a person of skill in the art before 1996. One would be able to work with groups of documents using document representations, while also being able to obtain a glance view specifying additional information about a document by sliding a cursor over a document representations, thus allowing work focused on both larger and smaller scale representations of documents from the same interface.

### 8.1 Sliding Without Clicking To Display The Glance View

Many of Mirror Worlds' asserted claims—'227:25-29; '313:1-4; '313:9-11; '427:1-2, 5, and 7; '427:8-10, 13, and 15; '427:16-19, 22, and 24; '427:25-26, 29, and 31; '427:32-34, 37, and 39; and '999:1—require that the glance view of the document be displayed in response to "sliding without clicking of the cursor" over the document representation whose glance view is to be displayed, or something similar. The concept of sliding a cursor without clicking over a document representation in order to display a glance view by was well known, commonly used, and routine to those of skill in the art before 1996.

For example, the CHI '92 article illustrates a number of different techniques for displaying glance views after sliding a cursor without clicking over a stack of document representations. *See, e.g.,* CHI '92 article at Figs. 4, 5, 7. The '724 patent also describes displaying glance views after sliding a cursor without clicking over a pile of document representations. '724 patent at Figs. 4a-f, 10a-b, 12a-b. *See also* JP '661 at Figs. 10, 11, 12. As these references show, it was well known in the art that it was useful to display glance views after sliding a cursor without clicking over a document representation, so that a user could readily access additional information about the document that was not ascertainable from the document representation. This use of glance views was routine to a person of skill in the art by 1996.

If it is determined that any of the base references listed above do not disclose sliding a cursor over a document representation without clicking in order to display a glance view, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the references mentioned above. The reasons for this obviousness include all the same reasons described above in section 8.0. Additional design incentives would have prompted such modification, because it was well known that in user interface design it is efficient to minimize the number of actions, such as clicks, that must be taken in order to obtain information, and the

display of a glance view using cursor movement without a click is useful and efficient. Furthermore, it is known in user interface design that a particular action, such as a mouse click in a particular location (such as over a document representation) can only invoke one response at a time. Thus, if clicking the document representation is reserved for another function—in most interfaces, this would select the document representation—then clicking cannot be used to invoke the glance view. In that case, design incentives would require something else, such as location of the cursor ("sliding without clicking"), to invoke the glance view. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions described in the base reference, and the sliding a cursor over a document representation without clicking in order to display a glance view would perform the same known and predictable functions known in the art as described above, and/or described in the CHI '92 article, the JP '661, or the '724 patent references. The results of such combination would also be predictable to a person of skill in the art before 1996. One would be able to obtain a glance view specifying additional information about a document by sliding a cursor over a document representation without clicking, thus reserving the act of clicking on a document representation for a different function such as selecting the document represented by the document representation.

### 8.2 Document Representations With Markings Common To A Class Of Documents

At least one of Mirror Worlds' asserted claims, '427:34, requires "visually identifying attributes" of documents using "markings" that are "visible in the displayed stack" and that are "common to a class of documents," or something similar. The concept of using markings common to a class of documents to visually identify the class of document to the user was well known, commonly used, and routine to those of skill in the art before 1996.

For example, virtually all GUIs for operating systems, including Finder in Macintosh System 7, NextSTEP, and Microsoft Windows, could use icons to represent files, with different icons being used for different types of documents. This use of icons was well-known and routine

to one of skill in the art by 1996. *See also, e.g.*, '724 patent at 7:1-6. Another example is the '602 patent, which describes how to generate and display a standard, content-based icon for each document. *See, e.g.,* 3:13-26, 4:62-5:1. Another example is the '135 patent, which describes using "detailed miniaturized images of all documents possessed by the user" and how to create those miniatures. *See, e.g.,* the '135 patent at Abstract.

If it is determined that any of the base references listed above do not disclose using markings common to a class of documents to visually identify the class of document to the user, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the references mentioned above. The reasons for this obviousness include all the same reasons described above in section 8.0.

**8.3 Glance View With Command Buttons**

Some of Mirror Worlds' asserted claims—'427:32-34, 37, and 39—require displaying "a set of command buttons" concurrently with and in the same display as the "glance view." The concept of displaying command buttons concurrently with a glance view was well known and routine to those of skill in the art before 1996.

For example, TR-1070, the '330 patent and the '724 patent both disclose displaying command buttons for browsing through glance views. '330 patent at Fig. 3; '724 patent at Figs. 13a, 13b, 22e; TR-1070 at YALE000440-441. As these references show, it was well known in the art that command buttons could be used to manipulate a representation of a document, for example to play a video or sound file, or to page through a multi-page document. This use of command buttons was routine to a person of skill in the art by 1996.

If it is determined that any of the base references listed above do not disclose displaying command buttons concurrently with a glance view, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill

in the art as described above, and/or any of the references mentioned above. The reasons for this obviousness include all the same reasons described above in section 8.0. Additional design incentives would have prompted such modification, because it was well known that command buttons were useful for manipulating representations of data units, particularly where the initial representation of the data unit does not normally represent all the information in the data unit, as is the case with multi-page documents and video files, for example. Additional market forces would also have prompted such modification for this reason, because there was market demand for the ability to browse through representations of data units. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions described in the base reference, and the displaying command buttons concurrently with a glance view would perform the same known and predictable functions known in the art as described above, and/or described in the TR-1070, ’330 patent or the ’724 patent references. The results of such combination would also be predictable to a person of skill in the art before 1996. The display of a glance views would concurrently include a display of command buttons.

### 8.4 Document Representation(s) With Timestamps

Some of Mirror Worlds’ asserted claims—’227:12 and ’227:25-29—require the document representation (browse card) to “include the timestamp of the respective data unit,” or something similar, such as “associating the alternative version data unit with the chronological indicator of the another data unit.” The concept of including the timestamp of the data unit with the document representation, and ordering the representations by timestamp, was well known and routine to those of skill in the art before 1996.

For example, the Macintosh OS (System 7) provides the ability to list the files in a directory through the Finder. That listing includes a representation of each document (file) that can include an icon, and the file’s name, size, and timestamp. The listing could be sorted by timestamp. Many other operating systems, including NextSTEP, and Windows provided the same functionality. The ability to display a timestamp along with a representation of a document

in a file listing, and order the file listing by the timestamp, is also present in UNIX and DOS and is routine and commonly used, as would be known by one of skill in the art. The concept of including the timestamp of the data unit with the document representation, and ordering the representations by timestamp, is also described in the references. *See, e.g.*, CHI '92 article at Fig. 6; Using Lotus Magellan at pp.36-37; '724 patent at Fig.4e-f; 13a, 7:1-6 ("each document may be a miniature of the first page of the actual document, an icon of the document type … or a set of attributes relating to the document").

If it is determined that any of the base references listed above do not disclose including the timestamp of the data unit with the document representation, and ordering the representations by timestamp, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or any of the references mentioned above. The reasons for this obviousness include all the same reasons described above in section 8.0. Additional design incentives would have prompted such modification, because it was well known that sorting and searching by date was useful and desirable in locating documents. Additional market forces would also have prompted such modification for this reason, because there was market demand for the ability to sort and search by date. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions described in the base reference, and the including the timestamp of the data unit with the document representation, and ordering the representations by timestamp would perform the same known and predictable functions known in the art as described above, and/or described in the CHI '92 article, Lotus Magellan, or the '724 patent references. The results of such combination would also be predictable to a person of skill in the art before 1996. The document representations would include a timestamp, and could be ordered/sorted by timestamp.

## 9.0 Enterprise Information Management System

One of Mirror Worlds' asserted claims—'999:1—requires "operating an enterprise information management system." The concept of operating an enterprise information management system was well known, commonly used, and routine to those of skill in the art before 1996.

For example, the '972 patent discloses and a file system "designed to support the storage of, and access to, remote files" on data servers and operating on top of the standard operating system running on the client computers. *See, e.g.,* the '972 patent at Abstract. Another example is the SFS article, which describes an implementation build on a Unix client-server foundation, as well as the use of distributed file systems for "file sharing among groups of people and over wide geographic areas." The result of operating an enterprise information management system was predictable to those of skill in the art before 1996. A client-server architecture allows the benefits and duties of a computing system be distributed across computers and provides greater data security because servers are generally safer than client computers.

If it is determined that any of the base references listed above do not disclose operating an enterprise information management system, it would have been obvious to a person of skill in the art before 1996 to combine any such reference with the knowledge of a person of skill in the art as described above, and/or the '972 patent. Market forces would also have prompted such modification because it was well known before 1996 that large-scale client-server networking would become a conventional system architecture in business and academic environments. In the combination, each of the original elements of the base reference would be performing the same known and predictable functions described in the base reference, and the enterprise information management system would perform the same known and predictable functions described in the '972 patent. The results of such combination would also be predictable to a person of skill in the art before 1996.

### C. Additional Obviousness References Showing The State Of The Art

#### 20. United States Patent No. 5,060,135 (Levine *et al.*)

The Levine ’135 patent describes a graphical user interface that uses the concept of stacks for document organization. It describes using "detailed miniaturized mages of all documents possessed by the user," and how to create these miniatures, which it calls "stamps." Abstract, 3:27-4:29; 21:60-25:14. It describes how to use the drag and drop concept to move individual documents and stacks of documents around the desktop. 4:41-62. It describes using an "in-box" to collect documents received from other computers on a network, or from email. 5:12-17. It also describes a variety of applications that can be used to interact with the "stacks" and "stamps" that represent documents in this particular graphical user interface. In particular, the Levine ’135 patent describes an "info" application. This application is invoked by dragging the icon of the application over the stamp of a particular document, and it then displays more information about the document represented by the stamp, specifically the "history and composition" of the document. Thus, the Levine ’135 patent shows that by 1991 the concept of using a "stack" of document representations to represent a set of documents was known, as was the concept of associating two different kinds of document representations (*i.e.*, a 'browse card' and a 'glance view') with a document.

#### 21. United States Patent No. 6,262,732 (Coleman *et al.*)

The Coleman ’732 patent describes an improved method of graphically representing a "stack" of pages using miniaturized versions of those pages. Figs. 2, 3a, 3b, 3c. It describes techniques for creating miniature replicas of pages of a document and using buffers to improve the speed of displaying them. 14:10-15:15. It describes organizing the miniatures into stacks, and browsing through and manipulating those stacks, including by dragging and dropping stacks or portions of stacks into other stacks. Figs. 2, 3a, 3b, 3c; 6:34-7:55. The Coleman ’732 patent also describes associating a 'descriptor block' with each of the stacks. 12:30-56. Thus, the Coleman ’732 patent shows that the concept of using a "stack" to represent a set of documents

was well known, as was the concept of associating two different kinds of document representations (*i.e.*, a 'browse card' and a 'glance view') with a document.

**22. "The Role of Time in Information Processing: A Survey," by Bolour *et al.*, ACM SIGART Bulletin (Apr. 1982)**

The SIGART '82 article is an early survey of approximately 70 references addressing the role that time plays in computerized information systems. Among other things relevant to the patents-in-suit, it describes how K.M. Kahn designed a module to store and retrieve "inexact temporal facts." *See* SIGART '82 article, p. 35. It also describes how Ariav and Morgan designed and implemented a system which handled time in a linear, non-hierarchical "date line" fashion. *See* SIGART '82 article, p. 47. Although the individual articles may comprise prior art for particular concepts, the survey as a whole shows that by 1982, the concept of time organization of data in a computer system was well known in the art.

**23. United States Patent No. 5,764,972 (Crouse)**

The Crouse '972 patent describes a "completely transparent" archiving file system. Abstract. It describes an archiving file system running on top of the native file system and which allows storage and retrieval of remote files based on selectable archival attributes. 4:22-42. The Crouse '972 patent shows that the concept of automatic archiving and the concept of archiving files according to "attributes" was well known in the art by 1993.

**24. United States Patent No. 5,479,602 (Baecker & Small)**

The '602 patent describes creating and displaying document icons or thumbnails that are content-based rather than a standard icon. '602 patent at 3:12-20. It describes techniques for partitioning a document representation and generating a replica. '602 patent at 3:27-41. Thus the '602 patent shows that the concept of creating alternative or abbreviated document representations was well known by 1995.

**25. "Recovery Concepts for Data Sharing Systems," by Ehrard Rahm (1991)**

The Rahm reference describes datasharing in a distributed system architecture in which recovery of data is possible. Rahm at 368. It also describes how to generate a "global log file," which stores in chronological order any modifications to documents. Rham at 368. Thus, the Rahm reference shows that the concept of indexing the contents of a filesystem, coupled with an enterprise management system or distributed network, was well known by 1991.

**26. Inside Macintosh: Files (1992) (HFS Manual)**

The HFS file system used on Macintosh computers organized every file received by the computer on a volume-by-volume basis. *See, e.g.,* Inside Macintosh: Files at 2-53. The HFS file system also included a catalog file containing metadata about each file, that could be searched using the function "pbCatSearch" in order to find files based on their metadata. *See, e.g.,* Inside Macintosh: Files at 2-53. Thus, the HFS filesystem shows that the concepts of organizing every data unit, and organizing and searching those data units according to each file's metadata attributes was well known by 1992.

**27. Email clients and systems (*E.g.* Elm, Pine, Eudora, Outlook, Lotus Notes, cc:Mail)**

Email was widely known and commonly used before 1996, and the functionality of email clients (such as Elm, Eudora, Outlook, and Lotus Notes) was well known and familiar to those of skill in the art by that time. Similarly, the functionality of email servers (*e.g.* Lotus Notes Server / Domino, Microsoft Exchange) was well known to those of skill in the art before 1996.

Databases were widely known and commonly used before 1996, and the functionality of databases, including relational databases, was well known and familiar to those of skill in the art by that time. *See generally,* C. J. Date, An Introduction to Database Systems, 3rd ed. (Addison-Wesley, 1981).

### 28. Software-Distribution and Change Management Software (*E.g.* Novadigm's EDM)

By 1996, electronic software distribution and management software was well known and commonly used to solve problems of software distribution and change-management in enterprises. For example, Novadigm Inc., (Nasdaq: NVDM) sold a product called EDM to enterprises. As reported in Novadigm's 6/28/1996 Form 10-K, "Novadigm, Inc. ('Novadigm' or the 'Company') is a provider of automated software management solutions that reduce the cost and complexity of managing enterprise client/server and internet computing environments. The Company's products, collectively known as Enterprise Desktop Manager(TM) ("EDM"), automate the 'continuous configuration' of distributed software across thousands of desktops and servers for medium and large organizations in the financial services, government, transportation, telecommunications, healthcare, utilities and other industries. The Company's solutions are highly scaleable and interoperable, and therefore uniquely suitable for managing rapidly changing business software across large distributed corporate and public networks. Novadigm's patented technologies for software management allow for high levels of automation in managing distributed software configurations, ensuring that the right software is available to the right users at the right time without manual intervention."

### 29. The World Wide Web

By 1995, the World Wide Web was well known and commonly used. As early as 1992, GUI web browsers were known and used. By 1993, a GUI browser was available for the Mac. *See W3C: A Little History of the Web* (http://www.w3.org/History.html, last visited November 3, 2008). In 1994, the World Wide Web Consortium was founded, and the Second International WWW Conference was held (entitled "Mosaic and the Web") in Chicago. *See Id.*

### 30. "Names should mean What, not Where" by O'Toole & Gifford (1992)

By 1992, it was known that an alternative to the traditional hierarchical or "tree structured" file system was a "semantic file system," wherein files are located by searching an index of their contents or attributes. This Gifford article, like the SFS article, describes an implemented semantic file system. Through this file system, users seeking files obtain them by

entering search criteria and browsing through "virtual directories" containing the results of those searches, rather than browsing through traditional static directories. These Gifford articles also show that it was known that a user need not name files or assign them a specific location in a traditional hierarchy. Instead, file storage and retrieval can be handled automatically by a computer, by automatically indexing the attributes and contents of the files.

**31. United States Patent No. 5,649, 182 (Reitz)**

Reitz describes a method for organizing data based on time, as well as for filtering the data based on its attributes in order to generate subsets of time-ordered data. *See* Reitz at Abstract; 1:22-2:50. Thus, as recognized by the examiner during prosecution of the '227 patent, Reitz shows that it was known to generate "subsets of the main stream of records organized by timestamps and determined by attributes."

**32. "Representation in Virtual Space: Visual Convention in the Graphical User Interface" by L. Staples (1993)**

By 1993 it was well known to use three-dimensional virtual spaces as elements of a graphical user interface. It was also well known that this can be accomplished using perspective / foreshortening. *See* Staples at Figs. 4, 6, and 7. In fact, it was known that such a three-dimensional space could be used to provide an alternative interface for finding or browsing files. Staples at 350 ("Attempts at perspective have already been applied to the GUI. A novel alternative to the Macintosh Finder is currently available in Ark's Workspace software (Fig. 7).") It was also known that a receding foreshortened stack of document representations could be used to represent a set of documents. Staples at Figs. 9-11.

**33. "Using Collaborative Filtering To Weave An Information Tapestry" by D. Goldberg et al. (1992)**

The Tapestry reference shows that by 1992, it was well known that the increasing use of electronic documents, including electronic mail, was "resulting in users being inundated by a huge stream of incoming documents." Tapestry at p. 1. Tapestry teaches that filtering (*i.e.* searching) is an efficient way to manage these large streams of electronic documents. Tapestry

also describes a system that is used to organize both all current information and to act as a repository of all older information.

**34. United States Patent No. 5,729,730 (Wlaschin '730 patent)**

The Wlaschin '730 patent describes an information management and database system for storing any type of data in a table and providing an interface to other application programs in order to allow efficient and effective searching across diverse types of files. Abstract, 2:52-60, 525-7:16. In describing the advantages over the prior art, Wlaschin describes a database with increased flexibility, search time and smaller memory requirements and that supports text attributes. 2:31-35. It also describes the integration, into a single database, of preexisting source files developed under various types of application programs such as other databases, spreadsheets and word processing programs. 2:43-47. The Wlaschin '730 patent also describes storing data in a table based on an object identification number (OID), which is generated using a timestamp, session identification and tiebreaker to resolve conflicts between identical timestamps. 8:16-57. Figure 4. The Wlaschin '730 patent goes on to describe techniques for allowing users to search for data, including text-indexing, date-indexing and associative queries. 13:50-15:10. Thus, the Wlaschin '730 patent shows that by 1995, the concept of organizing data from diverse applications in a table based on timestamp information was well known, as was the need and techniques for providing users with increased search flexibility and decreased search time.

**35. Washington Post Article, "The Cyber-Road Not Taken" by David Gelernter (1994)**

The "Cyber-Road Not Taken" article shows that well before 1996 it was known that an alternative to the traditional hierarchical or "tree structured" file system, wherein files are located by searching an index of their contents or attributes, was desirable. Particularly, the article explains that the author does not want to organize his information into files, or to name those files. It describes the concept of a "lifestream," which "captures your whole life, in terms of chunks of information." And it describes the need to be able to both visualize this stream of

documents or info-chunks, as well as to be able to filter or search it in order to only display certain kinds of documents.

## II. PRIOR ART: DERIVATION

### A. Yale Technical Report TR-1070

The claims of the patents-in-suit are invalid under 35 U.S.C. § 102(f) because, on information and belief, the named inventors of those patents did not invent the subject matter of those patents. On information and belief, the claimed inventions were derived, at least in part, from Yale Technical Report TR-1070. Yale Technical Report TR-1070 entitled "The 'Lifestreams' Approach to Reorganizing the Information World" lists as named authors Nicholas Carriero, Scott Fertig, Eric Freeman, and David Gelernter. The Yale Technical Report TR-1070 contains key features of the asserted claims of the patents-in-suit, as shown in the appended claim charts.

### B. 1995 AAAI Fall Symposium Article

The claims of the patents-in-suit are invalid under 35 U.S.C. § 102(f) because, on information and belief, the named inventors of those patents did not invent the subject matter of those patents. On information and belief, the claimed inventions were derived, at least in part, from a 1995 AAAI Fall Symposium article entitled, "Lifestreams: Organizing your Electronic Life." The 1995 AAAI Fall Symposium article entitled, "Lifestreams: Organizing your Electronic Life" lists as named authors Eric Freeman and Scott Fertig. "Lifestreams: Organizing your Electronic Life" contains key features of the asserted claims of the patents-in-suit.

### C. Yale Technical Report TR-1083

The claims of the patents-in-suit are invalid under 35 U.S.C. § 102(f) because, on information and belief, the named inventors of those patents did not invent the subject matter of those patents. On information and belief, the claimed inventions were derived, at least in part, from Yale Technical Report TR-1083. Yale Technical Report TR-1083 entitled "Lifestreams:

Organizing your Electronic Life" lists as named authors Eric Freeman and Scott Fertig. The Yale Technical Report TR-1083 contains key features of the asserted claims of the patents-in-suit.

**D. Yale Research Report RR-1098**

The claims of the patents-in-suit are invalid under 35 U.S.C. § 102(f) because, on information and belief, the named inventors of those patents did not invent the subject matter of those patents. On information and belief, the claimed inventions were derived, at least in part, from Yale Research Report RR-1098. Yale Research Report TR-1098 entitled "Lifestreams: Bigger than Elvis" lists as named authors Nicholas Carriero, Scott Fertig, Eric Freeman, and David Gelernter. The Yale Research Report TR-1098 contains key features of the asserted claims of the patents-in-suit.

**E. The '227 patent, '313 patent, and '427 patent**

The claim of the '999 patent is invalid under 35 U.S.C. § 102(f) because, on information and belief, the named inventors of the '999 patent did not invent the subject matter of the '999 patent. On information and belief, the claimed invention was derived, at least in part, from the '227 patent and the '313 patent. The '999 patent is a continuation-in-part of the '227 patent and the '313 patent. The '227 and the '313 list as inventors Eric Freeman and David Gelernter. The '227 and the '313 patent contain key features of the asserted claim of the '999 patent, as shown in the appended claim charts.

The facts pertaining to the inventors' derivation of the claimed subject matter are being further investigated, and Apple expects that discovery taken in this litigation will further reveal facts concerning this defense. Accordingly, Apple reserves the right to supplement or modify these Invalidity Contentions with respect to derivation as further discovery occurs.

## III. INVALIDITY UNDER 35 U.S.C. § 112 & PRIORITY

The claims of the patents-in-suit are invalid under 35 U.S.C. § 112 because the claims are indefinite, lack a proper written description, and/or do not enable on of ordinary skill

in the art at the time the invention was made to make or use the claimed invention. Pursuant to P.R. 3-3(d), Apple lists below the grounds upon which the asserted claims of the patents-in-suit are invalid for failure to meet the requirements of 35 U.S.C. §§ 112(1) and/or 112(2). Furthermore, Mirror Worlds states in its 3-1(e) disclosure that the earliest priority date to which all claims of the patents-in-suit are entitled is June 28, 1996, the filing date of the '227 patent. However, the claims recited in the patents-in-suit contain limitations that are not supported by the application to which they claim priority. Therefore, the claims of the '313, '427, and '999 patents are not entitled to a June 28, 1996 priority date.

### A. Independent Claim 1 and Dependent Claims 2, 3, 4, 5, 6, 9, 10, 11, and 12 of the '227 Patent

Independent Claim 1 and dependent claims 2-6 and 9-12 of the '227 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

The specification of the '227 patent does not disclose and/or does not clearly link structure for performing the functions corresponding to the "means for generating a main stream of data units and at least one substream," "means for receiving data units from other computer systems," "means for generating data units by 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" limitations, and accordingly independent Claim 1 and dependent claims 2-6 and 9-12 of the '227 patent are invalid as indefinite under 35 U.S.C. § 112(2).

Furthermore, the specification of the '227 patent does not disclose and/or does not clearly link structure for performing the functions corresponding to the "wherein the means for receiving further comprises means for receiving data units from the World Wide Web," "wherein said means for receiving further comprises means for receiving data units from a client

computer," "means for displaying alternative versions of the content of the data units," "means for archiving a data unit associated with a timestamp . . . ," "means for operating on any of the streams using a set of operations selected by a user," "means to generate substreams from existing substreams," "means for generating a data unit comprising an alternative version of the content of another data unit" or "means for associating the alternative version data unit with the chronological indicator of another data unit" limitations, and accordingly dependent claims 4-6 and 9-12 of the '227 patent are invalid as indefinite under 35 U.S.C. § 112(2).

In addition, independent claim 1 and dependent claims 2-6 and 9-12 of the '227 patent are invalid for failure to satisfy the written description and/or enablement requirements because the specification of the '227 patent does not disclose and/or enable the "timestamp to identify," "chronological indicator having the respective timestamp" and "including each data unit according to the timestamp in the respective chronological indicator" requirements of these claims, particularly insofar as the '227 patent does not disclose and/or enable the relationship, if any, between a "timestamp to identify" and a "chronological indicator having the respective timestamp" or disclose and/or enable including data units according to the timestamp in the respective chronological indicator.

### B. Independent Claim 13 and Dependent Claims 14, 15, 16, 17, 20 and 22 of the '227 Patent

Independent Claim 13 and dependent claims 14-17, 20 and 22 of the '227 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Independent claim 13 and dependent claims 14-17, 20 and 22 of the '227 Patent are invalid for failure to satisfy the written description and/or enablement requirements because the specification of the '227 Patent does not disclose and/or enable the "timestamp to identify," "chronological indicator having the respective timestamp" and "including each data unit according to the timestamp in the respective chronological indicator" requirements of these claims, particularly insofar as the '227 Patent does not disclose and/or enable the relationship, if

any, between a “timestamp to identify” and a “chronological indicator having the respective timestamp” or disclose and/or enable including data units according to the timestamp in the respective chronological indicator.

### C. Independent Claim 25 and Dependent Claims 26-29 of the ’227 Patent

Independent Claim 25 and dependent claims 26-29 of the ’227 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

The specification of the ’227 patent does not disclose and/or does not clearly link structure for performing the functions corresponding to the "means for generating a main stream of data units and at least one substream . . .,” “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 . . .,” “means for selecting which data units are represented on the display device . . .,” and “means for selecting one or more of the document representations with a point device so that the data units represented by the selected document representations are further displayed . . .” limitations, and accordingly independent claim 25 and dependent claims 26-29 of the ’227 Patent are invalid as indefinite under 35 U.S.C. § 112(2).

In addition, independent claim 25 and dependent claims 26-29 of the ’227 patent invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’227 patent does not disclose and/or enable the “chronological indicator having a respective timestamp” and “including each data unit according to the timestamp in the respective chronological indicator” requirements of these claims, particularly insofar as the ’227 patent does not disclose and/or enable the relationship, if any, between a “timestamp” and a “chronological indicator having the respective timestamp” or disclose and/or enable including data units according to the timestamp in the respective chronological indicator.

**D. Independent Claim 1 and Dependent Claims 2, 3 and 4 of the ’313 Patent**

Independent claim 1 and dependent claims 2-4 of the ’313 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Independent claim 1 and dependent claims 2-4 of the ’313 patent are invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’313 patent does not disclose and/or enable the “time-based indicators” requirement of these claims.

**E. Independent Claim 9 and Dependent Claims 10 and 11 of the ’313 Patent**

Independent claim 9 and dependent claims 10 and 11 of the ’313 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Independent claim 9 and dependent claims 10 and 11 of the ’313 patent are invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’313 patent does not disclose and/or enable the “time-based indicators” requirement of these claims.

**F. Independent Claim 1 and Dependent Claims 2, 5, and 7 of the ’427 Patent**

Independent claim 1 and dependent claims 2, 5 and 7 of the ’427 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Independent claim 1 and dependent claims 2, 5 and 7 of the ’427 patent are invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “selected indicators” requirement of these claims.

In addition, dependent claim 2 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “said selected indicators are time-based” requirement of this claim.

Furthermore, dependent claim 7 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document” requirement of this claim.

### G. Independent Claim 8 and Dependent Claims 9, 10, 13, and 15 of the ’427 Patent

Independent claim 8 and dependent claims 9, 10, 13 and 15 of the ’427 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Independent claim 8 and dependent claims 9, 10, 13 and 15 of the ’427 patent are invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 Patent does not disclose and/or enable the “selected indicators” requirement of these claims.

In addition, dependent claim 9 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “said selected indicators are time-based” requirement of this claim.

Furthermore, dependent claim 15 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ‘427 patent does not disclose and/or enable the “important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document” requirement of this claim.

### H. Independent Claim 16 and Dependent Claims 17, 18, 19, 22, and 24 of the ’427 Patent

Independent claim 16 and dependent claims 17-19, 22 and 24 of the ’427 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Independent claim 16 and dependent claims 17-19, 22 and 24 of the ’427 patent are invalid for failure to satisfy the written description and/or enablement requirements because

the specification of the ’427 patent does not disclose and/or enable the “selected indicators” requirement of these claims.

In addition, dependent claim 17 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “said selected indicators are time-based” requirement of this claim.

Furthermore, dependent claim 24 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document” requirement of this claim.

### I. Independent Claim 25 and Dependent Claims 26, 29, and 31 of the ’427 Patent

Independent claim 25 and dependent claims 26, 29 and 31 of the ’427 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Independent claim 25 and dependent claims 26, 29 and 31 of the ’427 patent are invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “chronological indicators” requirement of these claims.

Furthermore, dependent claim 31 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document” requirement of this claim.

### J. Independent Claim 32 and Dependent Claims 33, 34, 37, and 39 of the ’427 Patent

Independent claim 32 and dependent claims 33, 34, 37 and 39 of the ’427 patent are invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Independent claim 32 and dependent claims 33, 34, 37 and 39 of the ’427 patent are invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “time-based indicators” requirement of these claims.

In addition, dependent claim 34 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “visually identifying attributes of selected documents in the displayed stack of document representations by markings that are visible in the displayed stack, each marking being common to a class of documents” requirement of this claim.

Furthermore, dependent claim 39 is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’427 patent does not disclose and/or enable the “important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document” requirement of this claim.

**K. Independent Claim 1 of the ’999 Patent**

Claim 1 of the ’999 patent is invalid for failure to satisfy the written description, enablement, and/or definiteness requirements of 35 U.S.C. § 112 for at least the following reasons.

Claim 9 of the ’999 patent is invalid for failure to satisfy the written description and/or enablement requirements because the specification of the ’313 patent does not disclose and/or enable the “glance views being displayed essentially in real time in response to passing a cursor over respective ones of the browse cards” requirement of this claim.

A more detailed basis for Apple’s 35 U.S.C. § 112 invalidity defenses will be set forth in Apple’s expert report of invalidity and, for failure to satisfy the definiteness requirement, in its claim construction briefing to the Court. Apple reserves the right to amend and/or supplement these Invalidity Contentions based on 35 U.S.C. § 112 as discovery progresses. Apple further reserves the right to amend and/or supplement these Invalidity Contentions based

on 35 U.S.C. § 112 depending on the claim construction positions taken by Mirror Worlds, and the Court's claim construction Order.

Date: May 11, 2009

*/s/ Stefani Smith*

Matthew Powers
Lead Attorney
Steven Cherensky
Sonal N. Mehta (*Pro Hac Vice*)
Stefani C. Smith (*Pro Hac Vice*)
WEIL, GOTSHAL & MANGES LLP
201 Redwood Shores Parkway
Redwood Shores, CA 94065
650-802-3000 (Telephone)
650-802-3100 (Facsimile)
matthew.powers@weil.com
steven.cherensky@weil.com
sonal.mehta@weil.com
stefani.smith@weil.com

Eric M. Albritton
Texas State Bar No. 00790215
ALBRITTON LAW FIRM
P.O. Box 2649
Longview, Texas 75606
(903) 757-8449 (Telephone)
(903) 758-7397 (Facsimile)
ema@emafirm.com

**Attorneys for Defendant**
**APPLE INC.**

# EXHIBIT 13A

# Invalidity of U.S. Pat. No. 6,006,227

## by Mander *et al.*, A 'Pile' Metaphor for Supporting Casual Organization of Information, CHI '92

Based upon the claim interpretations Plaintiff appears to be asserting and the applications of those interpretations to Defendant's products in Plaintiff's Preliminary Infringement Contentions, Mander, Piles anticipates and/or renders obvious, alone or in combination with other prior art identified in Defendant's Preliminary Invalidity Contentions, the asserted claims as described in part below. Nothing in these contentions should be interpreted as an acquiescence to or assertion of a particular claim construction by Defendant.[1]

| Claim Language | Disclosure |
|---|---|
| 1. A computer system which organizes each data unit received by or generated by the computer system, comprising: | *See, e.g.*, pp. 627-628. |
| 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; | *See, e.g.*, p. 629. |
| means for receiving data units from other computer systems: | *See, e.g.*, p. 633. |
| means for generating data units by the computer system; | *See, e.g.*, p. 627-629. |
| means for selecting a timestamp to identify each data unit; | *See, e.g.*, p. 631; Fig. 6. |
| means for associating each data unit with at least one chronological indicator having the respective timestamp; | *See, e.g.*, p. 631; Fig. 6. |
| means for including each data unit according to the timestamp in the respective chronological indicator in the main stream; and | *See, e.g.*, pp. 628-629. |
| means for maintaining the main stream and the substreams as persistent streams. | *See, e.g.*, p. 629; Fig. 3. |

[1] To the extent Mander, Piles is argued by Plaintiff or found by the Court not to explicitly teach certain limitations in the asserted claims, such limitations would have been inherent and/or obvious as described in Defendant's Preliminary Invalidity Contentions.

| Claim Language | Disclosure |
|---|---|
| 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. | *See, e.g.*, p. 633. |
| 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. | *See, e.g.*, p. 633. |
| 6. The computer system according to claim 1, further comprising:<br>means for displaying alternative versions of the content of the data units. | *See, e.g.*, p. 629; Figs. 1-2. |
| 9. A computer system according to claim 1 further comprising:<br>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:<br>means for operating on any of the streams using a set of operations selected by a user. | *See, e.g.*, Figs. 3-5. |
| 11. The computer system of claim 1 further comprising: | *See, e.g.*, p. 633. |

| Claim Language | Disclosure |
|---|---|
| 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 | *See, e.g.*, p. 629; Figs. 1-2. |
| means for associating the alternative version data unit with the chronological indicator of the another data unit. | *See, e.g.*, p. 629; Figs. 1-2. |
| | |
| 13. A method which organizes each data unit received by or generated by a computer system, comprising the steps of: | *See, e.g.*, pp. 627-628. |
| 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; | *See, e.g.*, p. 629. |
| receiving data units from other computer systems; | *See, e.g.*, p. 633. |
| generating data units in the computer system; | *See, e.g.*, p. 627-629. |
| selecting a timestamp to identify each data unit; | *See, e.g.*, p. 631; Fig. 6. |
| associating each data unit with at least one chronological indicator having the respective timestamp; | *See, e.g.*, p. 631; Fig. 6. |
| including each data unit according to the timestamp in the respective chronological indicator in at least the main stream; and | *See, e.g.*, pp. 628-629. |
| maintaining at least the main stream and the substreams as persistent streams. | *See, e.g.*, p. 629; Fig. 3. |
| | |
| 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 | *See, e.g.*, Figs. 1-2, 6. |

| Claim Language | Disclosure |
|---|---|
| display device as visual streams. | |
| | |
| 16. The method of claim 15, wherein the step of displaying the streams further comprises the steps of: | *See, e.g.*, Figs. 1-2, 6. |
| 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 | *See, e.g.*, Fig. 6. |
| b) displaying the selected segments. | *See, e.g.*, Fig. 6. |
| | |
| 17. The method of claim 13, wherein each data unit includes textual data, video data, audio data and/or multimedia data. | *See, e.g.*, p. 633. |
| | |
| 20. The method of claim 13 further comprising the step of:<br>displaying data from one of the data units in abbreviated form. | *See, e.g.*, p. 629; Figs. 1-2. |
| | |
| 22. The method of claim 13, further comprising the step of:<br>archiving data units having timestamps older than a specified time point. | |
| | |
| 25. A computer system for organizing each data unit received by or generated by the computer system, comprising: | *See, e.g.*, pp. 627-628. |
| 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; | *See, e.g.*, p. 629. |
| means for associating each data unit with at least one chronological indicator having a respective timestamp which identifies the data unit; | *See, e.g.*, p. 631; Fig. 6. |

| Claim Language | Disclosure |
|---|---|
| means for including each data unit according to the timestamp in a respective chronological indicator in the main stream; | *See, e.g.*, pp. 628-629. |
| means for maintaining the main stream and the substreams as persistent streams; | *See, e.g.*, p. 629; Fig. 3. |
| 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; | *See, e.g.*, p. 629; Figs. 1-2, 6(a). |
| 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 | *See, e.g.*, Fig. 6. |
| 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. | *See, e.g.*, p. 629; Figs. 1-2. |
| | |
| 26. A computer system as in claim 25, wherein the document representations form a visual stream having a three-dimensional effect. | *See, e.g.*, Figs. 1-2, 6. |
| | |
| 27. A computer system as in claim 26, wherein the three-dimensional effect further comprises a perspective view. | *See, e.g.*, Figs. 1-2, 6. |
| | |
| 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. | *See, e.g.*, Figs. 1-2, 6. |
| | |
| 29. A computer system as in claim 25, wherein the alternate version is an abbreviated version. | *See, e.g.*, p. 629. |

# EXHIBIT 13B

# Invalidity of U.S. Pat. No. 6,638,313

## by Mander et al., A 'Pile' Metaphor for Supporting Casual Organization of Information, CHI '92

Based upon the claim interpretations Plaintiff appears to be asserting and the applications of those interpretations to Defendant's products in Plaintiff's Preliminary Infringement Contentions, Mander, Piles anticipates and/or renders obvious, alone or in combination with other prior art identified in Defendant's Preliminary Invalidity Contentions, the asserted claims as described in part below. Nothing in these contentions should be interpreted as an acquiescence to or assertion of a particular claim construction by Defendant.[2]

| Claim Language | Disclosure |
|---|---|
| 1. A method of utilizing a document stream operating system that in turn utilizes subsystems from at least one other operating system, comprising: | *See, e.g.*, p. 633. |
| receiving documents from diverse applications in formats that are specific to the respective applications and differ as between at least some of said applications; | *See, e.g.*, pp. 629, 632. |
| automatically associating time-based indicators with the documents received in the receiving step from the diverse applications; | *See, e.g.*, p. 631; Fig. 6. |
| automatically archiving the received documents; | |
| automatically creating glance views that are abbreviated versions of respective ones of said documents; | *See, e.g.*, p. 629. |
| selectively displaying at least some of said documents as a receding, foreshortened stack of partly overlapping documents so that only a part of each of said documents in the displayed stack, after the first document in the stack, is visible to the user; | *See, e.g.*, p. 629; Figs.1-2, 6. |
| said displaying further including displaying a cursor or pointer and responding to a user sliding the cursor or pointer over said displayed stack to display the glance view of the document in the stack that is currently touched by the cursor or pointer, without requiring clicking on the document; and | *See, e.g.*, Fig. 5(b). |

[2] To the extent Mander, Piles is argued by Plaintiff or found by the Court not to explicitly teach certain limitations in the asserted claims, such limitations would have been inherent and/or obvious as described in Defendant's Preliminary Invalidity Contentions.

| Claim Language | Disclosure |
|---|---|
| utilizing, in said document stream operating system, subsystems from said at least one other operating system for operations including writing documents to storage media, interrupt handling and input/output. | *See, e.g.*, p. 633. |
| | |
| 2. A method as in claim 1 including storing said documents as a main stream that is time-based and selectively generating a substream of documents that are a subset of the documents in the main stream matching selected criteria. | *See, e.g.*, p. 629; Fig. 6. |
| | |
| 3. A method as in claim 2 in which said generating a substream comprises generating a substream that persists unless selectively destroyed by a user. | |
| | |
| 4. A method as in claim 3 in which said generating a substream comprises generating a live substream that collects new documents that are added to said main stream and meet said criteria. | |
| | |
| 9. A method of automatically archiving documents received from diverse applications in different formats such that the archived documents can be searched for documents meeting selected criteria, comprising: | *See, e.g.*, pp. 629, 632. |
| receiving documents from diverse applications in formats that are specific to the respective applications and differ as between at least some of said applications; | *See, e.g.*, pp. 629, 632. |
| automatically associating time-based indicators with the documents received in the receiving step from the diverse applications; | *See, e.g.*, p. 631; Fig. 6. |
| automatically archiving the received documents together with said time-based indicators; | |
| selectively displaying at least some of said documents as a receding, foreshortened stack of partly overlapping documents so that only a part of each of said documents in the displayed stack, after the first document in the stack, is visible to the user; and | *See, e.g.*, p. 629; Figs.1-2, 6. |
| said displaying further including displaying a cursor or pointer and responding to a user | *See, e.g.*, Fig. 5(b). |

| Claim Language | Disclosure |
|---|---|
| sliding the cursor or pointer over said displayed stack to display a glance view of the document in the stack that is currently touched by the cursor or pointer, wherein said glance view is an abbreviated version of the documents. | |
| | |
| 10. A method as in claim 9, including utilizing subsystems from at least one other operating system for operations including writing documents to storage media and input/output in said archiving and displaying. | *See, e.g.*, p. 633. |
| | |
| 11. A method as in claim 9 including selectively searching said archived documents for documents meeting selected criteria and generating and displaying a substream comprising documents identified in said searching, said substream being in time order and comprising documents in different formats matching respective different applications from which the documents originated. | *See, e.g.*, pp. 628-629, 632; Fig. 6(a). |

# EXHIBIT 13C

# Invalidity of U.S. Pat. No. 6,725,427

## by Mander et al., A 'Pile' Metaphor for Supporting Casual Organization of Information, CHI '92

Based upon the claim interpretations Plaintiff appears to be asserting and the applications of those interpretations to Defendant's products in Plaintiff's Preliminary Infringement Contentions, Mander, Piles anticipates and/or renders obvious, alone or in combination with other prior art identified in Defendant's Preliminary Invalidity Contentions, the asserted claims as described in part below. Nothing in these contentions should be interpreted as an acquiescence to or assertion of a particular claim construction by Defendant.[3]

| Claim Language | Disclosure |
|---|---|
| 1. A stream-based operating system utilizing subsystems from another operating system running a computer, comprising: | *See, e.g.*, p. 633. |
| a document organizing facility receiving documents created by diverse applications in diverse formats specific to the respective applications; | *See, e.g.*, pp. 629, 632. |
| said document organizing facility automatically associating respective selected indicators with the received documents, automatically archiving the documents and indicators in consistent format for selective retrieval, and automatically creating information specifying respective glance views of said documents and respective document representations of said documents; | |
| a display facility displaying at least selected document representations as a receding, foreshortened stack of partly overlapping document representations such that only a part of each displayed document representation, after the first in the stack, is visible to the user; | *See, e.g.*, p. 629; Figs.1-2, 6. |
| said display facility further displaying a cursor or pointer and responding to user-controlled sliding without clicking of the cursor over said displayed stack to display a glance view of a document whose document representation is currently touched by the cursor or pointer; and | *See, e.g.*, Fig. 5(b). |

[3] To the extent Mander, Piles is argued by Plaintiff or found by the Court not to explicitly teach certain limitations in the asserted claims, such limitations would have been inherent and/or obvious as described in Defendant's Preliminary Invalidity Contentions.

| Claim Language | Disclosure |
|---|---|
| said stream-based operating system utilizing subsystems from said another operating system for operations including writing documents to storage media, interrupt handling, and input/output. | *See, e.g.*, p. 633. |
| | |
| 2. A stream-based operating system as in claim 1 in which said selected indicators are time-based. | *See, e.g.*, p. 631; Fig. 6. |
| | |
| 5. A stream-based operating system as in claim 1 in which said display of said glance view comprises an abbreviated version of the respective document. | *See, e.g.*, p. 629. |
| | |
| 7. A stream-based operating system as in claim 1 in which said display of a glance view comprises important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document. | *See, e.g.*, p. 629. |
| | |
| 8. A controlling operating system utilizing subsystems from another operating system running a computer, comprising: | *See, e.g.*, p. 633. |
| a document organizing facility receiving documents from diverse applications in diverse formats specific to the respective applications; | *See, e.g.*, pp. 629, 632. |
| said document organizing facility automatically associating selected indicators with the received documents, automatically archiving the documents and indicators in consistent format for selective retrieval, and automatically creating information specifying respective glance views of said documents and respective document representations of said documents; | |
| a display facility displaying at least selected ones of said document representations; | *See, e.g.*, p. 629; Figs. 1-2. |
| said display facility further displaying a cursor or pointer and responding to user-controlled sliding without clicking of the cursor or pointer over the displayed document representations to display at least a glance view of a document whose document representation is currently touched by the cursor or pointer; | *See, e.g.*, Fig. 5(b). |

| Claim Language | Disclosure |
|---|---|
| said controlling operating system utilizing subsystems from said another operating system for operations including writing documents to storage media, interrupt handling, and input/output. | *See, e.g.*, p. 633. |
| | |
| 9. An operating system as in claim 8 in which said selected indicators are time-based. | *See, e.g.*, p. 631; Fig. 6. |
| | |
| 10. An operating system as in claim 8 in which said display facility displays said document representations as a receding, foreshortened stack of partly overlapping document representations such that only a part each but the first document representation in the displayed stack is visible to a user. | *See, e.g.*, p. 629; Figs.1-2, 6. |
| | |
| 13. A stream-based operating system as in claim 8 in which said display of said glance view comprises an abbreviated version of the respective document. | *See, e.g.*, p. 629. |
| | |
| 15. A stream-based operating system as in claim 8 in which said display of a glance view comprises important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document. | *See, e.g.*, p. 629. |
| | |
| 16. A controlling operating system utilizing subsystems from another operating system running a computer, comprising: | *See, e.g.*, p. 633. |
| a document organizing facility associating selected indicators with received or created documents and creating information specifying glance views of the respective documents and information specifying document representations of the respective documents; | *See, e.g.*, p. 629; Figs. 1-2. |
| a display facility displaying at least selected ones of said document representations; | *See, e.g.*, p. 629; Figs. 1-2. |
| said display facility further displaying a cursor or pointer and responding to a user sliding without clicking the cursor or pointer over a portion of a displayed document representation to display the glance view of the document whose document | *See, e.g.*, Fig. 5(b). |

| Claim Language | Disclosure |
|---|---|
| representation is touched by the cursor or pointer; and | |
| said controlling operating system utilizing subsystems from said another operating system for operations including writing documents to storage media, interrupt handling and input/output. | *See, e.g.*, p. 633. |
| | |
| 17. An operating systems as in claim 16 in which said selected indicators are time-based. | *See, e.g.*, p. 631; Fig. 6. |
| | |
| 18. An operating system as in claim 16 in which said display facility displays said document representations as a receding, foreshortened stack of partly overlapping document representations such that only a part of most document representations in the displayed stack is visible to a user. | *See, e.g.*, p. 629; Figs.1-2, 6. |
| | |
| 19. An operating system as in claim 16 in which said document organizing facility receives said document in formats specific to heterogeneous applications and creates said information specifying said glance views to enable display of the glance views in a consistent format. | *See, e.g.*, pp. 629, 632. |
| | |
| 22. A stream-based operating system as in claim 16 in which said display of said glance view comprises an abbreviated version of the respective document. | *See, e.g.*, p. 629. |
| | |
| 24. A stream-based operating system as in claim 16 in which said display of a glance view comprises important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document. | *See, e.g.*, p. 629. |
| | |
| 25. A document stream operating system utilizing subsystems from another operating system running a computer, comprising: | *See, e.g.*, p. 633. |

| Claim Language | Disclosure |
|---|---|
| a document organizing facility associating chronological indicators with documents received from diverse applications in diverse formats and creating information specifying glance views of the respective documents and information specifying document representations of respective documents; | *See, e.g.*, pp. 629, 632; Figs. 1-2. |
| a display facility displaying at least selected ones of said document representations as a receding, foreshortened stack of partly overlapping document representations such that only a part each document representation except the first one in the displayed stack is visible to a user; | *See, e.g.*, p. 629; Figs.1-2, 6. |
| said display facility further displaying a cursor or pointer and responding to a user sliding without clicking the cursor or pointer over said displayed stack of document representations to display the glance view of the document whose document representation is currently touched by the cursor; and | *See, e.g.*, Fig. 5(b). |
| said document stream operating system utilizing subsystems from said another operating system for operations including writing documents to storage media, interrupt handling and input/output. | *See, e.g.*, p. 633. |
| | |
| 26. A document operating system as in claim 25 in which said document organizing facility associates said chronological indicators with documents at the time of receipt or creation of said documents without requiring a user to name the documents. | *See, e.g.*, p. 629. |
| | |
| 29. A stream-based operating system as in claim 25 in which said display of said glance view comprises an abbreviated version of the respective document. | *See, e.g.*, p. 629. |
| | |
| 31. A stream-based operating system as in claim 25 in which said display of a glance view comprises important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document. | *See, e.g.*, p. 629. |
| | |
| 32. A method of displaying heterogenous documents from different applications in a receding, foreshortened stack of selected document representations of said documents and providing a set of commands applicable to the document representations in the | *See, e.g.*, pp. 629, 632; Figs.1-2, 6. |

| Claim Language | Disclosure |
|---|---|
| stack, comprising: | |
| first displaying document representations of said documents received from different applications as a receding, foreshortened stack of partly overlapping document representations such that only a part of each of most document representations in the displayed stack is visible to the user; | *See, e.g.*, p. 629; Figs.1-2, 6. |
| said stack being in a time order related to respective time-based indicators automatically associated with the documents at the time of receipt or creation thereof; | *See, e.g.*, Fig. 6(a). |
| subsequently, while displaying the stack of document representations, responding automatically to touching a document representation in the stack with a user-operated cursor or pointer, without further action by the user, to display separately from the displayed stack of document representations, a glance view of the document whose document representation is currently touched by the cursor or pointer, said glance view being displayed while the displayed stack of document representations remains visible; and | *See, e.g.*, Fig. 5(b). |
| concurrently with displaying said glance view, displaying in the same display a set of command buttons, said command buttons being responsive to user clicks to cause respective operations to be performed on the document whose glance view is displayed at the time. | |
| | |
| 33. A method as in claim 32 in which said displaying of document representations comprises displaying at least the top line of each document whose document representation is displayed in the stack. | *See, e.g.*, p. 629. |
| | |
| 34. A method as in claim 32 including visually identifying attributes of selected documents in the displayed stack of document representations by markings that are visible in the displayed stack, each marking being common to a class of documents. | *See, e.g.*, p. 631. |
| | |
| 37. A stream-based operating system as in claim 32 in which said display of said glance view comprises an abbreviated version of the respective document. | *See, e.g.*, p. 629. |

| Claim Language | Disclosure |
| --- | --- |
| | |
| 39. A stream-based operating system as in claim 32 in which said display of a glance view comprises important words, pictures, and/or sounds of the respective document resulting from complex analysis of the document. | *See, e.g.*, p. 629. |

# EXHIBIT 13D

# Invalidity of U.S. Pat. No. 6,768,999

## by Mander et al., A 'Pile' Metaphor for Supporting Casual Organization of Information, CHI '92

Based upon the claim interpretations Plaintiff appears to be asserting and the applications of those interpretations to Defendant's products in Plaintiff's Preliminary Infringement Contentions, Mander, Piles anticipates and/or renders obvious, alone or in combination with other prior art identified in Defendant's Preliminary Invalidity Contentions, the asserted claims as described in part below. Nothing in these contentions should be interpreted as an acquiescence to or assertion of a particular claim construction by Defendant.[4]

| Claim Language | Disclosure |
|---|---|
| 1. A method of operating an enterprise information management system comprising at least one server and a number of personal computers selectively communicating with each other comprising: | *See, e.g.,* p. 633. |
| creating document object models comprising selected information from and about information assets of diverse types, created by diverse software, said document object models having a consistent structure; | *See, e.g.,* pp. 629, 632. |
| displaying browse cards related to respective ones of the information assets in a time-ordered stream, together with glance views related to the document object models of the respective displayed documents, said glance views being displayed essentially in real time in response to passing a cursor over respective ones of the browse cards. | *See, e.g.,* p. 629; Figs. 1-2, 5(b). |

[4] To the extent Mander, Piles is argued by Plaintiff or found by the Court not to explicitly teach certain limitations in the asserted claims, such limitations would have been inherent and/or obvious as described in Defendant's Preliminary Invalidity Contentions.