

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC and YOUTUBE, LLC
Petitioners

v.

ROBOCAST, INC.
Patent Owner

Case IPR2022-00590
Case IPR2023-00591
U.S. Patent No. 7,155,451 B1
Issue Date: December 26, 2006

Title: AUTOMATED BROWSING SYSTEM FOR PUBLISHERS AND USERS
ON NETWORKS SERVING INTERNET AND REMOTE DEVICES

DECLARATION OF CHRISTOPHER M. SCHMANDT

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

TABLE OF CONTENTS

	Page
I. INTRODUCTION AND QUALIFICATIONS	9
A. Qualifications and Experience	9
B. Materials Considered.....	13
II. PERSON OF ORDINARY SKILL IN THE ART	15
III. STATEMENT OF LEGAL PRINCIPLES.....	18
A. Claim Construction	18
B. Obviousness.....	20
IV. THE '451 PATENT.....	26
A. Overview of the Specification.....	26
B. The Challenged Claims	29
V. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS	29
A. Brief Summary and Overview of Prior Art.....	31
1. References Common to All Grounds.....	31
(a) Shimizu (EX1003).....	31
(b) Benedict (EX1006).....	40
(c) Fox (EX1005).....	44
2. Additional References.....	46
(a) Sagman (EX1006)	46
(b) Filepp (EX1007).....	49
(c) Guinan (EX1008).....	53
(d) Robertson (EX1009).....	56
(e) Miller (EX1010)	62
(f) Ackermann (EX1012) and Resource Kit (EX1013).....	64
(g) Schloss (EX1011).....	68

TABLE OF CONTENTS
(continued)

	Page
B. Ground 1: Claims 1-8 Are Obvious Over Shimizu in view of Benedict and Fox.....	71
1. Independent Claim 1	71
(a) “creating a show structure of nodes, each node identifying a resource from a plurality of accessible resources;” (Claim 1[a])	76
(b) “without requiring user input, automatically accessing a plurality of said resources each of said resources identified by each of said nodes; and” (Claim 1[b])	98
(c) “displaying a content corresponding to each of said resources automatically in accordance with said show structure, wherein said step of creating further comprises the step of providing an interactively variable duration information, representing the duration within which a corresponding content to said resource is being displayed so as to enable a user to vary said duration.” (Claim 1[c])	103
2. Claim 2	109
3. Claim 3	110
4. Claim 4	111
5. Claim 5	115
6. Claim 6	117
7. Claim 7	118
8. Claim 8	120
C. Ground 2: Claims 10-12 Are Obvious Over Shimizu in view of Benedict, Fox, and Sagman.....	123
1. Independent Claim 10	123

TABLE OF CONTENTS

(continued)

	Page
(a) “identifying at least one resource from a plurality of accessible resources via said communications network, wherein each of said resources is associated with a corresponding icon displayed on a first section of said user’s terminal;” (Claim 10[a]).....	125
(b) “creating a show structure of nodes, each node identifying a corresponding resource by recording a sequence of said resources by dragging and dropping said icons on a second section of said user’s terminal;” (Claim 10[b])	130
(c) “without requiring user input, automatically accessing each of said resources identified by each of said nodes in accordance with said show structure;” (Claim 10[c])	140
(d) “automatically retrieving a content corresponding to each of said accessed resources; and” (Claim 10[d])	140
(e) “displaying each of said retrieved contents automatically in accordance with said show structure.” (Claim 10[e]).....	140
2. Claim 11	141
3. Claim 12	142
D. Ground 3: Claims 13-20 Are Obvious Over Shimizu in view of Benedict, Fox, Sagman, and Filepp	143
1. Claim 13	143
2. Claim 14	149
3. Claim 15	156
4. Claim 16	158
5. Claim 17	158
6. Claim 18	160

TABLE OF CONTENTS
(continued)

	Page
7. Claim 19	160
8. Claim 20	161
E. Ground 4: Claims 22-25 Are Obvious Over Shimizu in view of Benedict, Fox, and Guinan	162
1. Independent Claim 22	162
(a) “performing an on-line search, in response to which a plurality of search results and their corresponding URL resources are received;” (Claim 22[a])	164
(b) “automatically creating and storing a show structure of nodes, each node identifying a resource from a plurality of accessible resources via said communications network wherein said show structure is created in response to said search results received in response to said on-line search;” (Claim 22[b])	172
(c) “without requiring user input accessing each of said resources identified by each of said nodes;” (Claim 22[c])	174
(d) “retrieving a content corresponding to each of said accessed resources; and” (Claim 22[d])	174
(e) “automatically delivering said content to said user for display on said user’s computer in accordance with said show structure.” (Claim 22[e])	174
9. Claim 23	175
10. Claim 24	176
11. Claim 25	176
F. Ground 5: Claims 26-32, 35-36 Are Obvious Over Shimizu in view of Benedict, Fox, Sagman, Guinan, and Filepp	177
1. Claim 26	177
2. Claim 27	180

TABLE OF CONTENTS
(continued)

	Page
3. Claim 28	180
4. Claim 29	181
5. Claim 30	181
6. Claim 31	182
7. Claim 32	182
8. Claim 35	183
9. Claim 36	186
G. Ground 6: Claims 37-38 Are Obvious Over Shimizu in view of Benedict, Fox, and Robertson	187
1. Independent Claim 37	187
(a) “creating a multidimensional show structure of nodes, each node identifying a resource from a plurality of accessible resources;” (Claim 37[a])	188
(b) “without requiring user input accessing each of said resources identified by each of said nodes wherein at least two of said nodes are spanned concurrently;” (Claim 37[b])	204
(c) “automatically displaying a content corresponding to each of said resources in accordance with said show structure, wherein contents of said at least two nodes are displayed during an overlapping time period.” (Claim 37[c])	207
2. Claim 38	208
H. Ground 7: Claims 39-44 Are Obvious Over Shimizu in view of Benedict, Fox, and Miller	209
1. Independent Claim 39	209
(a) “selecting a category from a predefined list of categories;” (Claim 39[a])	211

TABLE OF CONTENTS

(continued)

	Page
(b) “performing an on-line search related to said selected category, in response to which a plurality of search results and their corresponding URL resources are received;” (Claim 39[b]).....	216
(c) “automatically creating and storing a show structure of nodes, each node identifying a resource from a plurality of accessible resources via said communications network wherein said show structure is created in response to said search results received in response to said on-line search;” (Claim 39[c])	217
(d) “without requiring user input retrieving each of said resources identified by each of said nodes; and” (Claim 39[d]).....	217
(e) “automatically delivering said content to said user for display on said user’s computer in accordance with said show structure.” (Claim 39[e])	217
2. Claim 40	218
3. Claim 41	219
4. Claim 42	220
5. Claim 43	221
6. Claim 44	222
I. Ground 8: Claim 9 is Obvious Over Shimizu in view of Benedict, Fox, Robertson, and Sagman	223
J. Ground 9: Claim 21 is Obvious Over Shimizu, Benedict, Fox, and Filepp, in further view of Robertson and Sagman	230
K. Ground 10: Claims 33-34 Are Obvious Over Shimizu in view of Benedict, Fox, Guinan, Filepp, in further view of Robertson and Sagman	231
L. Ground 11: Claim 40 is Obvious Over Shimizu in View of Benedict, Fox, and Miller, in Further View of Schloss	232

TABLE OF CONTENTS
(continued)

	Page
M. Ground 12: Claim 45 is Obvious Over Shimizu in view of Benedict, Fox, and Miller, in further view of Ackermann and Resource Kit	237
VI. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS.....	245
VII. CONCLUSION.....	247

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

I, Christopher M. Schmandt, declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

A. Qualifications and Experience

1. I have recently retired from my position as a Principal Research Scientist at the Media Laboratory at Massachusetts Institute of Technology (“M.I.T”), after 40 years of employment by M.I.T. In that role, I also served as faculty for the M.I.T. Media Arts and Sciences academic program. I have more than 40 years of experience in the field of Media Technology, and was a founder of the M.I.T. Media Laboratory.

2. I received my Bachelor of Science degree in Electrical Engineering and Computer Science from M.I.T. in 1978, and my Master of Science degree in Visual Studies (Computer Graphics), also from M.I.T. I had been employed at M.I.T. since 1980, initially at the Architecture Machine Group, which was an early computer graphics research lab. In 1985, I helped found the Media Laboratory and continued to work there until my retirement. I ran a research group titled “Living Mobile.” My research spanned distributed communication and collaborative systems, with an emphasis on multi-media and user interfaces; I have more than 70 published conference and journal papers and one book in these fields.

3. In my faculty position, I taught courses and directly supervised student

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

research and theses at the Bachelors, Masters, and Ph.D. level. I oversaw the Masters and Ph.D. thesis programs for the entire Media Arts and Sciences academic program. Based on the above experience and qualifications, I have a solid understanding of the knowledge and perspective of a person of ordinary skill in this technical field since at least 1980.

4. In the course of my nearly 40 years of research at MIT, I conceived of and helped implement, with my students, a number of hyper-media systems which are analogous to the invention claimed in the '451 patent. The earliest of these, Gaze Orchestrated Dynamics, dated from 1982, and provided an eye gaze driven user interface to browse a network of linked video snippets; this included managing and synchronizing a separate digital audio track for simultaneous presentation as a navigational cue.

5. I built a number of systems oriented toward voice access to information, often over a telephone connection. In these cases, the content consisted of text and/or audio material, with most voice user interfaces providing a serialized default path through the content. Consider a unified voice / text message system (Voiced Mail, 1982 and Phone Slave, 1983). A visual user interface to email typically presents headers for all the new messages simultaneously, and a user clicks on what they wish to hear. But when the content is audio, the interaction is by necessity

serialized, with the system providing a default path of reading or playing messages in some order. Of course in this situation the user maintains some control over the presentation, and can always change the time spent on each message by interrupting (the equivalent to clicking on a control button in a GUI). Additionally, as we learned that users quickly gain facility in understanding synthetic speech, each user could independently control the playback speed, and hence time on each message, of both the synthetic and recorded audio content.

6. Although any individual presentation in such systems follows a given path, the content nodes are in fact a network, and can be traversed along different paths. For example, a user may request only urgent messages, in which a new path is calculated to present those messages first, or messages from a particular sender, or a particular date.

7. Another example of a complex network of nodes was non-real-time direction giving (Direction Assistance, 1987) in which the content nodes were literally each street intersection in Cambridge, MA, with the content consisting of instructions for a driver. This system was built so a user could take notes while listening, before starting to drive, and as we learned that people take notes at very different speeds, voice control of the pause between each piece of directions was added, for flexible flow. Systems such as these also exhibit search, in that the user

specifies a destination address, and the system calculates a route (i.e. a path through the intersections) for presentation. We also incorporated much of the same logic into one of the earliest real-time spoken driving direction systems (Back Seat Driver, 1988) in which temporal presentation takes on the added burden of calculating the proper time to speak based on driving speed, and sometimes needs to merge two direction turns into a single presentation to give adequate warning.

8. Hyperspeech (1993) used a more conventional hypertext model, only each content node was an audio recording by a particular speaker. The recordings were collected automatically by the system phoning the talent, asking questions, and recording answers. A user's path through the linked content recordings was then chosen according to a question, or series of questions, a Hyperspeech caller would ask. I also worked with the Unix cron facility, which is used for scheduling tasks to occur at specified times. I also extended the crontab file functionality to support recurring events, and built speech and graphical user interfaces around the resulting extended crontab functionality (Caltalk, 1990).

9. The above are representative examples of projects from my research group which involved hypertext-like systems analogous to the '451 patent for navigating through a set of information using one or more paths.

10. My Curriculum Vitae is attached as **Exhibit A**.

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

11. I have been retained by counsel for Petitioners to provide my expert opinion in connection with the above-captioned proceeding as set forth herein.

B. Materials Considered

12. The analysis that I provide in this Declaration is based on my education, research, and experience, as well as the documents I have considered. In forming my opinions, I have read and considered U.S. Patent No. 7,155,451 B1 (“451 patent”) (EX1001) and its prosecution history. I have cited to the following documents in my analysis below:

Exhibit No.	Description of Document
1001	U.S. Patent No. 7,155,451 B1 to Damon C. Torres (filed Sep. 1, 1998, issued Dec. 26, 2006) (“451” or “451 patent”)
1003	U.S. Patent No. 5,634,062 to Takeshi Shimizu et al. (filed October 20, 1994, issued May 27, 1997) (“Shimizu”)
1004	Excerpts from Robert C. Benedict, Jr., <i>Using Harvard Graphics 2 for Windows</i> (1993) (“Benedict”)
1005	Excerpts from David Fox & Troy Downing, <i>HTML Web Publisher’s Construction Kit</i> (1995) (“Fox”)
1006	Excerpts from Stephen W. Sagman, <i>Running Microsoft PowerPoint 4 for Windows</i> (1994) (“Sagman”)

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Exhibit No.	Description of Document
1007	U.S. Patent No. 5,347,632 to Robert Filepp et al. (filed July 28, 1989, issued September 13, 1994) (“Filepp”)
1008	Catherine Guinan et al., <i>Information Retrieval from Hypertext Using Dynamically Planned Guided Tours</i> , Proceedings of the ACM Conference on Hypertext (Nov. 1992), pp.122-130 (“Guinan”)
1009	U.S. Patent No. 6,486,895 B1 to George G. Robertson et al. (filed September 8, 1995, issued November 26, 2002) (“Robertson”)
1010	Excerpts from Michael Miller, <i>Using CompuServe</i> (2d ed. 1994) (“Miller”)
1011	U.S. Patent No. 5,706,507 to Robert J. Schloss (filed July 5, 1995, issued January 6, 1998) (“Schloss”)
1012	Excerpts from Ernest Ackermann et al., <i>Using Unix</i> (1994) (“Ackermann”)
1013	Excerpts from Microsoft Press, <i>Microsoft Office for Windows 95 Resource Kit</i> (1995) (“Resource Kit”)
1020	Excerpts from <i>Microsoft Press Computer Dictionary</i> (2d ed. 1994)
1023	Claim Construction Order issued in <i>Robocast, Inc. v. Microsoft Corp.</i> , No. 1:10-cv-01055 RGA (D. Del.) and <i>Robocast, Inc. v. Apple Inc.</i> , No. 1:11-cv 00235 RGA (D. Del.), dated June 28, 2013
1024	Prosecution history of U.S. Patent No. 7,155,451 B1
1025	Prosecution history of U.S. Patent No. 8,606,819 B2

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Exhibit No.	Description of Document
1026	Prosecution history of U.S. Patent No. 8,965,932 B2
1027	Prosecution history for U.S. Patent Appl. Serial No. 08/922,063
1028	Prosecution history for U.S. Provisional Patent Appl. Serial No. 60/025,360

II. PERSON OF ORDINARY SKILL IN THE ART

13. I understand that, under the patent laws in effect before the America Invents Act (“AIA”) of 2011, an assessment of claims of a patent filed before the AIA took effect should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date (i.e., the “time the invention was made”). I have preliminarily for purposes of my Declaration only assumed that date to be September 3, 1996, the filing date of the earliest application to which the ’451 patent claims priority (provisional application no. 60/025,360).¹

14. I have also been advised that to determine the appropriate level of a

¹ I express no opinion on whether the ’451 patent is actually entitled to the benefit of the provisional application to which it claims priority (60/025,360). This issue is not relevant here because all of the prior art references relied upon predate that date. In the event entitlement to the provisional filing date may later become relevant, for example based on arguments later presented by Patent Owner, I reserve my right to provide opinions on that issue at that time.

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

person having ordinary skill in the art, the following factors may be considered: (1) the types of problems encountered by those working in the field and prior art solutions thereto; (2) the sophistication of the technology in question, and the rapidity with which innovations occur in the field; (3) the educational level of active workers in the field; and (4) the educational level of the inventor.

15. The '451 patent states that it generally relates to “computer networks, and specifically to a method of sequencing and scheduling web pages, via a suite of software applications.” ('451, 1:16-18.)

16. In my opinion, a person of ordinary skill in the art as of September 1996 would have possessed a bachelor's degree in electrical engineering, computer science, or similar field, with at least two years of experience in developing and implementing network-based computer systems, such as systems for storing and retrieving information over the Internet or communicating using the Web. A person could also have qualified as a person of ordinary skill in the art with some combination of (1) more formal education (such as a master's of science degree) and less technical experience, or (2) less formal education and more technical or professional experience. I am also informed that, in an earlier *inter partes* review proceeding involving related U.S. Patent No. 8,965,932, Patent Owner proposed the following articulation of the level of ordinary skill in the art:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

A person of ordinary skill in the art at the time of the invention would have the equivalent of a four-year degree from an accredited institution (usually denoted as a B.S. degree) in computer science, computer engineering or the equivalent, and experience with, or exposure to, topics in computer networking, multimedia, and web systems. A person of ordinary skill in the art would also have approximately 3-5 years of professional experience with these topics. Additional graduate education could substitute for professional experience, while significant experience in the field might substitute for formal education.

For purposes of my analysis, Patent Owner's proposed articulation is not meaningfully different from mine, and therefore my analysis, opinions and conclusions in my Declaration would not change if Patent Owner's articulation were applied.

17. My opinions regarding the level of ordinary skill in the art are based on, among other things, my over 40 years of experience in the field of computer science, my understanding of the basic qualifications that would be relevant to an engineer or scientist tasked with investigating methods and systems in the relevant area, and my familiarity with the backgrounds of colleagues, co-workers, and employees, both past and present.

18. Although my qualifications and experience exceed those of the hypothetical person having ordinary skill in the art defined above, my analysis and

opinions regarding the '451 patent have been based on the perspective of a person of ordinary skill in the art as of September 1996.

III. STATEMENT OF LEGAL PRINCIPLES

A. Claim Construction

19. I understand that a purpose of claim construction is to determine what a person of ordinary skill in the art would have understood the claim terms to mean. Claim terms are generally given their ordinary and customary meaning, which is the meaning that the term would have to a person of ordinary skill in the art in question as of the effective filing date.

20. I understand that the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification. I understand that the patent specification, under the legal principles, has been described as the single best guide to the meaning of a claim term, and is thus highly relevant to the interpretation of claim terms. And I understand for terms that do not have a customary meaning within the art, the specification usually supplies the best context of understanding the meaning of those terms.

21. I further understand that other claims of the patent in question, both asserted and unasserted, can be valuable sources of information as to the meaning of

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

a claim term. Because the claim terms are normally used consistently throughout the patent, the usage of a term in one claim can often illuminate the meaning of the same term in other claims. Differences among claims can also be a useful guide in understanding the meaning of particular claim terms.

22. I understand that the prosecution history can further inform the meaning of the claim language by demonstrating how the inventors understood the invention and whether the inventors limited the invention in the course of prosecution, making the claim scope narrower than it otherwise would be. Extrinsic evidence, such as dictionaries, may also be consulted in construing the claim terms.

23. I understand that, in *Inter Partes* Review (IPR) proceedings, a claim of a patent shall be construed using the same claim construction standard that would be used to construe the claim in a civil action filed in a U.S. district court (which I understand is called the “*Phillips*” claim construction standard), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.

24. I have been instructed by counsel to apply the “*Phillips*” claim construction standard for purposes of interpreting the claims in this proceeding, to the extent they require an explicit construction. The description of the legal

principles set forth above thus provides my understanding of the “*Phillips*” standard as provided to me by counsel.

25. I understand that certain terms appearing in the challenged claims have been the subject of district court claim constructions in previous litigations involving Apple Inc. and Microsoft Corporation. I have reviewed a document called “**Claim Construction**,” dated June 28, 2013, which states that it is a claim construction order from the District of Delaware issued in *Robocast, Inc. v. Microsoft Corp.*, No. 10-1055 RGA and *Robocast, Inc. v. Apple Inc.*, No. 11-235 RGA, which provides constructions for various terms recited in the challenged patents. I will identify those constructions in my analysis in **Part V** below. For purposes of my analysis here, I do not believe express claim constructions are necessary because the prior art renders the claims obvious under the claim constructions adopted by the district court or any other reasonable construction.

B. Obviousness

26. I understand that a patent claim is obvious if, as of the effective filing date, it would have been obvious to a person having ordinary skill in the field of the technology (the “art”) to which the claimed subject matter belongs.

27. I understand that the following factors should be considered in analyzing obviousness: (1) the scope and content of the prior art; (2) the differences

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

between the prior art and the claims; and (3) the level of ordinary skill in the pertinent art. I also understand that certain other facts known as “secondary considerations” such as commercial success, unexplained results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others may be utilized as indicia of nonobviousness. I understand, however, that secondary considerations should be connected, or have a “nexus,” with the invention claimed in the patent at issue.

28. I understand that a reference qualifies as prior art for obviousness purposes when it is analogous to the claimed invention. The test for determining what art is analogous is: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the inventor’s endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved.

29. I understand that a person of ordinary skill in the art is assumed to have knowledge of all prior art. I understand that one skilled in the art can combine various prior art references based on the teachings of those prior art references, the general knowledge present in the art, or common sense. I understand that a motivation to combine references may be implicit in the prior art, and there is no requirement that there be an actual or explicit teaching to combine two references.

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Thus, one may take into account the inferences and creative steps that a person of ordinary skill in the art would employ to combine the known elements in the prior art in the manner claimed by the patent at issue. I understand that one should avoid “hindsight bias” and ex post reasoning in performing an obviousness analysis. But this does not mean that a person of ordinary skill in the art for purposes of the obviousness inquiry does not have recourse to common sense.

30. I understand that when determining whether a patent claim is obvious in light of the prior art, neither the particular motivation for the patent nor the stated purpose of the patentee is controlling. The primary inquiry has to do with the objective reach of the claims, and that if those claims extend to something that is obvious, then the entire patent claim is invalid.

31. I understand one way that a patent can be found obvious is if there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent’s claims. I understand that a motivation to combine various prior art references to solve a particular problem may come from a variety of sources, including market demand or scientific literature. I understand that a need or problem known in the field at the time of the invention can also provide a reason to combine prior art references and render a patent claim invalid for obviousness. I understand that familiar items may have obvious uses beyond their

primary purpose, and that a person of ordinary skill in the art will be able to fit the teachings of multiple prior art references together like the pieces of a puzzle. I understand that a person of ordinary skill is also a person of at least ordinary creativity. I understand when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this finite number of predictable solutions leads to the anticipated success, I understand that the invention is likely the product of ordinary skill and common sense, and not of any sort of innovation. I understand that the fact that a combination was obvious to try might also show that it was obvious, and hence invalid, under the patent laws. I understand that if a patent claims a combination of familiar elements according to known methods, the combination is likely to be obvious when it does not more than yield predictable results. Thus, if a person of ordinary skill in the art can implement a predictable variation, an invention is likely obvious. I understand that combining embodiments disclosed near each other in a prior art reference would not ordinarily require a leap of inventiveness.

32. I understand that obviousness may be shown by demonstrating that it would have been obvious to modify what is taught in a single piece of prior art to create the patented invention. Obviousness may also be shown by demonstrating

that it would have been obvious to combine the teachings of more than one item of prior art. I understand that a claimed invention may be obvious if some teaching, suggestion, or motivation exists that would have led a person of ordinary skill in the art to combine the invalidating references. I also understand that this suggestion or motivation may come from the knowledge of a person having ordinary skill in the art, or from sources such as explicit statements in the prior art. I understand that when there is a design need or market pressure, and there are a finite number of predictable solutions, a person of ordinary skill may be motivated to apply common sense and his skill to combine the known options in order to solve the problem.

33. I understand the following are examples of approaches and rationales that may be considered in determining whether a piece of prior art could have been combined with other prior art or with other information within the knowledge of a person having ordinary skill in the art:

- (1) Some teaching, motivation, or suggestion in the prior art that would have led a person of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- (2) Known work in one field of endeavor may prompt variations of it for use in the same field or a different field based on design incentives or other

market forces if the variations would have been predictable to a person of ordinary skill in the art;

(3) Combining prior art elements according to known methods to yield predictable results;

(4) Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;

(5) Applying a technique or approach that would have been “obvious to try” (choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success);

(6) Simple substitution of one known element for another to obtain predictable results; or

(7) Use of a known technique to improve similar products, devices, or methods in the same way.

34. I understand that, when determining whether a claimed combination is obvious, the correct analysis is not whether one of ordinary skill in the art, writing on a blank slate, would have chosen the particular combination of elements described in the claim. Instead, I understand the correct analysis considers whether one of ordinary skill, facing the wide range of needs created by developments in the field of endeavor, would have seen a benefit to selecting the combination claimed.

35. I understand that the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference. The test for obviousness, in other words, is not whether the references could be physically combined but whether the claimed inventions are rendered obvious by the teachings of the prior art as a whole.

IV. THE '451 PATENT

A. Overview of the Specification

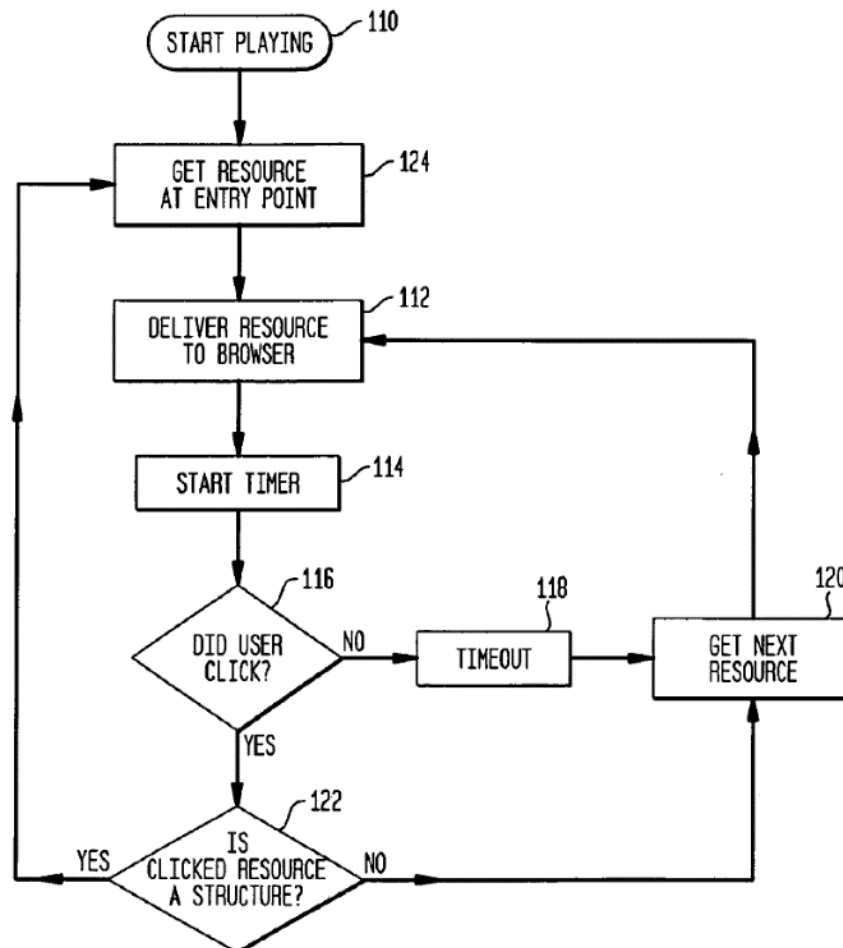
36. The '451 patent, entitled "Automated Browsing System for Publishers and Users on Networks Serving Internet and Remote Devices," states that it "relates generally to computer networks, and specifically to a method of sequencing and scheduling web pages, via a suite of software applications." ('451, 1:16-18.)

37. The '451 patent acknowledges that the Internet was already well-known and had "grown tremendously" with the popularization of the World Wide Web. ('451, 1:22-25.) The patent explains that various technology standards relating to the Web were popularized including Hypertext Transfer Protocol ("HTTP"), Hypertext Markup Language ("HTML") and Uniform Resource Locators ("URLs"). ('451, 1:35-37.) Using these and other technologies, the patent states that web browser software provided an environment that "proved to be very popular" and led to the web's "mass adoption." ('451, 1:57-60.) The patent further

acknowledges known web technologies for a user to automatically receive personalized information updates both by selecting content sources from a list and by using collected user interest profiles. ('451, 2:10-15, 2:23-28.)

38. The '451 patent describes methods for a user to be automatically presented with content of a series of resources, such as web pages, using what the patent calls a “show structure.” ('451, 2:53-57.) A “show structure” contains a set of nodes, each node “represent[ing] a resource such as a web page, an executable file, or a data file stored at a predetermined location and accessed via its corresponding address.” ('451, 2:55-60.) Each node includes both an address from which the content may be accessed and an indication of the duration for which the content will be presented. ('451, 3:26-31.) A show structure defines one or more paths through the set of nodes. ('451, 3:4-6.) Figure 4 of the '451 patent, reproduced below, outlines how a user may either “on-the-fly” interactively navigate the arrangement of resources specified by the show structure (*e.g.*, steps 116 and 122) or, much like traditional broadcast television or an automatic slideshow, “simply watch[] a series of pre-edited resources” (*e.g.*, steps 114-118):

FIG. 4
RoboSurf



(’451, Fig. 4; see also id., 3:13-16, 11:8-40.) The ’451 patent explains that a show structure itself can be generated in a number of ways, including manually by a user collecting URLs, or automatically in connection with providing results to a user’s search query. (’451, 8:27-31, 11:66-12:8.) I discuss additional aspects of the ’451 patent in my analysis of the claims below.

B. The Challenged Claims

39. This Declaration addresses claims 1-45 of the '451 patent. Independent claim 1 is representative and recites:

1. A method for displaying on a user's computer, content derived from a plurality of resources in an organized arrangement comprising the steps of:
 - [a] creating a show structure of nodes, each node identifying a resource from a plurality of accessible resources;
 - [b] without requiring user input, automatically accessing a plurality of said resources each of said resources identified by each of said nodes; and
 - [c] displaying a content corresponding to each of said resources automatically in accordance with said show structure, wherein said step of creating further comprises the step of providing an interactively variable duration information, representing the duration within which a corresponding content to said resource is being displayed so as to enable a user to vary said duration.

('451, 18:25-39 (Claim 1; bracketed notation (e.g., [a]) added).)

40. I address the claims further in my detailed analysis in **Part V** below.

V. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS

41. I have reviewed and analyzed the prior art references and materials listed in **Part I.B** above. In my opinion the claims of the '451 patent would have

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

been obvious to a person of ordinary skill in the art based on the following combinations of the prior art.

Ground	Claims	Basis for Challenge Under § 103
1	1-8	Shimizu, Benedict, Fox
2	10-12	Shimizu, Benedict, Fox, Sagman
3	13-20	Shimizu, Benedict, Fox, Sagman, Filepp
4	22-25	Shimizu, Benedict, Fox, Guinan
5	26-32, 35-36	Shimizu, Benedict, Fox, Guinan, Filepp
6	37-38	Shimizu, Benedict, Fox, Robertson
7	39-44	Shimizu, Benedict, Fox, Guinan, Miller
8	9	Ground 1 Prior Art, with Robertson and Sagman
9	21	Ground 3 Prior Art, with Robertson and Sagman
10	33-34	Ground 5 Prior Art, with Robertson and Sagman
11	40	Ground 7 Prior Art, with Schloss
12	45	Ground 7 Prior Art, with Ackermann and Resource Kit

42. I am informed by counsel that each of the references cited in the grounds above qualifies as prior art to the challenged claims because each reference was filed and/or published before the earliest filing date to which the '451 patent can claim priority (September 3, 1996). I also understand that the grounds listed above may be numbered differently from the numbering in the IPR petitions filed by

Petitioner, as I am informed that they may be split among multiple petitions.

A. Brief Summary and Overview of Prior Art

1. References Common to All Grounds

(a) Shimizu (EX1003)

43. **Shimizu**, U.S. Patent No. 6,534,062, is entitled “System for Managing Hypertext Node Information and Link Information.” I am informed that Shimizu qualifies as prior art because it was filed on October 20, 1994, which is before the earliest priority date of the challenged claims.

44. As its title indicates, Shimizu discloses a system for managing information about nodes and links in a “hypertext” system. “Hypertext” generally refers to a way of organizing and presenting information in which information is presented with links or references (commonly referred to as “hyperlinks”) to other information. Shimizu explains that “[h]ypertext apparatuses were first developed as a system capable of holding information in units called nodes, of displaying character information held by the nodes, and of linking a plurality of nodes as desired.” (Shimizu, 1:17-20.) As Shimizu observes, although hypertext systems could present textual information, they also supported many other types of information such as images, sounds, and motion pictures. (Shimizu, 1:23-29.)

45. Hypertext systems were well-known since at least the 1980s and, by the

priority date, were exceedingly popular. An early example of a well-known and commercially available hypertext-based system was “HyperCard” from Apple Computer, Inc., which allowed users to organize information into virtual “stacks” of “cards,” which could be hyperlinked. Another example of a “hypertext” system common by the mid-1990s – and no doubt the most widely known example by the priority date – was the World Wide Web. The Web provides an example of a hypertext system because web pages can present text, graphics, and/or other types of information and can include hyperlinks to other pages. (Fox, EX1005, pp.8-10 (describing the Web as an example of a hypertext system).)

46. Figure 1 of Shimizu depicts an exemplary hypertext system with a number of nodes and links:

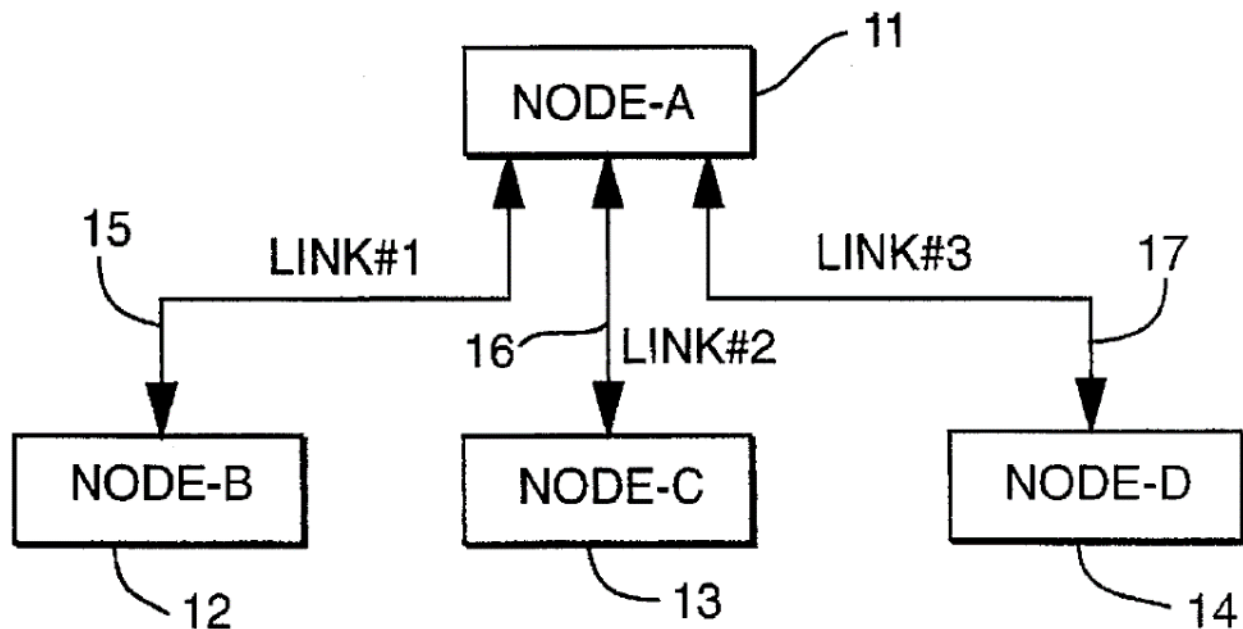


FIG. 1

(Shimizu, Fig. 1.) The hypertext arrangement depicted in Figure 1 shows Node-A through Node-D (11, 12, 13, and 14) and links 15, 16, and 17. Node-A 11 has three links 15, 16, and 17 to Node-B, Node-C, and Node-D, respectively. (Shimizu, 1:56-60.) “Link information is held by each node.” (Shimizu, 1:61.) “In the example above, the access from node A 11 is made first to node B 12 then to node C 13 and to node D 14, in the order in which the first link information 21 (link #1), second link information 22 (link #2), and third link information 23 (link #3) are read out consecutively.” (Shimizu, 2:21-26.)

47. Shimizu focuses on techniques for managing and keeping track, in a

hypertext system, of (a) node information and (b) link information. (Shimizu, 1:8-15.) Node information can include “not only characters but also monochromatic and color images, sounds and motion pictures.” (Shimizu, 1:23-28.) Two key components of Shimizu, for purposes of my analysis, are node information management part 5 and link information management part 6 shown in Figure 10:

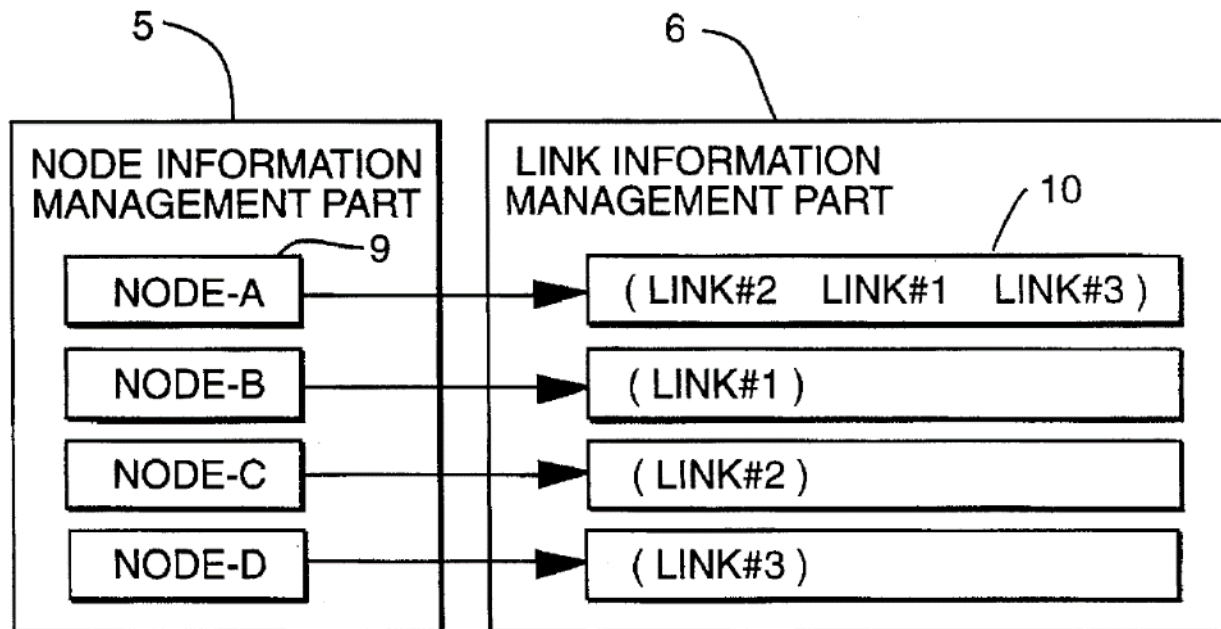


FIG. 10

(Shimizu, Fig. 10.) Node information management part 5, shown on the left above, “manages a plurality of node information structures holding the information about each node.” (Shimizu, 6:31-33.) Node information management part 5 manages node information structures for each node, each structure specifying information about the node. Figure 6 shows an exemplary node information structure 9:

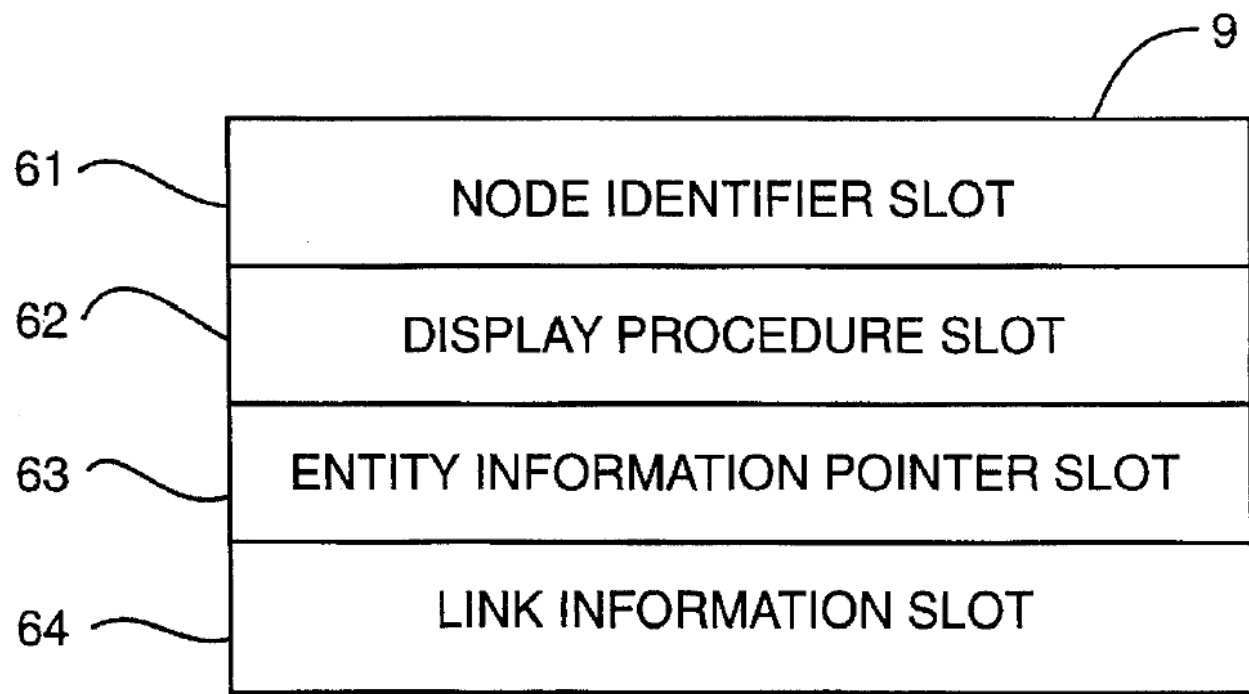


FIG. 6

(Shimizu, Fig. 6.) Node information structure 9 holds a number of members including, as shown, node identifier slot 61, display procedure slot 62, entity information pointer slot 63, and link information slow 64. As Shimizu explains:

The node identifier slot 61 is used to identify uniquely this node information structure within the system. The display procedure slot 62 retains the pointer pointing to a procedure program function used when the node in question is to be displayed on the screen. The entity information pointer slot 63 retains the pointer pointing to the entity information to be displayed by the node in question. The entity information thus pointed to may be a character string, image

information, motion pictures or sounds. The link information slot **64** retains, in a list format data constitution, the pointer pointing to the link information structure for holding the link information.

(Shimizu, 6:50-62.)

48. Figure 10 above also shows, on the right, link information management part **6**, which “manages a plurality of link information structures defining the relations between the nodes.” (Shimizu, 6:33-35.) As its name implies, the link information structure specifies information about a link associated with a node. Figure 7 shows a “link information structure managed by the link information management part **6**.”

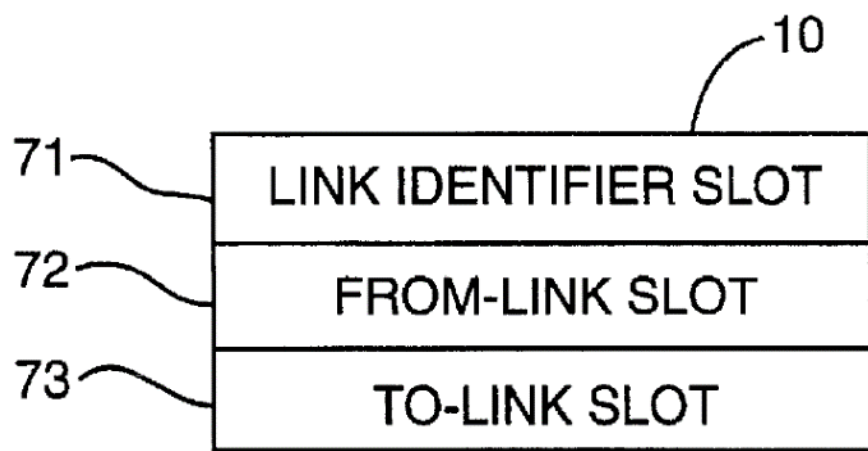


FIG. 7

(Shimizu, Fig. 7.) Link information structure **10** as shown in Figure 7 includes a

link identifier slot **71**, a from-link slot **72**, and a to-link slot **73**. Shimizu explains:

The link identifier slot **71** is used to identify uniquely this link information structure within the system. The from-link slot **72** retains the pointer pointing to the node information structure of, say, the node A when a link is set illustratively to link the node A to the node B. The to-link slot **73** retains the pointer pointing to the node information structure of the node B when, as in the example above, a link is set to link the node A to the node B.

(Shimizu, 7:1-9.)

49. Shimizu goes on to explain how the system uses the node and link information structures discussed above to present the nodes to the user in a particular order as indicated by the link information. Figure 8 shows a user interface screen for presenting node content to the user:

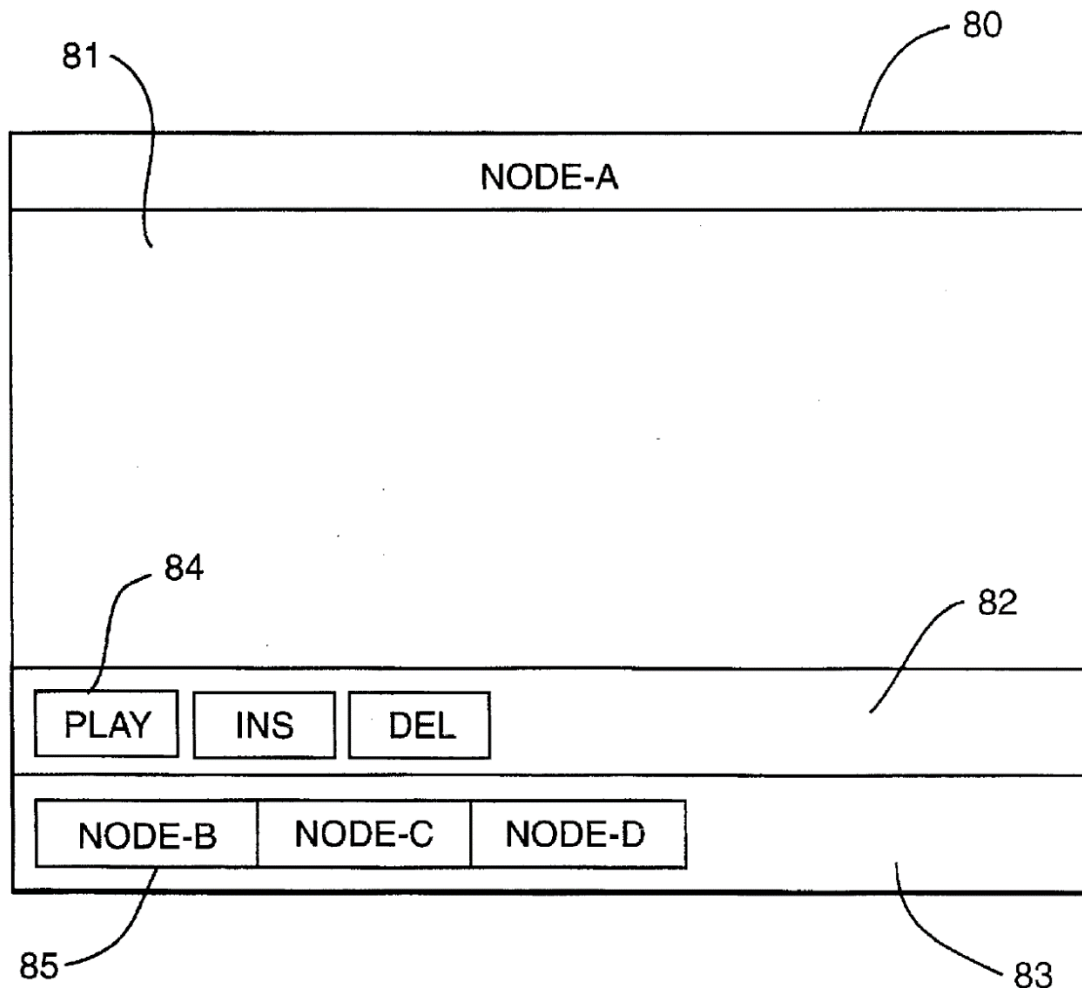


FIG. 8

(Shimizu, Fig. 8, 7:12-14 (“FIG. 8 is a view of a typical user interface screen display provided by an application program running on the embodiment.”).) When the user presses PLAY in Figure 8 (one of buttons **84**), the content associated with the current node (in this case NODE-A) is displayed in area **81** of window **80**. (Shimizu, 8:11-25, 8:49-59.) Figure 8 also includes display area **83** that “displays, in a left-justified manner and according to the link information order, icons **85** representing each of

the nodes linked starting from the current node.” (Shimizu, 7:30-33.) The node content displayed to the user, as mentioned, can include textual character strings, images, sounds, and motion pictures. (Shimizu, 8:52-59.)

50. Shimizu further discloses that node content is accessed and presented on the user’s computer in a continuous sequential fashion, one node after another, without requiring any input from the user beyond an initial click of the PLAY button. “The contents of the linked nodes are displayed in the order set by the link information.” (Shimizu, 8:15-16; *see also id.*, 8:67-9:4 (“In step **105** [of Figure 11], the next node information is reproduced in accordance with the next link information. The processing above is carried out so that the node C Is displayed followed by node B and node D, in accordance with the order of the link information.”).)² As Shimizu further explains, nodes can be displayed and switching to the next node can occur at predetermined intervals:

As described, the hypertext apparatus of the invention offers a user interface capable of putting into a desired order the links between the nodes to be managed and of manipulating the established link order. This makes it easy to create illustratively an application program

² Unless otherwise noted, all underlining appearing in quotes from prior art references was added for emphasis.

allowing the display of node contents to be switched in a desired order
and at predetermined intervals on the hypertext apparatus.

(Shimizu, 9:60-66.) I will provide more information about Shimizu in my discussion
of the claim limitations below.

(b) Benedict (EX1006)

51. **Benedict** is a book entitled *Using Harvard Graphics Version 2 for Windows*. I am informed that Benedict qualifies as prior art because it was published before the earliest priority date of the challenged claims.

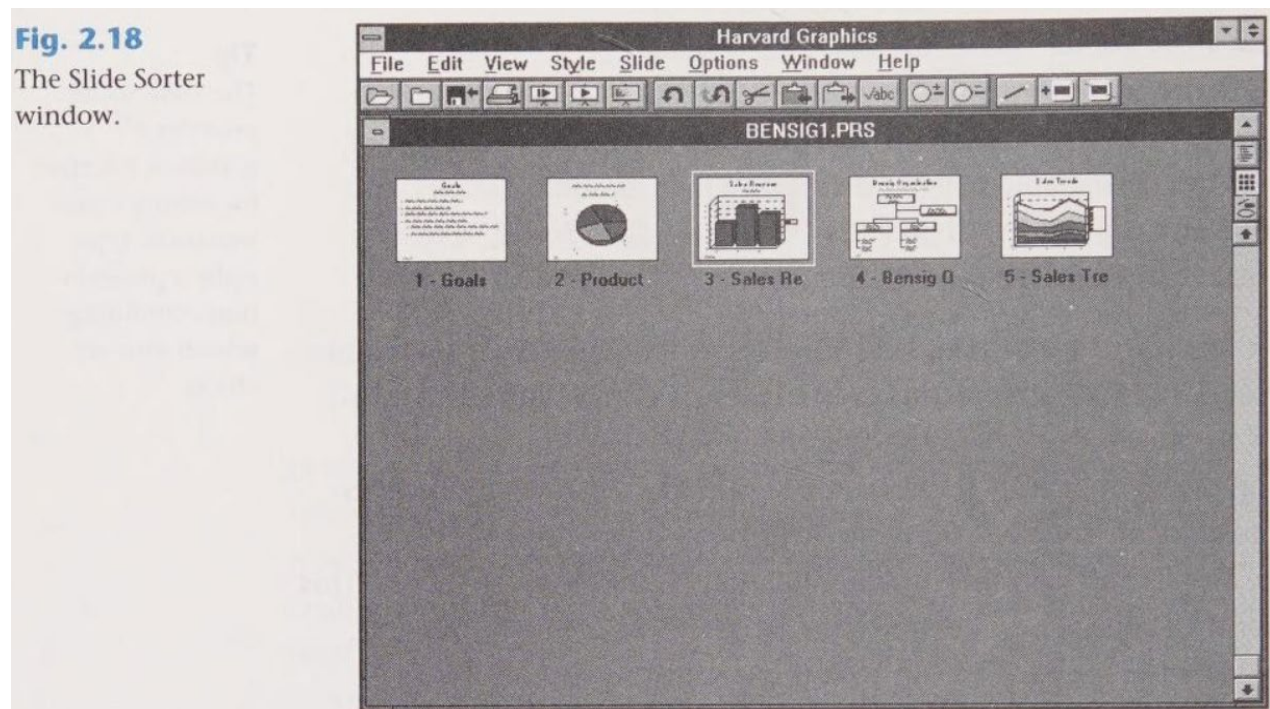
52. Benedict describes a software program known as Harvard Graphics that provided functionality similar, in some respects, to well-known business presentation programs such as Microsoft PowerPoint. (Benedict, p.1 (“Harvard Graphics for Windows is a program designed for creating business presentations.”).) Harvard Graphics allowed users to create a “presentation” that included a series of slides. (Benedict, p.24 (“A *presentation* is a collection of slides stored in computer memory in a file on your hard disk.”) (italics in original).)

53. Benedict also discloses features for viewing a presentation (referred to as a “ScreenShow”), for example, by displaying slides in a specified sequential order. “As you create and enhance presentations, you can use the ScreenShow feature to evaluate your work. ScreenShows are on-screen presentations in which

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

the program displays each slide in order.” (Benedict, p.19.) Benedict further discloses a feature known as “HyperShows” in which slides in a ScreenShow can include hypertext-like links allowing the user to jump to other slides in the ScreenShow via on-screen hyperlinks. (Benedict, pp.442-453.)

54. Benedict describes a number of features for creating presentations, including adding new slides and specifying their content. (Benedict, pp.32-42.) Benedict also describes a “Slide Sorter” feature that displays icons for each slide in the presentation and allows the user to manipulate the order in which they will appear in the presentation:



(Benedict, p.44, Fig. 2.18; *see also id.*, p.43 (“Figure 2.18 shows the Slide Sorter

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

displaying five slides. The Slide Sorter represents these slides as scaled down images.”).) The exemplary presentation above in Figure 2.18 includes five numbered and ordered slides (e.g., “1 - Goals,” “2 - Product,” “3 - Sales Revenue,” etc.). The user can reorder slides in a presentation by simply clicking on a slide with the mouse and dragging the slide to a new position. (Benedict, p.44.) “When you release the mouse button, the slide appears in its new location, and the other slides adjust to make room for the relocated slide.” (Benedict, p.44.)

55. Benedict discloses additional features (known as “ScreenShow Effects”) that allow the user to customize the display of the presentation. (Benedict, pp.438-442.) As shown in Figure 17.1 of Benedict, one of these ScreenShow Effects features allows the user to specify how long a particular slide in a presentation will be displayed before moving to the next slide:

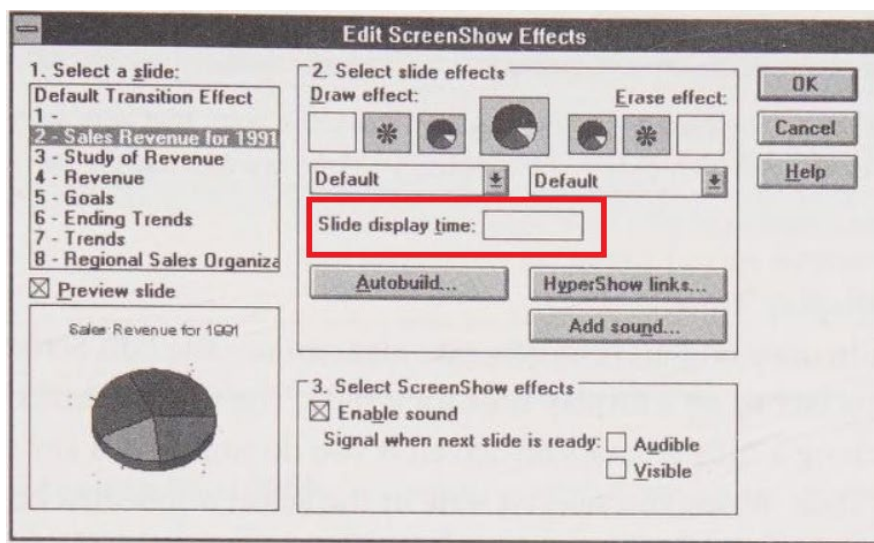


Fig. 17.1
The Edit
ScreenShow
Effects dialog box.

(Benedict, p.439, Fig. 17.1 (red box added).) The portion in the red box, as shown above, allows the user to specify how long the selected slide will be displayed:

The display time determines how long a slide remains on-screen if you do not press a key to move to the next slide...

The default for the time is blank because Harvard Graphics continues to display the slide until you press a key. You can set a display time, however, by typing a time in the Slide display time: text box. You can enter the time in seconds or in minutes and seconds separated by a colon. Entering 23, for example, sets the display time to 23 seconds; entering 1:30 sets the display time to 1 minute and 30 seconds.

(Benedict, p.440 (italics in original).) Benedict explains that, using this feature, the user can create “self-running” presentations in which slides advance automatically after a specified period, without requiring user action:

Creating Self-Running Presentations

Self-running presentations automate your ScreenShow by displaying the slides for a specified length of time before the next slide appears—pressing a key is not necessary. At the end of the presentation, a HyperShow link causes the ScreenShow to start over from the beginning. The presentation continues until you press the Escape key. To create a self-running presentation, you must set the display time for the slides and create the link that begins the presentation again.

(Benedict, p.453 (boldface in original).)

56. I will provide more details about Benedict in my discussion of the claim limitations below.

(c) Fox (EX1005)

57. **Fox** is a book entitled *HTML Web Publisher's Construction Kit*. I am informed that Fox qualifies as prior art because it was published before the earliest priority date of the challenged claims.

58. Fox covers a number of topics relating to the World Wide Web and how it works, including how to navigate the Web using Internet web browsers and how to create customized websites. (Fox, pp.xxiv-xxv.) Fox explains that, as is well-known today, “[a] Web page can be anything you imagine it to be: a hypertext document, graphics, sound, animation, tabular data, software files, a discussion group, a search index, a specialized program, and more.” (Fox, p.8; *see also id.*, p.9 (“The World Wide Web uses hypertext to link all its pages together.”).) Fox explains, as was well-known to ordinarily skilled artisans, that each resource on the web is identified by a Uniform Resource Locator (URL). (Fox, p.13; *see also id.*, pp.326-327 (describing IMG tag for identifying an image file).)

59. My Declaration relies on Fox primarily to demonstrate that it would have been obvious to implement the hypertext system described in Shimizu, along with the additional features of Benedict upon which I rely in this Declaration, in the

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

form of a web-based hypertext application. For example, Fox confirms that web-based techniques can be used to create sequential presentations akin to a slide show, similar to Shimizu and Harvard Graphics. For example, Fox discloses a technique in which a web page, after a specified amount of time, can cause the web browser to “refresh” and load another web page, which can be used to create slide show-type presentations of web content:

To load up a different document after a set amount of seconds, simply specify a full URL using the URL attribute. For example, to load the page at *http://www.smartpants.edu/sample.html*, after a minute, use the following markup as the first line of your HTML document:

```
<META HTTP-EQUIV="Refresh" CONTENT=60  
URL= http://www.smartpants.edu/sample.html>
```

You can use this trick to flip between two documents, animate a sequence of images, or create an automatic slide show of many Web pages.

(Fox, p.316 (italics in original; underlining added).) The concept of “refreshing” a web page was well-known to those familiar with web browsers, and was a common feature of many websites. Refreshing was commonly used to load an updated version of the *currently displayed* web page, but as shown above, the <META> tag used for this feature can specify the URL of any web page, including a different web page. In the example from Fox above, the <META> tag tells the web browser to

“refresh” after 60 seconds by switching to a different web page located at <http://www.smartpants.edu/sample.html>. I will provide more detail about Fox in my discussion of the claim limitations below.

2. Additional References

(a) Sagman (EX1006)

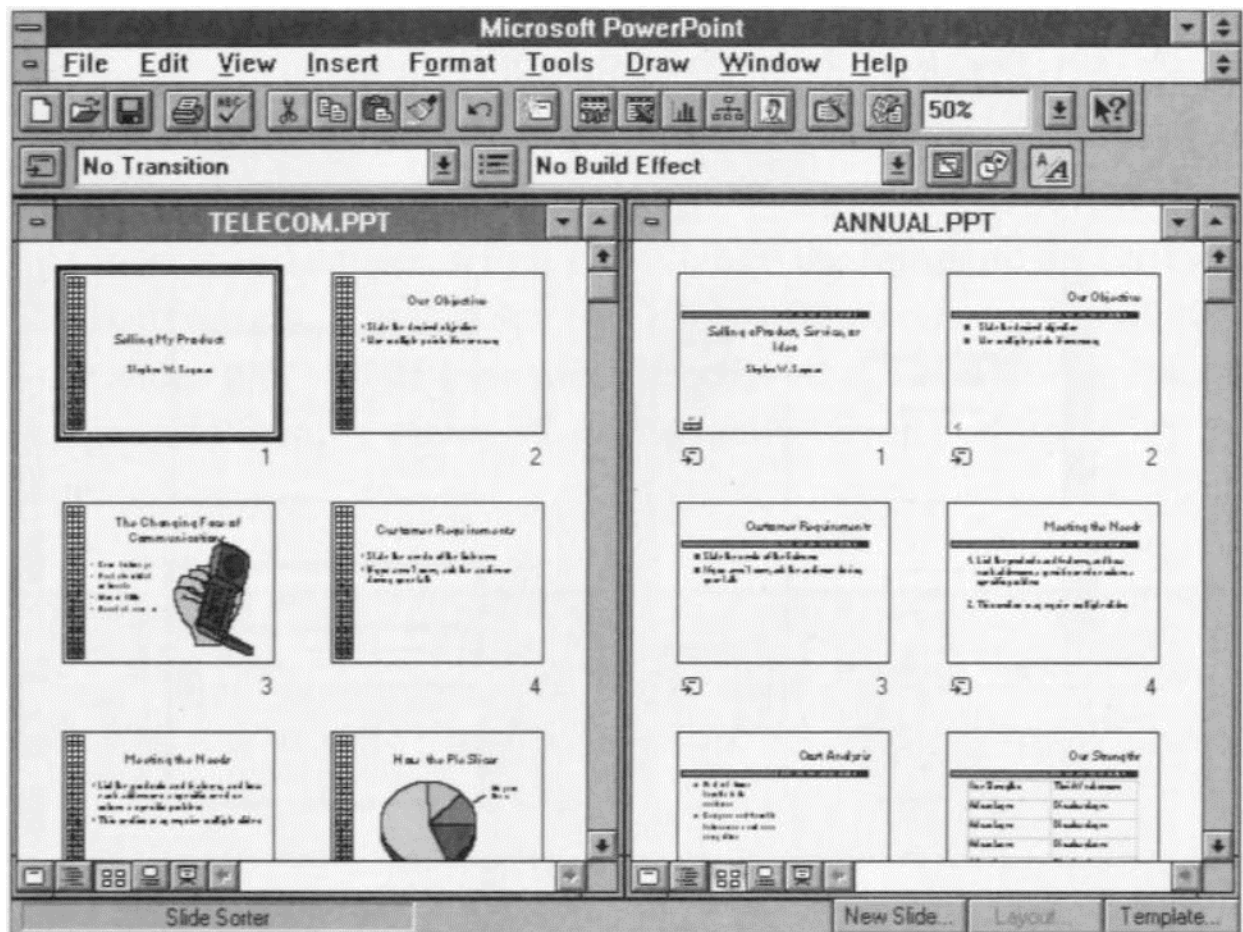
60. Sagman is a book entitled *Running Microsoft PowerPoint 4 for Windows*. I am informed that Sagman qualifies as prior art because it was published before the earliest priority date of the challenged claims.

61. Sagman describes a software program known as Microsoft PowerPoint that provided functionality similar, in some respects, to the Harvard Graphics product described above. PowerPoint allowed users to create a presentation including a series of slides. (Sagman, pp.6-7.) My Declaration relies on Sagman primarily for independent claim 11, which recites the creation of a show structure of nodes by recording a sequence of resources “**by dragging and dropping said icons on a second section of said user’s terminal.**” My Declaration also relies on Sagman for the “**pausing said display of said data files**” feature covered by dependent claims 9, 21, and 33-34.

62. For the former feature, I have cited the “Slide Sorter” feature of Sagman which is similar to some respects to the Harvard Graphics “Slide Sorter” feature

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

described by Benedict above. But the Slide Sorter feature in Sagman has an additional feature, *i.e.*, the ability to show two side-by-side presentations on the screen and to copy slides from one presentation to another using drag-and-drop:



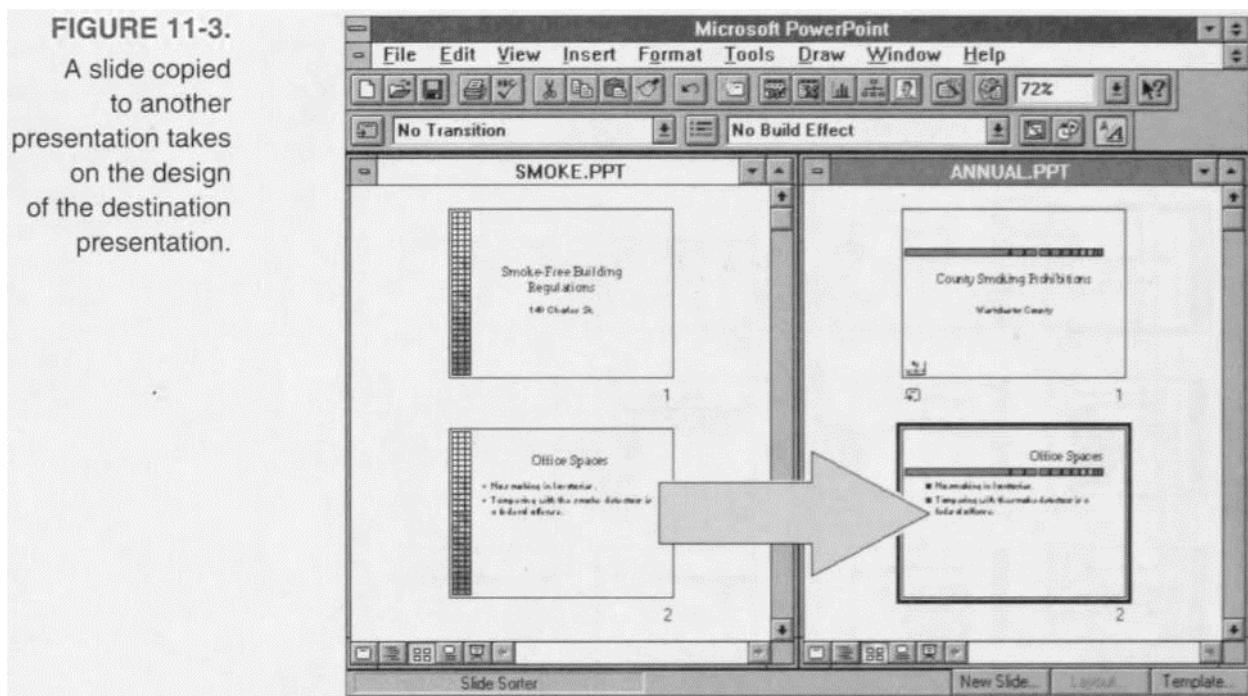
(Sagman, p.311.) The screen display above shows two PowerPoint presentations (TELECOM.PPT and ANNUAL.PPT) displayed side-by-side in two windows, with icons representing the individual slides in each presentation. Sagman explains that, using standard drag-and-drop techniques, the user can drag a slide from one

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

presentation into the other:

After the presentations are arranged side by side, you can move a slide from one presentation to another by clicking on a slide in one presentation and dragging it to the other presentation. The moved slide takes on the design of the destination presentation's template. To copy a slide rather than move it, hold down the Ctrl key as you drag the slide to the destination presentation. In Figure 11-3 on the next page, the bottom slide has been copied from one presentation to another. Note its appearance in each presentation.

(Sagman, p.313.) Figure 11-3, referenced above, shows the result of copying a slide from one presentation into another:



(Sagman, Fig. 11-3, p.312.) In this example, the slide “Office Spaces” was copied

from SMOKE.PPT into the ANNUAL.PPT presentation. As shown, the slide in the target presentation retains the same content as the original slide but takes on the design of the target presentation.

63. As noted, I have also relied on Sagman for the pausing feature of dependent claims 9, 21, and 33-34. Similar to Benedict and Robertson, Sagman also discloses the ability to automatically advance and display the next slide after a user-specified duration. (Sagman, p.391 & Fig. 16-2.) Sagman further discloses a keyboard shortcut for pausing the automatic advancement of slides. (Sagman, p.395, Table 16-1 (“Pause/resume automatic show”).)

64. I will provide additional detail about Sagman in my discussion of the claims below.

(b) Filepp (EX1007)

65. **Filepp**, U.S. Patent No. 5,347,632, is entitled “Reception System for an Interactive Computer Network and Method of Operation.” I am informed that Filepp qualifies as prior art because it was filed on July 28, 1989 and issued on September 13, 1994, before the earliest priority date of the challenged claims.

66. My Declaration cites Filepp primarily for limitations of certain dependent claims reciting display of “**a persistent content window**” (claims 14, 26), “**an advertising message**” (claims 16, 28), and displaying content “**in accordance**

with [a] user's profile" (claims 17, 29, 36). Filepp discloses an "interactive computer system that enables a user to display desired information, such as news, financial and cultural information, and perform desired transactional services, such as banking and shopping, through any of a plurality of types of personal computers." (Filepp, Abstract.) Filepp was originally assigned to Prodigy Services Company, an early provider of interactive on-line services that competed with companies such as America Online (AOL) before widespread adoption of the Web.

67. The interactive system in Filepp displays "pages" to the user, with each page divided into a series of partitions as shown in Figure 3a:

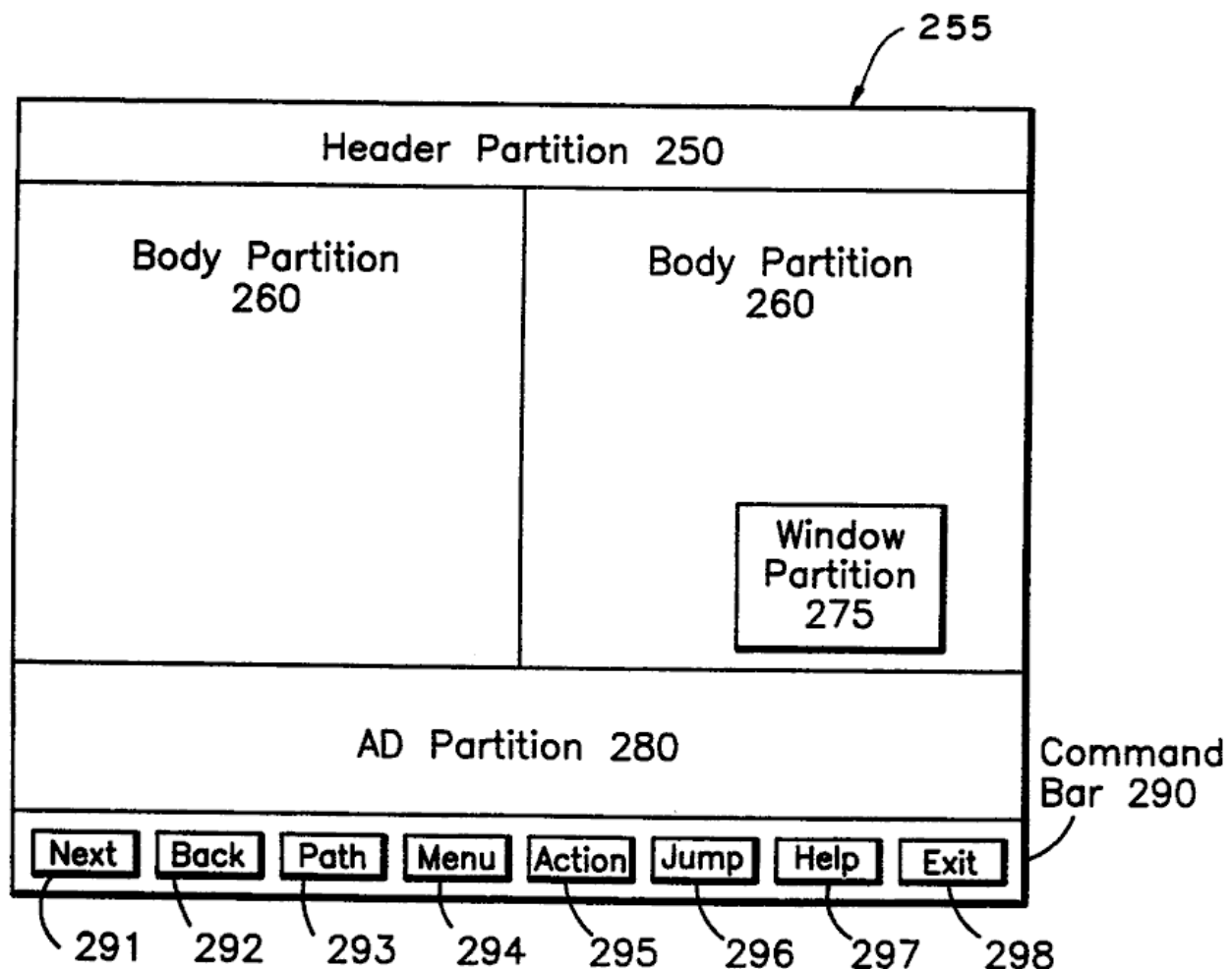


FIG. 3a

(Filepp, Fig. 3a.) “With reference to FIG. 3a, each page **255** is formatted into page partitions **250**, **260**, **280**, and **290** (not to be confused with applications partitions).”

(Filepp, 8:68-9:3.) Header partition **250** “typically conveys information on the page’s topic or sponsor,” and body partitions **260** and window partition **275** “gives the information and transactional content of the page.” (Filepp, 9:10-16.) AD

partition **280**, as its name implies, displays advertisements provided over the network. (Filepp, 9:27-30, 9:30-34 (“Advertisements **280** may be presented to the user on an individual basis from queues of advertisements that are constructed off-line by business system **130**, and sent to file server **205** where they are accessible to each RS **400**.”), 12:64-66.) AD partition **280** provides an example of a “**persistent content window**” because it provides a persistent window on the screen devoted to the display of a particular type of content (*i.e.* advertising).

68. With respect to the particular advertisements that can be displayed to the user, they may be selected based on user profile information such as information about the user’s past activities or its demographic information, ZIP code, etc.:

Individual queues of advertisements are constructed based upon data collected on the partitioned applications that were accessed by a user, and upon events the user generated in response to applications. The data are collected and reported by RS **400** to a data collection co-application in file server **205** for later transmission to business system **130**. In addition to application access and use characteristics, a variety of other parameters, such as user demographics or postal ZIP code, may be used as targeting criteria. From such data, queues of advertisements are constructed and targeted to either individual users or to sets of users who fall into certain groups according [sic; to] such parameters.

(Filepp, 9:35-47.)

69. I will provide more information about Filepp in my discussion of the claim limitations below.

(c) Guinan (EX1008)

70. **Guinan** is an article entitled “Information Retrieval from Hypertext Using Dynamically Planned Guided Tours.” I am informed that Guinan qualifies as prior art to the challenged claims because it was published before the earliest priority date of the challenged claims.

71. My Declaration relies on Guinan primarily for independent claim 22, which recites the step of “**performing an on-line search, in response to which a plurality of search results and their corresponding resources are received,**” and similar limitations in independent claim 39. As explained in more detail below, Guinan discloses a technique for (a) allowing a user to enter a natural language search query and then (b) searching for hypertext nodes in response to the query to create a “guided tour” for the user. (Guinan, pp.124-125.)

72. Guinan explains “[i]n using any hypertext system an end-user or browser will encounter many technical problems that have been well-documented elsewhere,” such as “disorientation and the quality of the information retrieved by the user from the hypertext.” (Guinan, p.122, §1.) This problem with hypertext was well-known and, as acknowledged in Fox, one of the key reasons the industry

adopted on-line searching in the Web context. (Fox, pp.10-11 (“Hypertext makes it easy to get lost... [W]hat if you want to find pages dealing with a specific topic? Luckily, the Web is full of indexes. An index is a special Web page that allows you to type in a search term or keyword. The index then spits out a list of Web pages matching your search.”).)

73. Guinan states that, to address problems of hypertext systems:

We propose a method for dynamically generating a guided tour through a hypertext in response to a user’s query which will plan a tour through a hypertext covering all the nodes the system believes relevant to the user’s information need as expressed in a natural language query, which eases disorientation problems by using the guided tour metaphor... and which takes advantage of typing of information links in the hypertext to present the nodes on the tour in a logical fashion. Effectively we extend the guided tour metaphor by including the information retrieval features... and we add in logical sequencing of nodes on the tour to complete the process.

(Guinan, p.123, §1 (Introduction).)

74. The first step in the proposed method is for the user to enter a “natural language” search query. (Guinan, p.124, §3.) The hypertext nodes are then searched for keywords matching the query, which are scored using a weighting algorithm. (*Id.*; see also *id.*, p.129, §6 (“*Finding information*: When the user enters a search

query in the form of a natural language statement it is matched to the nodes in the hypertext using standard information retrieval techniques.”) (*italics in original*.) Guinan explains that, in weighting hypertext nodes, search terms that appear less frequently throughout the collection are given more weight, and as such, hypertext nodes containing more of these less frequent terms are given higher weights. (*Id.*) Hypertext nodes are also weighted based on the number of connections they have to other nodes. (*Id.*) Hypertext nodes must satisfy a threshold value to be included in the tour. (Guinan, p.125 (“All nodes whose weight is greater than or equal to this cut-off or threshold value would be selected for inclusion in the tour.”).)

75. Once matching hypertext nodes have been identified, a guided tour is assembled. “At this stage, the nodes for the guided tour have been dynamically chosen in response to a user’s query and some form of logical ordering has to be imposed on the selected tour nodes.” (Guinan, p.125, §3.) Guinan discloses an ordering procedure that takes into account linking relationships between the matching nodes. (*Id.*) “This procedure is followed until every node selected for inclusion in the tour has been positioned correctly relative to each other node in the tour.” (*Id.*)

76. Guinan identifies several advantages of her guided tour technique. For example, “[u]sers don’t need to worry about getting lost or confused in the hypertext

as all they need to do is follow the tour which is generated in response to their query.”
(Guinan, p.129, §6.) I will provide more information about Guinan in my discussion
of claim 22 below.

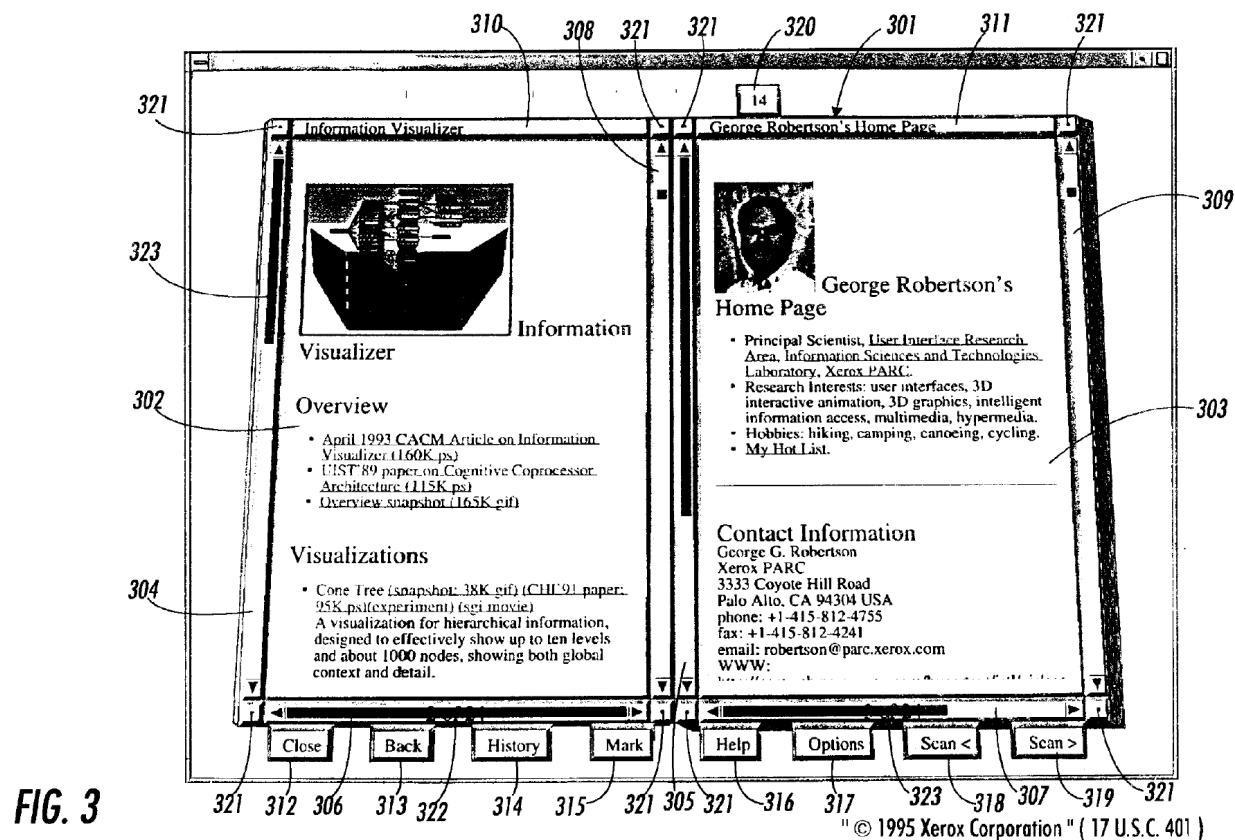
(d) Robertson (EX1009)

77. **Robertson**, U.S. Patent No. 6,486,895 B1, is entitled “Display System
for Displaying Lists of Linked Documents.” I am informed that Robertson qualifies
as prior art because it was filed on September 8, 1995, before the earliest priority
date of the challenged claims.

78. My Declaration cites Robertson primarily for two purposes. First, my
Declaration cites Robertson for limitations of claim 37 reciting that “**at least two of
said nodes are spanned concurrently**” and “**said at least two nodes are
displaying during an overlapping time period.**” Second, my Declaration cites
Robertson for certain “**control button**” features in dependent claims 9, 21, and 33.

79. Robertson discloses “[a] display system for displaying lists of linked
documents,” in which “[a]n example of linked documents are hyper-text documents
in a hyper-text system,” such as the Apple HyperCard system. (Robertson, 3:24-
27.) Robertson explains that his currently preferred embodiment uses “lists of linked
documents obtained from the portion of the Internet known as the World Wide Web
(hereafter the Web).” (Robertson, 2:33-36.)

80. Robertson discloses a technique for displaying a web-based book that presents the web pages in sequence. (Robertson, Abstract, 2:14-25.) “The list of pages is displayed in a book metaphor, termed WebBook, on a computer controlled display system... When displayed, two active pages are displayed at one time.” (Robertson, 2:24-28.) Figure 3 of Robertson shows an exemplary WebBook view showing two web pages visible in a side-by-side fashion:



(Robertson, Fig. 3.) As explained in Robertson:

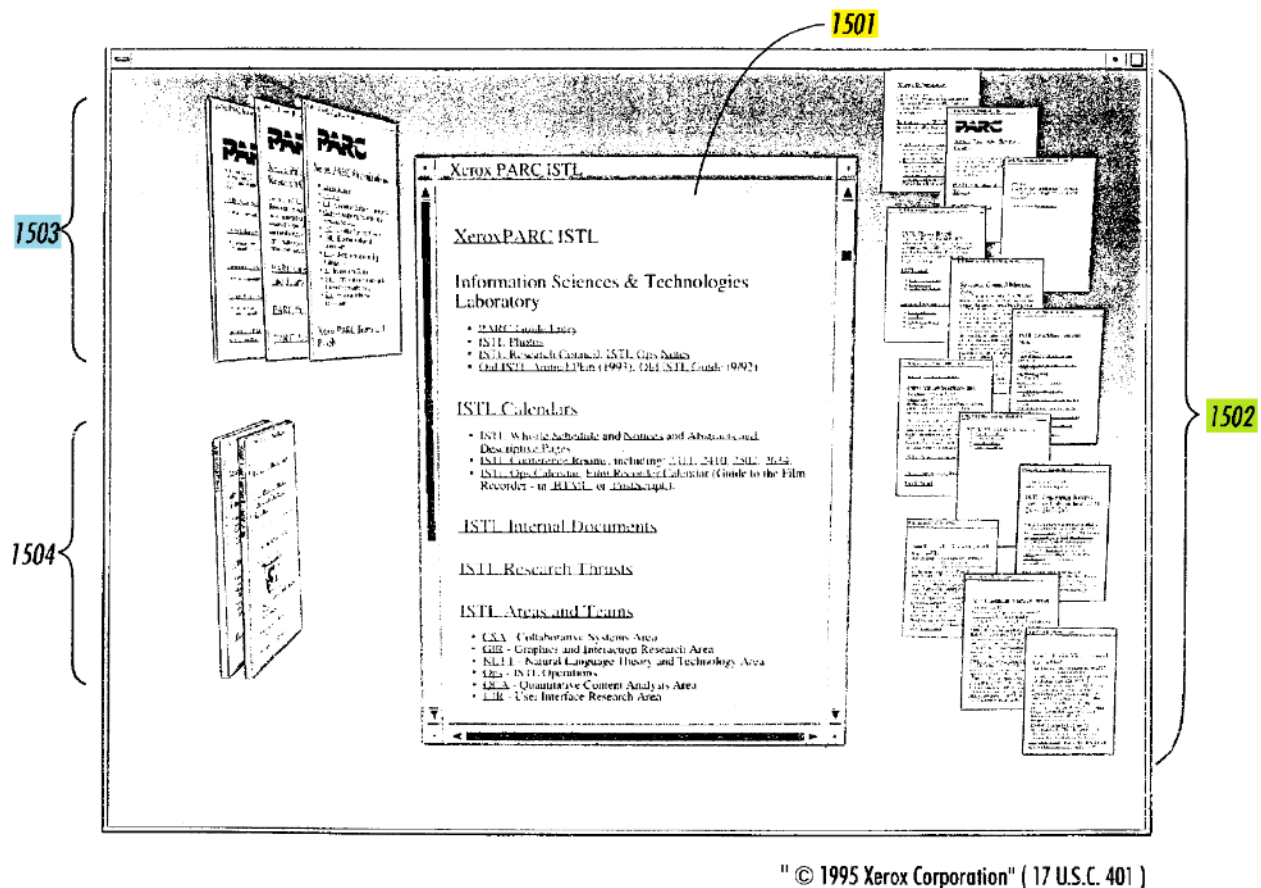
FIG. 3 is an illustration of a WebBook as displayed on a display.
Referring to FIG. 3, the WebBook 301 is comprised of a first page 302

and a second page **303**, each having an identical structure. The first page **302** is an instance of a left page and the second page **303** is an instance of a right page.

(Robertson, 6:15-19; *see also id.*, 6:28-30 (“The content and layout of the page is determined by the HTML macros defining the page.”).) Robertson explains that, in order to display web pages in a WebBook, each Web page is converted into a page object. (Robertson, 7:65-66.) “Each page is provided in the order in which it is to appear in the WebBook. Accordingly, each page object will represent either a left page or a right page in the WebBook.” (Robertson, 7:66-8:2.)

81. Robertson is also not limited to simply viewing two pages at the same time. Another embodiment shown in Figure 15 shows the ability to view “thumbnails” of potentially all of the pages in a WebBook:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

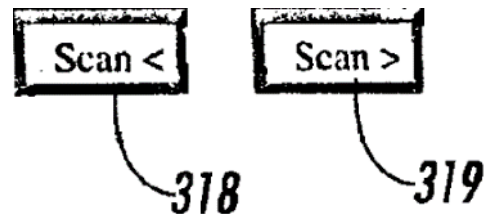


(Robertson, Fig. 15 (highlighting added).)

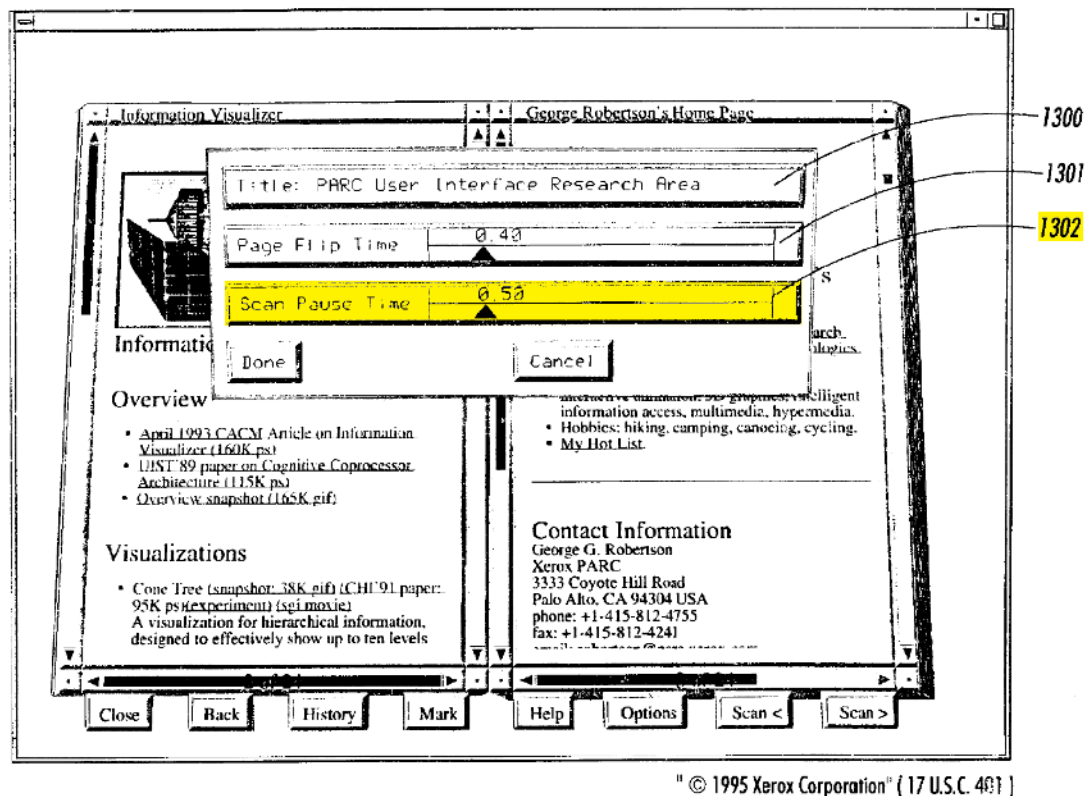
82. Figure 15 above shows a WebBook providing “a feature of a complete Web Browsing system.” (Robertson, 10:15-16.) Area 1501 in the center shows the current web page viewed by the user. (Robertson, 10:17-19.) Area 1503 to the left “contain[s] pages which have been viewed by the user.” (Robertson, 10:22-24.) Area 1502 on the right contains “a collection of ‘thumbnail’ images of pages,” which have “not yet been viewed by the user.” (Robertson, 10:19-22.) “As the user views pages and skips to different pages, the prior current page is moved to the area 1503.”

(Robertson, 10:24-26.) “It should be noted that the collection of pages, i.e. the current page **1501** and the pages in the areas **1502** and **1503** themselves may be pages from a WebBook.” (Robertson, 10:26-28.)

83. Robertson also discloses user interface features for controlling the WebBook display. These are shown on the bottom of Figure 3 above which shows “book control buttons” **312-319**. (Robertson, 6:30-57.) The two buttons on the far right side, “Scan <” **318** and “Scan >” **319** (shown at right), allow the user to automatically navigate backward or forward, respectively, through the web pages in a WebBook. “The button **318** labeled ‘Scan <’ provides for automatic scanning of pages to the lower numbered pages. The button **319** labeled ‘Scan >’ provides for automatic scanning of pages to the higher numbered pages.” (Robertson, 6:43-46.) “The scan operation will cause the pages to turn at a predetermined rate.” (Robertson, 9:5-6.) Figure 13 of Robertson discloses a user interface for adjusting the rate for the scan operation. (Robertson, 9:45-46.)



Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1



(Robertson, Fig. 13 (highlighting added).)

84. User interface bar **1302** in Figure 13 allows the user to vary the scan pause time, which in this example is currently set to 0.50. (Robertson, 9:49-50, 9:57-58.) “The scan pause time refers to a pause in between page turns during a scan operation.” (Robertson, 9:50-51.) The user can decrease or increase the scan pause time by moving the indicator to the left or right. (Robertson, 9:53-58.)

85. I will provide more information about Robertson in my discussion of the claim limitations below.

(e) **Miller (EX1010)**

86. **Miller** is a book entitled *Using CompuServe, Second Edition*. I am informed that Miller qualifies as prior art because it was published before the earliest priority date of the challenged claims.

87. My Declaration cites Miller in combination with Guinan primarily for claim 39, which recites “**selecting a category from a predefined list of categories,**” and “**performing an on-line search related to said selected category.**” Miller discloses the claimed category selection and performance of a search based on the selected category categories.

88. As its title implies, Miller explains how to use CompuServe, an early commercial on-line service that was popular in the mid-1990s. “CompuServe is like a giant electronic community.” (Miller, p.1.) CompuServe provided a feature known as “forums,” which were on-line communities where members could exchange information and messages about particular topics of interest. (Miller, p.29 (“You can find forums on just about any topic, from scuba diving to C programming.”); *id.*, p.30 (“A forum is like one of those after-school clubs you belonged to back in high school. A bunch of people with the same interests get together to talk and exchange information.”).) Forums included a message section, “where forum members exchange messages with each other,” and a file library,

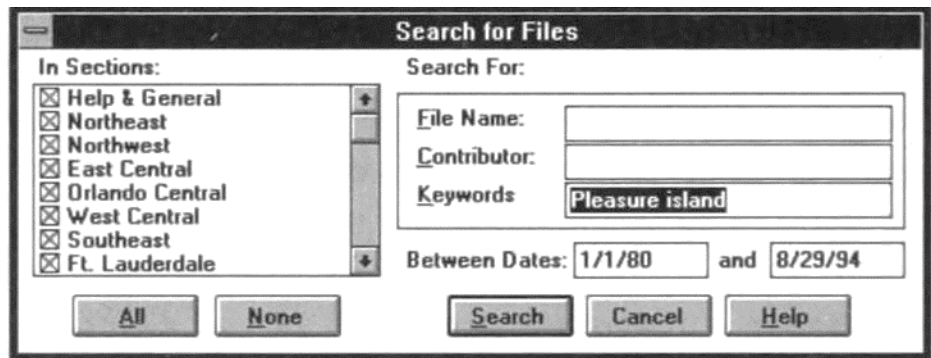
“where forum files are stored.” (Miller, p.34.)

89. Miller discloses that, “[i]n addition, every forum’s message sections and file libraries are comprised of multiple *sections*. Each section represents a more focused area of interest within the forum topic.” (Miller, p.35 (italics in original).) “For example, the Florida Forum has sections for Ft. Lauderdale, Tampa, St. Petersburg, and other specific locations.” (*Id.*) “The MacMillan Computer Publishing Forum (where you talk to Que [the publisher of Miller] has sections for Word Processing, Databases, Programming, and several other specific topics.” (*Id.*)

90. Miller also discloses features of CompuServe that allow users to search forums and sections within the forums. Figure 3-9 shows a user interface dialog box that allows a user to perform such a search:

Fig. 3.9

Use the Search for Files dialog box to search for specific forum files.



(Miller, p.42, Fig. 3-9; *see also id.*, p.42 (“A better way to find files... is to click the Search Libraries button in the toolbox. When you do this, a Search for Files dialog box appears (see fig. 3.9).”).) The dialog box, as shown, allows the user to select

one or more sections to search, and enter additional information such as keywords. (*Id.*) “When you click the Search button, the forum’s libraries are searched for any and all files that match your search criteria.” (Miller, p.43 (boldface in original); *see also id.*, p.419 (showing similar example).)

91. I will provide more details about Miller in my discussion of the claim limitations below.

(f) Ackermann (EX1012) and Resource Kit (EX1013)

92. These two final references are cited solely with respect to dependent claim 45, which further recites “the step of displaying a schedule window, for displaying a set of chosen line-up of show structures and time of their display, each corresponding to one of said categories.”

93. Ackermann is a book entitled *Using Unix*. I am informed that Ackermann qualifies as prior art because it was published before the priority date of the challenged claims. Ackermann describes features of the popular UNIX operating system. (Ackermann, pp.11-12.) Ackermann describes a well-known functionality of UNIX known as “cron” (short for chronograph) which allowed users to create schedules for performing commands at certain times. (Ackermann, p225.) Cron relied on “crontab” which provided one or more tables listing the commands to be executed and their corresponding times of execution. (Ackermann, pp.225-226.) As

further explained in Ackermann:

Use the crontab command to install a list of commands that will be executed on a regular schedule. The commands are scheduled to run at a specified time (such as once a month, once an hour, once a day, and so on). The list of commands to be performed on the specified scheduled must be included in the crontab file, which is installed with the crontab command. Once you install the crontab file, cron reads and executes the listed commands at the specified times.

(Ackermann, p.226.)

94. Crontab files follow a straightforward structure. Each line of a crontab file specifies both time information and a corresponding command:

Following is the syntax for the commands you enter in a file to be used by crontab. The first five fields are time-option fields (you must specify all five of these fields; use an asterisk * in a field if you want to ignore that field). Table 11.3 lists the time-field options available with crontab.

minute hour day-of-month month-of-year day-of-week command

(Ackermann, p.227.) Ackermann provides the following concrete example:

To sort a file named /usr/wwr/sales/weekly and mail the output to a user named twool at 7:30 each Monday, use the following entry in a file:

```
30 07 * * 01 sort /usr/wwr/sales/weekly |mailx -s"Weekly Sales" twool
```

This commands-specifies the *minute* as 30, the *hour* as 07, any day of month with the *, any month of year with another *, and the *day-of-week* as 01 (which represents Monday).

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

(Ackermann, p.227.)

95. In the example above, the first five pieces of information (“30 07 * * 01”) specify timing information for the command, and everything after and including “sort” is the string of the actual command to be executed. Each of the commands in a crontab file represents a command string that could be executed using the command line interface if input manually by the user.

96. Turning to the other reference relevant to claim 45, *Microsoft Office for Windows 95 Resource Kit* (“Resource Kit”) is a book published by Microsoft describing features of the Microsoft Office suite of applications for the Windows 95 operating system. I am informed that Resource Kit qualifies as prior art because it was published before the priority date of the challenged claims.

97. Microsoft Office was (and remains) a well-known suite of productivity tools that includes widely used applications such as Microsoft Word, Microsoft Excel, and most relevant here, Microsoft PowerPoint. My Declaration relies on disclosures in Resource Kit relating to PowerPoint, a business presentation program analogous to Harvard Graphics described in Benedict above.

98. Resource Kit explains that, like Harvard Graphics discussed above, PowerPoint supported self-running slide presentations in which slides change automatically and without user intervention. (Resource Kit, pp.338-339.) Resource

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Kit also describes a way to launch PowerPoint using straightforward commands executed at the Windows command line. As was well-known, the term “command line” (sometimes referred to as a “shell”) refers to an interface used by operating systems to receive text-based commands from the user. Command line interfaces existed (and continue to exist) for a number of popular operating systems including Windows and Unix, and were often used by developers to create simplified programs (sometimes referred to as batch files or shell scripts) to automate tasks.

99. Resource Kit confirms that, as was well-known, Microsoft Office applications could be launched from the Windows command line interface. Resource Kit provides an example of a complete command line string that instructs PowerPoint to launch and automatically begin displaying a particular presentation:

```
C:\MsOffice\PowerPnt\Powerpnt.exe /s c:\My Documents\MyKiosk.ppt
```

(Resource Kit, p.340, ¶5.) This command line string has three components: (1) the file path for the executable program for PowerPoint (Powerpnt.exe), (2) a command line switch (/s) that tells PowerPoint to automatically begin displaying a particular presentation, and (3) the name and path of the file of the presentation to display (MyKiosk.ppt). (Resource Kit, p.340; *see also id.*, p.341 (“**PowerPoint Command-Line Startup Switches.... /s** Starts PowerPoint in Slide Show view and runs the specification presentation or presentations”) (boldface in original).) The command

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

line string above, therefore, would have instructed PowerPoint to start and display the presentation named “MyKiosk.ppt” to the user.

100. I will provide more information about Resource Kit and Ackermann in my discussion of claim 45 below.

(g) Schloss (EX1011)

101. **Schloss**, U.S. Patent No. 5,706,507, is entitled “System and Method for Controlling Access to Data Located on a Content Server.” I am informed that Schloss qualifies as prior art because it was filed on July 7, 1995, which is before the priority date of the challenged claims.

102. My Declaration relies on Schloss for dependent claim 40, which depends from claim 39 and further recites “wherein prior to said step of automatically delivering the method further comprises the step of filtering the retrieved plurality of said resources to determine which of the retrieved resources are to be displayed.”

103. Schloss explains that, because of the growth of the Web, “millions of users began to have access to millions of documents through the World Wide Web,” with documents “contain[ing] text, graphics, audio, video, etc.” (Schloss, 1:30-35.) But easy access to large amounts of information created a problem – which persists to this day – of how to restrict access to undesirable or inappropriate content. “In

surfing the Web,” for example, “children may have easy access to inappropriate videos, graphical data and other related information.” (Schloss, 1:57-59.)

104. Schloss purports to address this problem through a technique that allows the client system to filter content downloaded from the Web to determine which content to display. The general technique is straightforward. A client system running a web browser requests and downloads content from a content server on the Web. (Schloss, 4:30-33 (“More specifically, a client system running a Web Browser request content from a content server **6** using a Hypertext Transfer Protocol (HTTP) request and receiving the content in a HTTP response.”).) The client can then communicate with an advisory server **20** that transmits “characterization data” about the content to the client, which the client uses to make a filtering decision:

The advisory servers maintain one or more knowledge bases **22** that characterize the content generated by one or more of the content servers **6**. In addition, the system includes one or more clients **25** (three shown) running a Web Browser that when set in an advisory mode, for each content request to the content servers **6**, requests characterization data from one or more of the advisor servers **20**. The advisory servers **20** generate the appropriate characterization data based upon the information stored in the knowledge base **22**, and transmit the characterization data to the client **25**. Upon receiving the characterization data, the client **25** utilizes the characterization data to

determine whether to filter the content data transmitted by the content server **6**.

(Schloss, 4:59-5:5.) The characterization data provided by advisory server **20** can indicate, for example, that the content is inappropriate for minors. (Schloss, 7:6-8 (“For example, the return code may indicate that the URL contains sexually explicit material not suitable for minors.”), 10:17-29 (“Not Appropriate for minors – violent”).)

105. Schloss discloses an alternative “advisory mode” in which the web browser inhibits downloading of the content until after it receives the corresponding characterization data from advisory server **20**. (Schloss, 5:24-33.) In a still further embodiment, the filtering functions of the browser are performed by a proxy server that sits between the client and advisory server **20**. (Schloss, 11:58-12:1.) Under this embodiment, the proxy server requests the characterization data from advisory server **20** about the content transmitted by content server **6**, and uses the characterization data to determine whether or not to transmit the content to the client. (Schloss, 11:61-12:4.) “Upon receiving the characterization data, the proxy server utilizes the characterization data to determine whether to filter the content data transmitted by the content server **6**.” (Schloss, 12:1-4.) I will provide more information about Schloss in my discussion of claim 40 below.

**B. Ground 1: Claims 1-8 Are Obvious Over Shimizu in view of
Benedict and Fox**

1. Independent Claim 1

106. For reference, claim 1 recites:

1. A method for displaying on a user's computer, content derived from a plurality of resources in an organized arrangement comprising the steps of:
 - [a] creating a show structure of nodes, each node identifying a resource from a plurality of accessible resources;
 - [b] without requiring user input, automatically accessing a plurality of said resources each of said resources identified by each of said nodes; and
 - [c] displaying a content corresponding to each of said resources automatically in accordance with said show structure, wherein said step of creating further comprises the step of providing an interactively variable duration information, representing the duration within which a corresponding content to said resource is being displayed so as to enable a user to vary said duration.

('451, 18:25-39 (Claim 1; bracketed notation (e.g., [a]) added).)

107. The preamble of claim 1 recites “[a] **method for displaying on a user's computer, content derived from a plurality of resources in an organized arrangement.**” It is unclear if the preamble of claim 1 recites a claim limitation,

but even if it does, it is disclosed by the prior art.

108. Shimizu discloses a method of displaying, on a user's computer, content such as text, pictures, motion pictures, etc.:

When the entity information (character string, image, sounds, motion pictures) is acquired from the node information entity management part **8**, the content of the acquired information is reproduced and displayed in the node content display area **81** of the window **80** by applying the program which is associated with the pointer in the display procedure slot 62 of the node information structure **9**.

(Shimizu, 8:52-59.) Figure 8 shows a user interface for presenting node content:

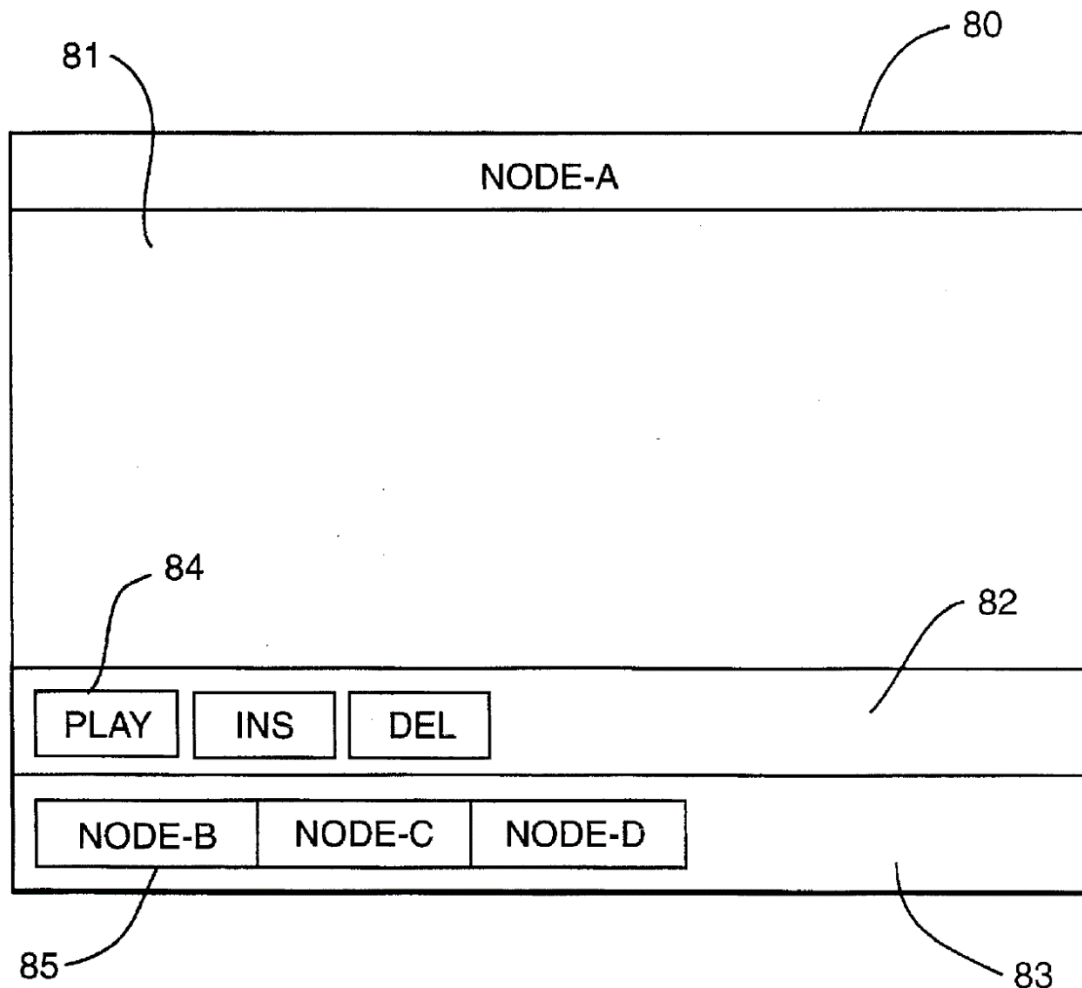


FIG. 8

(Shimizu, Fig. 8, 7:12-14 (“FIG. 8 is a view of a typical user interface screen display provided by an application program running on the embodiment.”).) As noted, when the user presses PLAY in Figure 8 (one of buttons **84**), the content associated with the current node (in this case NODE-A) is displayed in area **81** of window **80**. (Shimizu, 8:11-25, 8:49-59.) Shimizu therefore discloses “[a] **method for displaying on a user’s computer, content,**” as recited.

109. The content in Shimizu is also “**derived from a plurality of resources**,” as recited in the preamble. This is because the character strings, images, motion pictures, etc., in Shimizu are derived from stored information (for example, in computer files), from which the content can be derived. It would have been obvious that the content in Shimizu is stored in files, as these are basic units of data storage in a computer system. (See *Microsoft Press Computer Dictionary* (2d ed. 1994), EX1020, p.164 (“**file** A complete, named collection of information, such as a program, a set of data used by a program, or a user-created document. A file is the basic unit of storage that enables a computer to distinguish one set of information from another.”).) As will be described in more detail for claim 1**[b]** below, Shimizu further discloses that, in order to display content associated with a particular node, the system uses an entity information pointer slot **63** for the node, which points to the particular content that will be retrieved and displayed to the user. (Shimizu, 6:55-57 (“The entity information pointer slot **63** retains the pointer pointing to the entity information to be displayed by the node in question.”).) As explained in Shimizu:

Step **107** accesses and reads the entity information linked by the entity information pointer of the node information structure designated by the pointer pointing to the acquired node. When the entity information

(character string, image, sounds, motion pictures) is acquired from the node information entity management part 8, the content of the acquired information is reproduced and displayed in the node content display area **81** of the window **80** by applying the program which is associated with the pointer in the display procedure slot **62** of the node information structure **9**.

(Shimizu, 8:49-59.) Shimizu thus discloses that the displayed content is “**derived from a plurality of resources**,” with the resources corresponding to the sources of content identified by the entity information pointer **63**. As will be explained in more detail below, under the proposed combination with Fox, it would have been obvious that the entity information pointer **63** could take the form of a Uniform Resource Locator (URL) pointing to a resource (web page, image, etc.) on the Web.

110. Finally, Shimizu discloses displaying content on a user’s computer “**in an organized arrangement**,” as claimed. This is because the content of each node is presented in a particular order or sequence based on the sequencing of nodes:

The processing above is carried out so that the node C is displayed followed by node B and node D, in accordance with the order of the link information.

(Shimizu, 9:1-4.)

As described, the hypertext apparatus of the invention offers a user interface capable of putting into a desired order the links between the

nodes to be managed and of manipulating the established link order.
This makes it easy to create illustratively an application program
allowing the display of node contents to be switched in a desired order
and at predetermined intervals on the hypertext apparatus.

(Shimizu, 9:60-66; *see also id.*, 5:10-14.) Shimizu thus discloses displaying content
“**in an organized arrangement**” because, as noted above, node content is displayed
to a user in a particular sequence as indicated by the link information.

(a) “**creating a show structure of nodes, each node
identifying a resource from a plurality of accessible
resources;**” (Claim 1[a])

111. The proposed combination renders obvious this claim limitation. The
“**show structure of nodes**” in Shimizu corresponds to a set of nodes and their links
or connections to other nodes, as explained in more detail below. Shimizu explains
that node information management part **5** and link information management part **6**
are used to create the show structure of nodes:

The node information management part **5** manages a plurality of node
information structures holding the information about each node. The
link information management part **6** manages a plurality of link
information structures defining the relations between the nodes. The
link order manipulation part **7** changes the order of a plurality of link
information structures **10** referenced by the node information
management part **5** under the directions of the program execution part

3.

(Shimizu, 6:30-39.) A conceptual example of a show structure of nodes in Shimizu is shown in Figure 9:

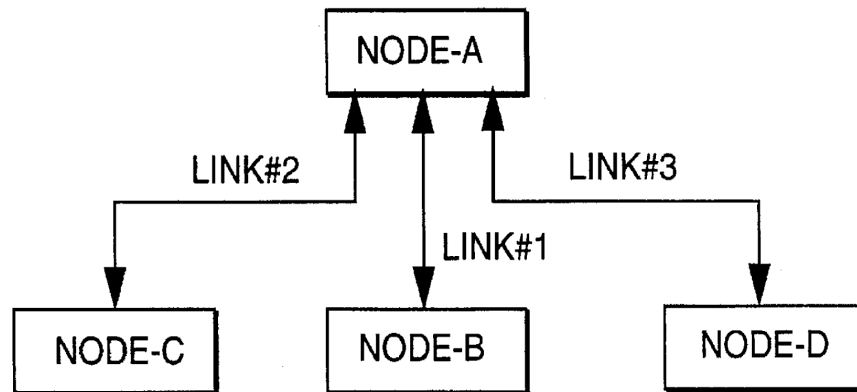


FIG. 9

(Shimizu, Fig. 9, 7:49-51 (“FIG. 9 is a conceptual view of typical nodes and links and the relations in effect therebetween when the order of link information is changed.”).) In this example, NODE-B, NODE-C, and NODE-D are linked to NODE-A by LINK#1, LINK#2, and LINK#3, respectively. (Shimizu, 7:51-54.) In this example, therefore, NODE-A would be displayed first followed by NODE-C, NODE-B, and NODE-D. (Shimizu, 7:54-65.) Figure 10 shows a more concrete representation of the show structure from Figure 9:

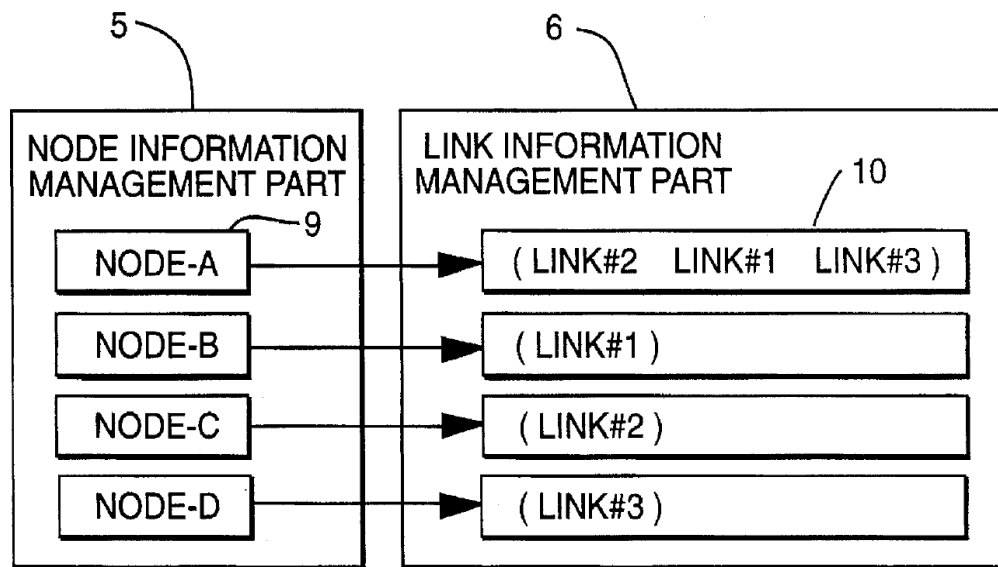


FIG. 10

(Shimizu, Fig. 10, 7:67-8:5 (“FIG. 10 is a schematic view of a typical data constitution in which the hypertext data of the nodes and links in FIG. 9 is stored in the node information management part and link information management part of the embodiment. As explained above, the node A is linked to the node C, node B and node D, in that order.”).)

112. As explained for the preamble, the node and link information is stored, respectively, in node and link information structures. (Shimizu, Fig. 6 (node information structure), Fig. 7 (link information structure).) Shimizu thus discloses “a **show structure of nodes**,” which corresponds in Shimizu to an arrangement or set of nodes that define their associated content and their links to other nodes. With

respect to the act of “**creating**” the claimed show structure, Shimizu discloses a number of techniques for adding new nodes and rearranging the order of existing nodes. (Shimizu, *e.g.*, 7:20-48, 9:5-59.) It would have been obvious to an ordinarily skilled artisan that the show structure in Shimizu was “**creat[ed]**,” as it must have been brought into existence in order to perform the node display and other functions described in Shimizu.

113. The proposed combination also discloses “**each node identifying a resource from a plurality of accessible resources.**” As noted, the node information management part **6** in Shimizu system holds information about individual nodes. Figure 6 shows an example of a node information structure **9**:

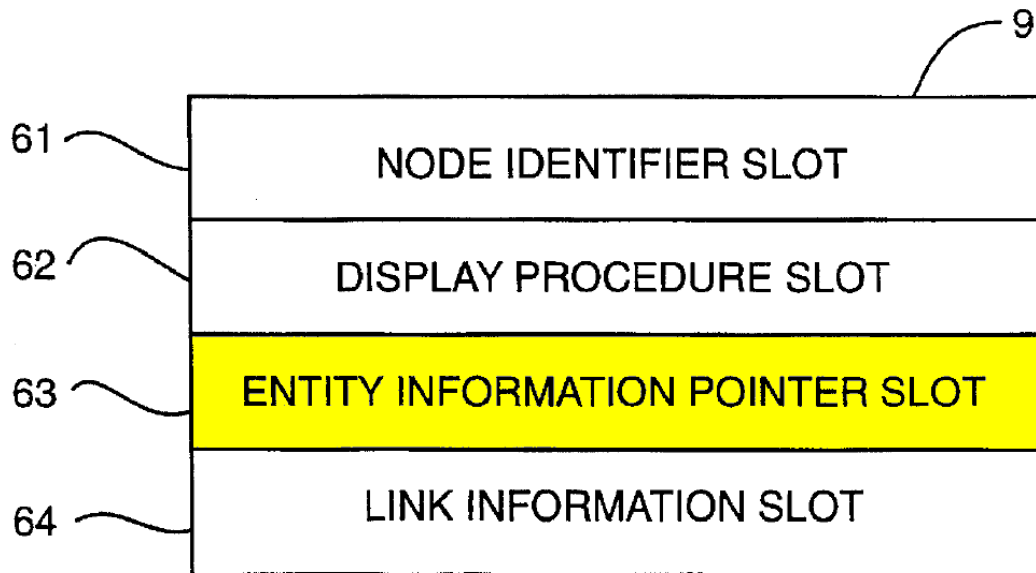


FIG. 6

(Shimizu, Fig. 6 (highlighting added), 6:44-45 (“FIG. 6 is a view showing members of a node information structure held by the node information management part 5.”).)

Entity information pointer slot **63** shown in yellow holds an identification of the resource associated with the node:

The entity information pointer slot **63** retains the pointer pointing to the entity information to be displayed by the node in question. The entity information thus pointed to may be a character string, image information, motion pictures or sounds.

(Shimizu, 5:55-59.) As noted previously, the system in Shimizu uses the entity information pointer to identify and retrieve the content of a node that will be displayed. (Shimizu, 8:49-59.) The nodes in Shimizu thus through its node

information identifies a particular content resource (e.g., entity information such as text, image, motion picture, sound, etc.) among a number of available resources. As explained below, it would have been obvious based on **Fox** that each of the “**plurality of resources**” could each have been on a different website on the Web identified by a URL.

114. I note that my mapping of Shimizu is also consistent with claim construction rulings in prior litigations. I am informed that “**show structure of nodes**” was defined in a previous litigation as “*a structure that is arranged for the display of content by specifying one or more paths through a plurality of nodes.*” I am further informed that “**node**” was defined as “*an identifier of a resource that includes an address to the resource and the duration for which the resource’s content is to be presented by default.*” It is not necessary for me to opine on the correctness of these constructions because, as I will show below, the prior art readily satisfies each of them. As discussed above, the node and link information in Shimizu identifies a set of nodes, the content associated with each node, and the links that specify the order in which the nodes are displayed. It would have been obvious that entity information pointer slot **63** part of the node information structure in Shimizu provides “*an address to the resource*” because it provides a pointer identifying the location of the content associated with the node. Ordinarily skilled artisans, in fact,

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

frequently used the words “pointer” and “address” interchangeably. (*See Microsoft Press Computer Dictionary* (2d ed. 1994), EX1020, p.308 (“**pointer**... In programming and information processing, a variable that contains the memory location (address) of some data rather than the data itself.”).) Under the further combination with Fox, as explained below, each resource could have been identified by a Uniform Resource Locator (URL) pointing to the location of the resource (*e.g.* web page, image, etc.) on the Web. It was well-known that a URL provides another example of an address. (Fox, p. 13 (“A *URL* (or *Universal Resource Locator*) is the address of a document, file, or other Internet resource... The World Wide Web gives every article, newsgroup, database, computer, file, or other Internet fragment its very own URL.”).)

115. I note that the challenged patent itself does not specify any particular format or organization for the claimed “**structure**,” but rather, describes it as broadly referring to an arrangement containing a set of nodes. (’451, 2:55-60 (“This arrangement of resources, is defined as a ‘show structure’ or ‘structure,’ which contains a set of nodes. Each node represents a resource such as a web page, an executable file, or a data file stored at a predetermined location and accessed via its corresponding address.”).) The proposed combination similarly discloses a set that includes an arrangement of nodes which are, as explained previously, defined by

their node and link information.

116. I note that the prior litigation construction of “**node**” further requires that the node also include “*the duration for which the resource’s content is to be presented by default.*” I note that this concept is separately recited in claim 1[c] as the “**interactively variable duration information.**” Nevertheless, this requirement of the construction of “**node**” is rendered obvious by the proposed combination.

117. For example, Shimizu discloses a default display time (*i.e.*, a “predetermined” time or interval in which node content is displayed before proceeding to the next node):

It follows that, on the hypertext apparatus, it is necessary to set the items of link information with their order of display properly established beforehand. Only this procedure makes it possible to create an application program that may be displayed on the screen by reading, from primary or secondary storage and within a predetermined time, the node information held by the nodes that are linked sequentially starting from a particular node.

(Shimizu, 3:9-16.)

This makes it possible to build with ease an application program illustratively causing the display of the nodes to be switched in a specific order and at predetermined intervals.

(Shimizu, 5:11-14.) Shimizu also suggests the ability to vary the display duration

between nodes by explaining that “[i]t is also possible to build with ease a system for arranging information that emphasizes temporal relations between nodes (e.g. chronological tables).” (Shimizu, 9:66-10:2.)

118. But Shimizu does not appear to disclose further details about how duration information is specified or stored within its system, or whether such intervals between nodes can be changed. To the extent it is determined that Shimizu alone does not disclose the durational requirement of the construction of “**node**,” it would have been obvious in further view of **Benedict**.

119. As explained in **Part V.A**, Benedict describes a software program known as Harvard Graphics that provided functionality similar, in some respects, to well-known business presentation programs such as Microsoft PowerPoint. (Benedict, p.1 (“Harvard Graphics for Windows is a program designed for creating business presentations.”).) Harvard Graphics allowed users to create a “presentation” including a series of slides. (Benedict, p.24 (“A *presentation* is a collection of slides stored in computer memory in a file on your hard disk.”) (italics in original).) Benedict also discloses features for viewing a presentation (referred to as a “ScreenShow”), for example, by displaying slides in a specified sequential order. “As you create and enhance presentations, you can use the ScreenShow feature to evaluate your work. ScreenShows are on-screen presentations in which

the program displays each slide in order.” (Benedict, p.19.) As noted, Benedict further discloses a feature known as “HyperShows” in which slides in a ScreenShow can include hypertext-like links allowing the user to switch to other slides in the ScreenShow by clicking on-screen links. (Benedict, pp.442-453.)

120. As relevant to the above-mentioned durational requirement, Benedict discloses additional features (known as “ScreenShow Effects”) that allow the user to customize the display of the presentation. (Benedict, pp.438-442.) As shown in Figure 17.1 of Benedict, one of these features allows the user to specify how long a particular slide in a presentation will be displayed before moving to the next slide:

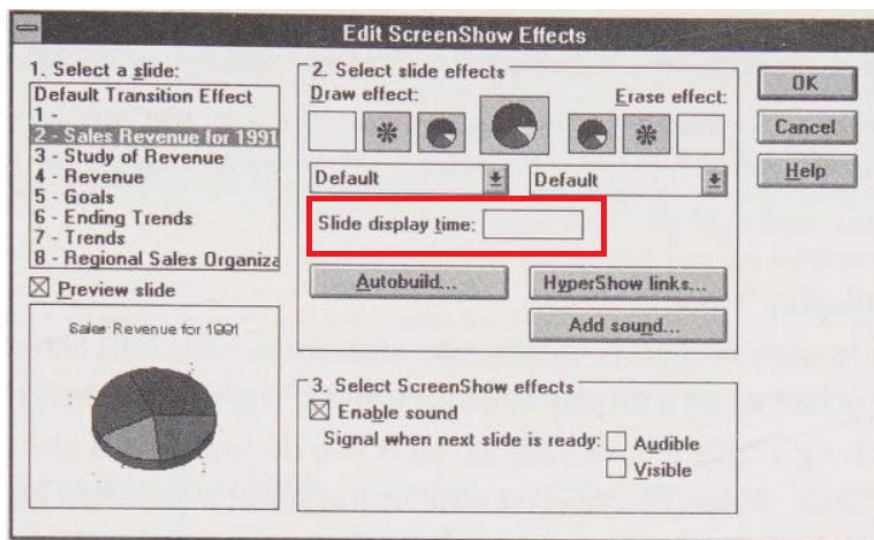


Fig. 17.1

The Edit
ScreenShow
Effects dialog box.

(Benedict, p.439, Fig. 17.1 (red box added).) The portion in the red box, as shown above, allows the user to specify how long the selected slide will be displayed:

The display time determines how long a slide remains on-screen if you

do not press a key to move to the next slide...

The default for the time is blank because Harvard Graphics continues to display the slide until you press a key. You can set a display time, however, by typing a time in the Slide display time: text box. You can enter the time in seconds or in minutes and seconds separated by a colon. Entering 23, for example, sets the display time to 23 seconds; entering 1:30 sets the display time to 1 minute and 30 seconds.

(Benedict, p.440 (italics in original).) Benedict explains that, using this feature, the user can create “self-running” presentations in which slides advance automatically after a specified period, without requiring user action:

Creating Self-Running Presentations

Self-running presentations automate your ScreenShow by displaying the slides for a specified length of time before the next slide appears—pressing a key is not necessary. At the end of the presentation, a HyperShow link causes the ScreenShow to start over from the beginning. The presentation continues until you press the Escape key. To create a self-running presentation, you must set the display time for the slides and create the link that begins the presentation again.

(Benedict, p.453 (boldface in original).)

121. The user can also specify a default display time applicable to the presentation, without having to separately assign the time to each slide:

Setting Default Effects... When you select a default transition,

Harvard Graphics uses the same effect to draw and erase all slides.

[* * *]

When the [Edit ScreenShow Effects] dialog box appears, follow these steps to change the default effects and time display:

[* * *]

4. Click the Set display time: text box and type the default display time.

If you do not set a time, the slides remain on-screen until you press a key.

(Benedict, p.441-442, ¶4.)

122. *Manner, Rationale and Motivation to Combine (Shimizu with Benedict)*: It would have been obvious to combine Shimizu with Benedict, predictably resulting in the hypertext display system of Shimizu with the ability to interactively vary, for each node, the duration of display for the node's content. Accordingly, the “**node**” in Shimizu under the proposed combination would have further included information specifying this user-specified duration. In a real-world implementation of the proposed combination, for example, the display time could have been specified within the node information structure **9** of Shimizu. I note that node information structure **9** in Shimizu also contains display procedure slot **62** (Fig. 6), which points to “a procedure program function used when the node in question is to be displayed on the screen.” (Shimizu, 6:52-55.) Shimizu thus contemplates

that node information structure 9 may contain information about how the node will be displayed to the user, and as such, it would have been obvious that durational information could also have been included as part of node information structure 9. This durational amount could have been a default display time specified by the user, or a display time specific to the node in question, as explained above.

123. Shimizu and Benedict are analogous references in the same field of techniques for providing organized or sequenced presentation of content, and both references would have been reasonably pertinent to a number of problems facing the inventor. For example, Shimizu and Benedict would have been reasonably pertinent to the problem of how to structure and organize information about nodes, in order to keep track of their associated content and the nodes to which they are linked. They would also have been reasonably pertinent to the problem of how to implement display of node content for a specified duration based on user preference.

124. Shimizu, as explained above, already discloses the ability to display node content for a “predetermined” time or interval (Shimizu, 3:9-16, 5:11-14, 9:63-66). Shimizu also provides a suggestion to vary this predetermined duration between by explaining that “[i]t is also possible to build with ease a system for arranging information that emphasizes temporal relations between nodes (e.g. chronological tables).” (Shimizu, 9:66-10:2.) It would have been obvious to enhance Shimizu

based on Benedict to allow the user to specify the display time. An ordinarily skilled artisan would have been motivated to make this enhancement to improve user experience by providing greater customization of the duration that node content is displayed.

125. This would have provided clear benefits over a fixed-length duration that could not be modified by the user. An ordinarily skilled artisan (and even laypersons) would have readily understood the benefits of allowing the user to specify the display time. For example, nodes may have included content that was more complex thus justifying a longer display time to provide the user with more time to view and digest the content. Conversely, nodes with less or simpler content may have benefitted from a shorter display time, as this would have allowed the user to progress more quickly through nodes. In any case, an ordinarily skilled artisan would have been motivated to combine Shimizu with the “display time” teachings of Benedict to obtain the advantages of an interactively variable duration for display of node content.

126. An ordinarily skilled artisan would have had at least a reasonable expectation of success and could have implemented the proposed combination using conventional programming techniques. As noted, Shimizu itself already contemplates switching the display of nodes at “predetermined intervals” (Shimizu,

5:11-14), and it would have been trivial to adapt the node information structure in Shimizu to include user-provided information specifying the display time for the corresponding node. Shimizu itself clearly contemplates the ability to customize the display of nodes, and as explained above, node information structure **9** (Fig. 6) includes display procedure slot **62** which points to a program function or procedure used to display the node content, thus making clear that node information structure **9** can hold information about how the node will be displayed. (Shimizu, Fig. 6, 6:52-55.) The disclosures of Shimizu and Benedict would have provided more than sufficient detail to enable an ordinarily skilled artisan to implement the proposed combination as described, without undue experimentation.

Issue: Use of URLs to Identify Web Resources

127. I note that Shimizu and Benedict do not expressly disclose a system in which resources are on the Web and identified by Uniform Resource Locators (URLs). Shimizu discloses that resources could have been stored on “primary storage” or “secondary storage” (*e.g.*, Shimizu, 6:39-43, 2:49-53). The term “secondary storage” was commonly understood as referring to any storage other than the computer’s “primary storage,” *i.e.* its volatile storage such as the random access memory (RAM) directly accessible to the computer’s processor. (*See Microsoft Press Computer Dictionary* (2d ed. 1994),

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

EX1020, p.351 (“**secondary storage** Any data storage medium other than a computer’s random access memory--typically, tape or disk. *Compare* primary storage”) (bold and italics in original), *see also id.*, p.314 (“**primary storage** Random access memory (RAM); the main general-purpose storage region to which the microprocessor has direct access. A computer’s other storage options, such as disks and tape, are called secondary storage or (sometimes) backing storage.”).) Accordingly, an ordinarily skilled artisan would have understood “secondary storage” includes a storage medium accessible through a network, and as such, it would have been obvious to an ordinarily skilled artisan based on Shimizu alone that content resources could have been accessible via a communications network. But Shimizu does not expressly disclose an example in which resources are expressly described as accessible via a URL, so to the extent that the “**plurality of resources**” requires distinct websites on the Web identified by URLs, this would have been obvious in further view of **Fox**.

128. Fox is a textbook describing various techniques for creating web pages. Fox confirms that a web page “can be anything you imagine it to be: a hypertext document, graphics sound, animation, tabular data, software files, a discussion group, a search index, a specialized program, and more.” (Fox, p.8; *see also id.*, p.9 (“The World Wide Web uses hypertext to link all its pages together.”).) As was

well-known to ordinarily skilled artisans, the Web provided an example of a hypertext system. (Fox, EX1005, pp.8-10 (describing the Web as an example of a hypertext system).) As was basic knowledge to ordinarily skilled artisans, and even laypersons, each resource on the Web was identified by a Uniform Resource Locator (URL). (Fox, p.13 (“A *URL* (or *Universal Resource Locator*) is the address of a document, file, or other Internet resource... The World Wide Web gives every article, newsgroup, database, computer, file, or other Internet fragment its very own URL.”); *see also id.*, pp.326-327 (describing IMG tag identifying location of image resource).)

129. In fact, Fox makes clear that URLs could be used to identify resources to carry out the same functions as Shimizu and Benedict, *i.e.*, to create sequential presentations such as the ordered hypertext display in Shimizu or the ordered show presentations in Benedict. For example, Fox discloses a technique in which a web page, after a specified amount of time, can cause the web browser to “refresh” and load another web page, which can be used to create slide show-type presentations of web content:

To load up a different document after a set amount of seconds, simply specify a full URL using the URL attribute. For example, to load the page at *http://www.smartpants.edu/sample.html*, after a minute, use the

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

following markup as the first line of your HTML document:

```
<META HTTP-EQUIV="Refresh" CONTENT=60  
URL= http://www.smartpants.edu/sample.html>
```

You can use this trick to flip between two documents, animate a
sequence of images, or create an automatic slide show of many Web
pages.

(Fox, p.316 (italics in original; underlining added).) The concept of “refreshing” a web page was well-known (and remains well known) to those familiar with web browsers, and this was (and remains) a common feature of many websites. Refreshing was commonly used to load an updated version of the *currently displayed* web page, but as shown above, the <META> tag used for this feature can specify the URL of any web page, including a different web page. In the example from Fox above, the <META> tag tells the web browser to “refresh” after 60 seconds by switching to the web page at <http://www.smartpants.edu/sample.html>. In any case, Fox confirms that it was known to use URLs to identify resources for a presentation system such as the Shimizu-Benedict combination in which the resources associated with a series of nodes are presented in a particular sequence and for a specified duration.

130. ***Rationale and Motivation to Combine (Shimizu and Benedict with Fox):*** It would have been obvious to combine Shimizu and Benedict with Fox,

predictably resulting in the Shimizu-Benedict system in which each node was associated with a resource identified by a URL, such a web page, image, or other Web resource. To the extent each of the “**plurality of resources**” is further construed to require that each resource be located at a distinct website identified by a unique domain, this would also have been obvious as this is a built-in and well-known capability of URLs.³ (Fox, pp.15-16 (listing several examples of URLs for webpages at different website domains).) One of the components of a URL is the name of the server where the resource is located, typically expressed as a domain name (*e.g.*, “uspto.gov”). The use of URLs to identify resources on the Web would have been a straightforward adaptation; for example, entity information pointer slot **63** discussed above could have been readily adapted to hold one or more Uniform Resource Locators (URLs) identifying resources (web pages, images, etc.) on the Web. (Fox, p.13 (“The World Wide Web gives every article, newsgroup, database, computer, file or other Internet fragment its own URL.”), pp.326-327 (describing

³ I understand Patent Owner does not contend that the claims require that each of the “**plurality of resources**” refer to a resource at different website/domain, but in the event the claim were construed in this manner, I have explained in the text why this would have been obvious. It would have been obvious that the URLs that identify the plurality of resources could each have pointed to a different website/domain. (Fox, pp.15-16 (showing exemplary HTML URLs pointing to moolah.com, testlabs.edu, strange.org, and abc.123.com).)

how to embed graphic files on a web page using IMG tags identifying the URL of images for display on the page).) This would have been a trivial adaptation, as the very purpose of a URL is to provide a pointer to a content resource to facilitate access by a web browser. For example, Shimizu explains that the node content information may be stored “in primary or secondary storage.” (Shimizu, 6:39-43.) A URL would have been a natural way to implement entity information pointer slot **63** because a URL provides a standardized and uniform way to identify the location of a file stored on the Internet or in local storage. (Fox, pp.15-17.)⁴

131. Fox is an analogous reference for the same reasons as Shimizu and Benedict, as it also falls with the same field of techniques for organized or sequenced presentation of content and would have been reasonably pertinent to a number of problems facing the inventor. For example, Fox would have been reasonably pertinent to the problem of how to structure and present a series of Web resources

⁴ For example, with respect to local storage, a standard URL format (*i.e.*, file://) can be used to identify the location of a file on the user computer’s local storage. (Fox, p.17 (“The file service is almost identical to the ftp service, but is a quicker way of retrieving local files. You can use file URLs to load and display sound, image, or text files....”).) URLs can also be used to identify a file on a remote server on the Internet, for example, by identifying its hostname in the URL. (Fox, p.14 (“The second part of most URLs is the full Internet address of the machine where the file is located.”).)

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

in a particular order, and as noted, Fox also contains pertinent teachings for specifying a duration for the display using industry standard <META> HTML elements. Fox would also have been relevant to understanding the composition and function of URLs, which would have been reasonably pertinent to problems relating to how to identify the location on the network of content resources associated with nodes.

132. An ordinarily skilled artisan would have been motivated to combine Shimizu and Benedict with Fox to take advantage of the growing popularity of the Web. The Background of the challenged patent, in fact, concedes that the web was “popularized in 1993.” (’451, 1:22-25.) The challenged patent further notes that companies such as Cisco and Boeing were already conducting “web-based sales in excess of one billion dollars,” with Dell having “sales in excess of \$1 million per day via their website.” (’451, 2:38-43.) As Fox confirms, ordinarily skilled artisans had recognized the enormous commercial potential of the Web and its ability to reach millions of potential users. (Fox, p.229.) Fox confirms, for example, that the Web can be used to “[p]ublish[] text, images, software, or any other computer file to a potential audience of millions.” (Fox, p.229; *see also id.*, p.316 (“Proofread your pages for spelling and grammar errors. The page will be seen by millions of adoring fans around the world.”), p.198 (“Money makes the world go round, and the

same can be said for the World Wide Web. A huge chunk of the Web's growth is due to large corporations and enterprising entrepreneurs hoping to turn the Web into tomorrow's most lucrative medium."").)

133. An ordinarily skilled artisan would thus have been motivated to adapt the hypertext system based on the combination of Shimizu and Benedict into a web-based system, among other reasons, to allow node content to be viewed by a potentially enormous number of users via their web browsers. An ordinarily skilled artisan developing a system based on the proposed combination would have wanted to avoid being left behind in the growing popularity of the Web, and thus, would have been motivated to adapt the system to support web-based resources.

134. As an additional but related motivation to combine, allowing node content to be identified using URLs as disclosed in Fox would also have expanded the universe of content that could have been incorporated into nodes. This is because, under the proposed combination, nodes could have identified any content on the Internet accessible via a URL, and thus, could have drawn from a massive universe of available content resources beyond the content stored on the user's local computer or the user's local area network.

135. An ordinarily skilled artisan would have had at least a reasonable expectation of success. As noted, the Web was well-known and identifying

resources using URLs, as described in Fox, was straightforward if not trivial. Fox itself confirms URLs were already being used to identify resources for purposes of sequential presentation systems, as confirmed by the example from Fox discussed above that uses industry standard HTML <META> elements. (Fox, p.316.) The disclosures of Shimizu, Benedict and Fox would have provided more than sufficient details to have enabled an ordinarily skilled artisan to implement the proposed combination as described, without undue experimentation.

(b) **“without requiring user input, automatically accessing a plurality of said resources each of said resources identified by each of said nodes; and” (Claim 1[b])**

136. This limitation adds little to what was already discussed for claim 1[a]. Shimizu discloses that the system automatically accesses a plurality of resources, i.e., the content items associated with the nodes, for example, when the content associated with the nodes (**“a plurality of said resources... identified by said nodes”**) is displayed:

FIG. 11 is a flowchart of steps in which the embodiment displays the contents of nodes in accordance with the order of their links. This processing is started using the mouse to click the ‘PLAY’ button in the operation designating area **82** of the window **80** in FIG. 8. The contents of the linked nodes are displayed in the order set by the link information.

(Shimizu, 8:11-16.) Shimizu further discloses that, in order to display the node contents, the system uses the entity information pointer slot **63** discussed in the previous limitation to identify the location of the content, and then to access and retrieve it:

Step 107 accesses and reads the entity information linked by the entity information pointer of the node information structure designated by the pointer pointing to the acquired node. When the entity information (character string, image, sounds, motion pictures) is acquired from the node information entity management part 8, the content of the acquired information is reproduced and displayed in the node content display area 81 of the window 80 by applying the program which is associated with the pointer in the display procedure slot 62 of the node information structure 9.

(Shimizu, 8:49-59.) Shimizu thus discloses “**automatically accessing a plurality of said resources each of said resources identified by each of said nodes,**” as claimed. Under the proposed further combination with Fox, as noted, the entity information pointer slot **63** would have included a URL associated with the text, image, sounds, or motion pictures on the Web to be displayed with the node.

137. Shimizu also confirms that the claimed step of accessing occurs **automatically**, and **without requiring user input**, because the access and subsequent display proceed from node-to-node without user intervention at

predetermined intervals, based on the link information. (Shimizu, Fig. 11, (Steps 104-109), 8:67-9:4 (“In step **105**, the next node information is reproduced in accordance with the next link information. The processing above is carried out so that the node C is displayed followed by node B and node D, in accordance with the order of the link information.”), 5:11-14 (“This makes it possible to build with ease an application program illustratively causing the display of the nodes to be switched in a specific order and at predetermined intervals.”).) Shimizu does not disclose any user controls (*e.g.*, a “Next” button) for advancing to the next node. It thus would have been obvious that the content resources would have been accessed and presented on the user’s computer in continuous sequential fashion, one after another, without requiring any input from the user beyond a single initial click of a PLAY button that serves to commence the node display process. As I will explain below, the claim does not exclude a system in which the automatic sequential display process is commenced in response to a user pressing a PLAY button.

138. Benedict similarly discloses accessing “**automatically**” and “**without requiring user input,**” because ScreenShow presentations can be “self-running” and thus proceed automatically:

Creating Self-Running Presentations

Self-running presentations automate your ScreenShow by displaying

the slides for a specified length of time before the next slide appears—pressing a key is not necessary. At the end of the presentation, a HyperShow link causes the ScreenShow to start over from the beginning. The presentation continues until you press the Escape key. To create a self-running presentation, you must set the display time for the slides and create the link that begins the presentation again.

(Benedict, p.453.)

139. Fox also similarly discloses the ability to access resources automatically, as the “REFRESH” technique discussed above automatically retrieves the next page after the elapsed period of time. (Fox, p.315 (“A Web page can be programmed to tell a server to reload its data after a given amount of time. The server waits and automatically gives the Web page the new information.”); *id.*, p.316 (“To load up a different document after a set amount of seconds, simply specify a full URL using the URL attribute.”).)

140. Under the proposed combination of Shimizu, Benedict and Fox, as noted, the user would initially activate a “PLAY” button or other control to initiate the automatic sequential display of the content resources identified by the nodes. (Shimizu, 8:17-19, Fig. 11 (Step 101).) This presents no obstacle to obviousness because the actual step of **“accessing a plurality of said resources each of said resources identified by each of said nodes,”** is separate from user activation of a

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

PLAY button – the step is performed automatically by the software. The “PLAY” button or other user control is merely unclaimed prerequisite user step that does not perform the actual claimed step of accessing the claimed plurality of resources. This is also consistent with the challenged patent, which similarly discloses a “Play button” functionality for initiating the automatic presentation of nodes. (’451, 17:7-8 (“The Play button begins the presentation of information.”).) The ’451 specification further explains that “[a] triggering event, such as a mouse click on a single ‘link’ commences the presentation of a series of resources instead of one resource at a time.” (’451, 2:53-55.) I note that the district court’s claim construction order expressly adopted Patent Owner’s argument that “**without requiring user input**” did not exclude user action that triggers the commencement of the show. Accordingly, an ordinarily skilled artisan would not have interpreted the “**automatically**” and “**without requiring user input**” limitations as precluding any type of input that causes initiation of node display.

141. I further note that prerequisite user steps, including the initial activation of a “PLAY” button, would not perform the actual claimed step of accessing “**a plurality**” said resources associated with the plurality of nodes, because advancement to sequential nodes (which would have been required to access and display the claimed “**plurality**” of resources) would have taken place automatically

and without requiring any user input. Finally, I note that even if the claim required display of resources from start-to-finish without an initial user step in the causal chain (such as clicking a “PLAY” button), this would still have been obvious. As noted, Benedict above expressly discloses that a slide presentation can be configured, after its display is complete, to automatically and repeatedly replay itself. (Benedict, p.453 (“At the end of the presentation, a HyperShow link causes the ScreenShow to start over from the beginning. The presentation continues until you press the Escape key.”).) As such, the second, third, etc., runs through the set of nodes under the proposed combination – any one of which would have been sufficient to satisfy the claim – are even further removed from any previous user steps in the causal chain including the initial press of a “PLAY” button.

- (c) **“displaying a content corresponding to each of said resources automatically in accordance with said show structure, wherein said step of creating further comprises the step of providing an interactively variable duration information, representing the duration within which a corresponding content to said resource is being displayed so as to enable a user to vary said duration.” (Claim 1[c])**

142. As just explained for claim 1[b], content corresponding to the resources associated with each node (*e.g.*, character string, image, sounds, motion pictures, etc.) is also automatically displayed to the user, in accordance with the show

structure. (Shimizu, 8:11-16, 8:49-59; Benedict, p.435 (“Self-running presentations automate your ScreenShow by displaying the slides for a specified length of time before the next slide appears—pressing a key is not necessary.”), Fox, p.316) The content is automatically displayed “**in accordance with said show structure,**” because it is presented in the sequence specified by the corresponding link information for the nodes. (Shimizu, 8:67-9:4, 5:11-14.)

143. Shimizu also discloses that the step of creating the show structure of nodes also comprises providing duration information (*i.e.* a “predetermined interval” for the display of the content associated with a node):

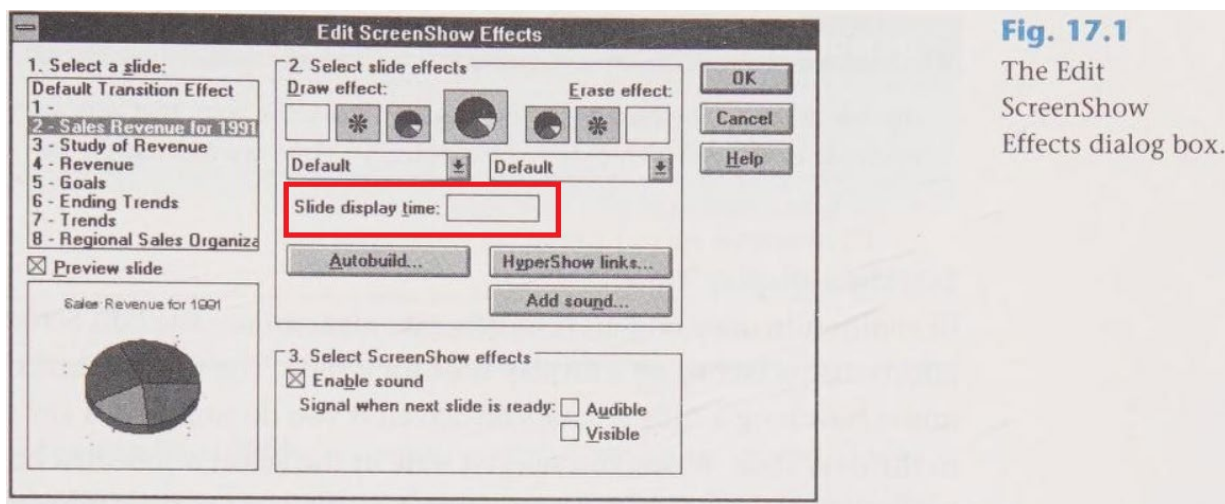
As described, the hypertext apparatus of the invention offers a user interface capable of putting into a desired order the links between the nodes to be managed and of manipulating the established link order. This makes it easy to create illustratively an application program allowing the display of node contents to be switched in a desired order and at **predetermined intervals** on the hypertext apparatus.”

(Shimizu, 9:60-66; *see also id.*, 5:11-14 (“This makes it possible to build with ease an application program illustratively causing the display of the nodes to be switched in a specific order and at predetermined intervals.”).)

144. As explained for claim 1[a], Shimizu does not disclose further details about the predetermined interval the switching the display from one node to the next,

and does not expressly disclose that the duration information is “**interactively variable**” or that the content is displayed “**so as to enable a user to vary said duration.**” But this would have been obvious in further view of Benedict.

145. As explained for claim 1[a], under the combination of Shimizu with Benedict, each node would have included a display time that could have been interactively varied by the user – under two separate theories. First, Benedict discloses that the user can change the display time for a node using the ScreenShow Effects feature. (Benedict, *e.g.*, p.453 (“You set links and display times in the Edit ScreenShow Effects dialog box.”), p.440 (“You can set a display time... by typing a time in the Slide display time: text box.”).)



(Benedict, p.439, Fig. 17.1 (red box added).) As Benedict explains:

You can enter the time in seconds or in minutes and seconds separated by a colon. Entering 23, for example, sets the display time to 23

seconds; entering *1:30* sets the display time to 1 minute and 30 seconds.

(Benedict, p.440 (*italics in original*)).) As noted previously, this technique can be used to edit the display time for each slide individually, or for all slides. Benedict thus discloses that a user can interactively (through the user interface) vary the display time for a slide. Under the proposed combination of Shimizu and Benedict, therefore, the duration information for a node (specifying how long its content will be displayed) can be interactively varied by a user – and if desired, on a node-by-node basis. For purposes of this first way, the “user” may be a user creating or editing the presentation, who could have been the same user who would also be viewing the presentation.

146. As a second and independent way of satisfying the “**interactively variable duration information**” limitation, I am informed that in the pending litigation, Patent Owner has taken the position that a user can interactively vary the duration information through a user interface control that allows the user to skip or advance to the next item. I am further informed that Petitioner does not agree that such functionality satisfies the requirement of interactively variable duration information. Nevertheless, in the event the claim were construed to encompass this functionality, the prior art discloses and renders it obvious. More specifically, Benedict discloses that a user can vary the display time by pressing a button during

a ScreenShow displaying a current slide to advance to the next slide without having to wait the full display time: “The display time determines how long a slide remains on-screen if you do not press a key to move to the next slide.” (Benedict, p.440; *see also id.*, p.443 (“Next slide... Moves to the next slide in the presentation”), *id.* (“In addition to keys and buttons, you can define a link that determines the length of the slide display time.”).) The user under the proposed combination can thus vary (*i.e.* shorten) the display duration by immediately advancing to the next node before expiration of the node’s specified duration. I note that for purposes of this second way, the “user” could include the user viewing the presentation, and as noted, this could be the same user who would be creating or editing it.

147. Under the proposed combination, therefore, a user can interactively vary the duration within which a corresponding content to a node is being displayed, either **(1)** by allowing a user to interactively change the “display time” for one or more nodes through a user interface or **(2)** by allowing a user to press a button during display of node content to shorten/truncate the display time, to the extent it is determined that the claim encompasses such functionality.

148. I am informed that in a previous litigation involving the ’451 patent that a district court construed “**interactively variable duration information**” as “*a parameter specifying how long a content is to be displayed by default before a*

subsequent content is displayed, where the viewer of the content can change the parameter.” This construction is also satisfied by the proposed combination. In situation (1) discussed in the previous paragraph, the parameter corresponds to the duration for the display of node content, *i.e.*, a default display time (“predetermined interval”) in Shimizu before advancing to the next node, which the viewer can change as needed through a user interface such as Edit ScreenShow Effects dialog box in Benedict. It would have been obvious that the “predetermined” interval in Shimizu is a default value, because the intervals are “predetermined” and nothing in Shimizu suggests that they were selected by the user. In situation (2), to the extent it is determined that the claim encompasses such functionality, the parameter is the default amount of time the software waits when actually displaying a particular node contents before advancing to the next node (based on the established display time) – which the user can shorten by taking action (such as pressing a button) to advance to the next node before expiration of the display time, as explained.

149. This is also consistent with **Fox**, which confirms that the ability to change the display time could have been accomplished using any number standard web-based techniques such as a text field to solicit the display time from the user (Fox, pp.396-397), and immediately switching to the next node could have been accomplished using a standard URL or hyperlink to implement a “Next” button or

link (Fox, p.238 (“The [Next] button loads up the next page in the series, and the [Previous] button, naturally, loads up the previous page.”).) The system built based on the proposed combination, therefore, would have been further adapted to include a user interface feature for allowing the user to specify the time for displaying the content of individual nodes (*e.g.*, as a text box or other suitable control), and a feature for allowing a user to advance to the next node without having to wait the full display time. Under either theory, therefore, the duration could have been interactively varied by the user/viewer of the content. The rationale and motivation to combine Shimizu, Benedict, and Fox was provided for claim 1 [a] and applies here.

2. Claim 2

150. Claim 2 depends from claim 1, and further recites “**the method in accordance with claim 1, wherein said step of creating further comprises the step of providing a duration information, representing the duration within which a corresponding content to said resource is being displayed.**” Claim 2 does not appear to add anything that I did not already cover above. As explained for claim 1 [a] and 1 [c] above, the proposed combination renders obvious providing duration information, *e.g.*, a display time that represents the duration within which a corresponding content (*i.e.*, node content) is being displayed.

3. Claim 3

151. Claim 3 depends from claim 1, and further recites “**the method in accordance with claim 1, wherein said step of accessing further comprises the step of executing an application file residing in the user’s computer.**” This claim adds nothing of patentable significance, as it would have been obvious that the accessing step of claim 1**[b]** would have involved executing an application file on the user’s computer, *e.g.*, an application program that accesses and displays the content associated with each node. Shimizu and Benedict make clear that these functions are performed by an application program on the user’s computer. (Shimizu, *e.g.*, 9:63-66 (“This makes it easy to create illustratively an application program allowing the display of node contents to be switched in a desired order and at predetermined intervals on the hypertext apparatus.”), 5:40-42 (“FIG. 8 is a view of a typical user interface screen display given by an application program running on the embodiment...”); *see also* Benedict, p.483 (“To install Harvard Graphics on your hard disk, you must use the INSTALL.EXE program, which is on the first disk. This program decompresses the program files to a directory that you specify...”)).

152. Under the further combination with Fox, the system would also have included a web browser executing on the user’s computer for displaying node content from the Web (*e.g.* web pages, images, etc.). It would have been obvious to

an ordinarily skilled artisan that this web browser would have required one or more application files (*e.g.* executable or “.EXE” files) in order to execute the web browser on the user’s computer.

153. It would have been obvious to an ordinarily skilled artisan that an application program such as the ones in Shimizu, Benedict and Fox would have included at least one computer “**file**” residing on the user’s computer. (*See Microsoft Press Computer Dictionary* (2d ed. 1994), EX1020, p.164 (“**file** A complete, named collection of information, such as a program, a set of data used by a program, or a user-created document. A file is the basic unit of storage that enables a computer to distinguish one set of information from another.”).)

4. Claim 4

154. Claim 4 depends from claim 3, and further recites “**the method in accordance with claim 3 further comprising the step of generating a data file resulting from said executing step.**” It is unclear whether the step of “**generating**” a data file requires that a new file actually be created, as opposed to creating a copy of an existing file. With respect to the latter interpretation, Shimizu discloses generation of a data file resulting from the executing step, *i.e.*, the data file that represents the content associated with the node. Shimizu explains that, when the node contents are being displayed, the node contents (which are pointed to by the

entity information pointer **63**) are accessed and obtained. (Shimizu, Fig. 11 (step 107), 8:49-59.) Under the further combination with Fox, the node content would have been identified by a URL and downloaded over the Internet from a remote server in response to a request by application executing on the user's computer. It therefore would have been obvious that the acquisition and retrieval of node content under the proposed combination would have included at least generating a data file, by downloading the content from the Internet as one or more files and then storing the file(s) to the user's computer (either in volatile storage such as RAM or non-volatile storage such as a hard disk). As I will explain for claim 5, this was basic and built-in functionality of "cache" and "history" features of conventional web browsers. (Fox, p.82 ("Once you retrieve a Web page, SkipKnot saves it on your hard disk—graphics and all—for as long as you want to keep it."), *see also id.*, p.93 (describing a History List feature showing previously-visited and locally stored web pages).) With respect to the then-popular Netscape web browser, Fox explains that "[w]henver you retrieve a Web page, images are stored in memory. If you return to a page you've already seen, images load much faster due to this *memory cache*." (Fox, p.138 (*italics in original*); *see also id.* p.164 ("Memory cache [for Mosaic]: Speed up page reloading by storing each page that is read.")) It therefore would have been obvious that a data file would have been stored on the user computer as a

result of the executing step, under the proposed combination.

155. In the event “**generating**” is interpreted as requiring creation of a brand new data file, under the further combination with Fox, data files would have been newly-generated as part of the built-in way in which web servers create content in response to requests from web browsers.

156. It was well-known that web servers on the Internet often generated content in response to requests from web browsers, such as a request to access a resource at a particular URL. Fox describes a process using a well-known process known as the Common Gateway Interface (CGI), which allowed web servers to execute scripts that dynamically generated web content on-the-fly in response to a request from the web browser. (Fox, p.391, p.423 (“CGI scripts are used for doing all of the ‘cool’ stuff on the Net. There are... sites that create graphic images on-the-fly.”).) For example, Fox shows an example of a web page that uses a CGI script to generate a web page image representing the current image captured by a camera in a laboratory. (Fox, p.424, Fig. 15-2.) Aside from images, it was well-known (and in fact more common) to use CGI techniques to generate HTML on-the-fly in response to a user request, which is discussed extensively in Fox. (Fox, pp.421-23.) It would have been obvious to generate new content for a wide variety of purposes, such as news, financial, and weather content, as these are areas where

information regularly changes.

157. Under the proposed combination with Fox, therefore, it would have been obvious that the step of “**generating a data file resulting from said executing step**” would have included a web server generating a new data file (*e.g.* an HTML web page file, an image file, etc.) in response to the request from the web browser. For example, if the entity information pointer slot **63** for a particular node of Shimizu under the proposed combination included a URL to a resource on the Web (such as an HTML web page or image), when that node was to be displayed, the server identified by the URL would respond to the request by generating the data file in response to that request. More concretely, using the on-the-fly image example from Fox discussed above, as a result of the request from the web browser in response to a URL, the web server would have dynamically generated a data file (*e.g.*, image file) and transmitted it to the web browser for inclusion on the page.

158. Fox itself provides multiple express motivations to employ this technique, as it explains that “CGI scripts are used for doing all of the ‘cool’ stuff on the Net,” such as creating graphic images on-the-fly, accessing live video feeds, accessing large databases, among other uses. (Fox, p.423.) The use of CGI to generate content on-the-fly for users was well-known, and an ordinarily skilled artisan would have been motivated to use this technique to provide customized and

tailored content to the requesting user – a clear advantage when compared to simply sending the same (static) content to every requesting user. For example, a CGI script generating an image for a requesting user could dynamically generate different images based on the time-of-day, the location and identity of the user, or based on the current input of a device such as a camera, as discussed.

159. It would also have been obvious that these resources would have taken the form of “**a data file**,” as claimed, because files are fundamental units of data storage in any computer system, and are used by web servers and browsers to store resources for the Web. (Fox, pp.46-47 (identifying standard “file formats” for images, text, movies, etc.), p.118 (describing how to view an HTML web page in a web browser and “save the entire HTML file to your hard disk”); *see also Microsoft Press Computer Dictionary* (2d ed. 1994), EX1020, p.164 (“**file** A complete, named collection of information, such as a program, a set of data used by a program, or a user-created document. A file is the basic unit of storage that enables a computer to distinguish one set of information from another.”).)

5. Claim 5

160. Claim 5 depends from claim 4, and further recites “**the method in accordance with claim 4 further comprising the step of storing said data file for each resource accessed.**” This limitation adds nothing of significance because it

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

would have been obvious that the data file described in claim 4 would have been stored in order to be used. I note that claim 5 does not specify any particular location where “**said data file**” must be stored. Under the proposed combination of Shimizu, Benedict, and Fox, it would have been obvious that storing of the data file would have occurred in multiple places including at least (1) the server on the Internet that must store “**said data file**” in order to make that data available to the user computer for download, and (2) the user computer that downloads that data file from the server in order to access it and display it to the user. In both cases, it would have been obvious that the proposed combination would have performed the step of storing the data file, either temporarily in volatile memory (such as RAM) or in non-volatile storage (such as a hard disk or other persistent storage), by both the server and the user computer. As noted, files were fundamental units of data storage in computer systems, and it would have been basic knowledge of an ordinarily skilled artisan that the functions performed by the proposed combination would have included storing “**said data file**,” potentially multiple times in multiple different locations in both volatile and non-volatile memory, as explained.

161. As noted for claim 4, with respect to the user device, Fox confirms that storage is a built-in feature of conventional web browser software, which were capable of storing all web content downloaded by and displayed by the web browser

including web pages and graphic images (“**each resource accessed**”). This would have been known by even end-users familiar with the history or web cache features of popular web browsing programs. For example, with respect to one conventional web browser, Fox explains: “Once you retrieve a Web page, SkipKnot saves it on your hard disk—graphics and all—for as long as you want to keep it.” (Fox, p.82; *see also id.*, p.93 (describing a History List feature showing previously-visited and locally stored web pages).) With respect to the then-popular Netscape web browser, Fox explains that “[w]henever you retrieve a Web page, images are stored in memory. If you return to a page you’ve already seen, images load much faster due to this *memory cache*.” (Fox, p.138 (*italics in original*); *see also id.* p.164 (“Memory cache [for Mosaic]: Speed up page reloading by storing each page that is read.”).) Under the proposed combination, therefore, it would have been obvious to store a data file for each resource accessed (*e.g.*, web page, image, etc.) as part of the operation of the web browser.

6. Claim 6

162. Claim 6 depends from claim 5, and further recites “**the method in accordance with claim 5, further comprises the step of retrieving each of said data files.**” This limitation adds nothing of significance to what was already discussed. As explained for claim 4, Shimizu discloses that the node contents (“**each**

of said data files”) are retrieved as part of the display process. (Shimizu, 8:49-9, Fig. 11 (Step 107).) The step of retrieving occurs in at least two independent ways. First, under the further combination with Fox, as explained, each resource would have been identified by a URL that identifies its corresponding data file on the Web and is used to **retrieve** (download) that data file over the Internet. As explained for claim 5, in order to store data files on the user’s computer that were downloaded from the Internet, those files must have been **retrieved**. Second and alternatively, the step of retrieving also occurs when those data files, once they are stored on the user computer as files (whether in volatile and/or non-volatile memory), are then accessed and read into the computer’s memory in order to be displayed to the user.

7. Claim 7

163. Claim 7 depends from claim 6, and further recites “**the method in accordance with claim 6 wherein said step of displaying further comprises the step of reformatting said data files in accordance with a predetermined arrangement.**” I note that the ’451 patent does not define “**reformatting**” but provides a number of exemplary tasks including “utilizing formatting parameters,” such as “restraining page size to a default screen size,” among several others. (’451, 14:10-23.) Claim 7 would have been obvious over Shimizu and Benedict in further view of Fox, which discloses the ability of the web browser to size or position data

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

files (such as a web page or graphic files) based on formatting parameters specifying size, location, alignment, etc., based on interpreting standard formatting tags in the HTML of web pages. (Fox, p.282 (“Perhaps the most typically used style of text alignment is centering. Centered text is flashy, professional, and easy to read.”), p.292 (“If you want, you can determine the exact width [of a line] in pixels... Or even more easily, you can just determine what percentage of the screen the line should take up.”); *see also id.*, p.347 (“It’s a good idea to change an image’s size if it seems out of place, or unwieldy. You should resize images to give your Web page some uniformity.”), 333 (“Netscape lets you solve this problem with the HEIGHT and WIDTH attributes. You specify the image’s size in pixels, as follows: Netscape then uses this information to predraw the image box.”), p.335 (“If you don’t feel like calculating out the exact size of a scaled image’s height and width in pixels, you can have Netscape do the work for you. Simply specify the HEIGHT or WIDTH as a percentage of the available screen. This way, the image always appears relative to the rest of the Netscape page.”), pp.328-330 (discussing image alignment techniques).)

164. Under the proposed combination, therefore, the claimed reformatting would have been performed as part of the ordinary and routine operations of the web

browser interpreting the HTML and making decisions about the layout (format) of the data on the page. Depending on the HTML in the web page, these decisions could have included repositioning text on the page, resizing and/or scaling images on the page, or other forms of reformatting. An ordinarily skilled artisan would have been motivated to adapt the proposed combination to use these HTML features in order to ensure that content is displayed properly on the screen. It was well-known in the mid-1990s that different web browsers could often display the same HTML differently to the user, and as such, the ability of web browsers to interpret the HTML and reformat the displayed data based on that HTML would have provided a way to ensure a more consistent appearance across different web browsers.

8. Claim 8

165. Claim 8 depends from claim 7, and further recites “**the method in accordance with claim 7 further comprising the step of displaying control buttons for controlling the display of said data files.**” I note that claim 8 does not specify any details about the “**control buttons**” other than that they are “**for controlling the display of said data files,**” and each of the references in the proposed combination discloses this.

166. Shimizu, for example, discloses a series of on-screen buttons **84**, one of which includes the PLAY button for initiating the display of the data files under the

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

proposed combination. (Shimizu, Fig. 8, 7:24-28.) Benedict discloses that, during presentation of a HyperShow, on-screen control buttons can appear for controlling the display of content, such as navigating to different slides in the presentation:

Table 17.3 HyperShow Destinations	
Destination	Result
Next slide	Moves to the next slide in the presentation
Previous slide	Moves to the previous slide in the presentation
First slide	Moves to the first slide in the presentation
Last slide	Moves to the last slide in the presentation
Back up	Moves to the last slide shown on-screen
Ignore	Ignores the key so that nothing happens when you press a key
Launch application	Starts a Windows application from within the ScreenShow
Play sound file	Plays a sound file within the ScreenShow
Stop sound files	Stops sound files which were started from a previous HyperShow link

(Benedict, p.443.) As explained previously, users can also define their own HyperShow links to jump to other parts of the presentation. (Benedict, pp.444-49.) These links take the form of buttons. (Benedict, p.446 (“You create HyperShow buttons in the Slide Editor.”).)

167. Fox also similarly discloses, as a best practice to maintain a consistent user interface for navigating between web pages, navigation buttons for controlling

the display of web pages:

The best way of achieving consistency is to use a menu bar, or a set of uniform links. You can include these navigation controls at the top or bottom of every page. For instance, many Web pages use the following links at the bottom of each document:

[Home] [Next] [Previous]

If a user clicks on [Home], then the home page appears. The [Next] button loads up the next page in the series, and the [Previous] button, naturally, loads up the previous page.

(Fox, p.238.)

168. It would therefore have been obvious, under the proposed combination, to display control buttons for controlling the display of the content of nodes (“**said data files**”), such as “previous” or “next” buttons as described in Benedict and Fox. An ordinarily skilled artisan would have been motivated to include these control buttons to enhance user experience by allowing the user to exercise control over the display of nodes. For example, a “previous” control button would have been useful to allow the use to go back and revisit a previously viewed node, and a “next” control button useful for allowing the user to skip to reach more relevant nodes (without having to wait for the automatic advancement after on the specified duration).

C. Ground 2: Claims 10-12 Are Obvious Over Shimizu in view of Benedict, Fox, and Sagman

1. Independent Claim 10

169. For reference, claim 10 recites:

10. A method for displaying in a sequential arrangement on a user's terminal display, content derived via a communications network from a plurality of resources the method comprising the steps of:
- [a] identifying at least one resource from a plurality of accessible resources via said communications network, wherein each of said resources is associated with a corresponding icon displayed on a first section of said user's terminal;
 - [b] creating a show structure of nodes, each node identifying a corresponding resource by recording a sequence of said resources by dragging and dropping said icons on a second section of said user's terminal;
 - [c] without requiring user input, automatically accessing each of said resources identified by each of said nodes in accordance with said show structure;
 - [d] automatically retrieving a content corresponding to each of said accessed resources; and
 - [e] displaying each of said retrieved contents automatically in accordance with said show structure.

(’451, 19:1-20 (Claim 10; bracketed notation (e.g., [a]) added).) Many of the limitations of claim 10 are substantially similar to limitations recited in claim 1 that were covered above. In the interests of brevity and consistency, therefore, I will refer back to my analysis of claim 1 where appropriate.

170. The preamble of claim 10 recites “[a] **method for displaying in a sequential arrangement on a user’s terminal display, content derived via a communications network from a plurality of resources...**” It is unclear if the preamble of claim 1 recites a claim limitation, but even if it does, it is disclosed by the prior art.

171. As fully explained for the preamble of claim 1, the proposed combination of Shimizu, Benedict and Fox renders obvious a method for “**displaying in a sequential arrangement on a user’s terminal display, content,**” *i.e.*, content such as text, pictures, motion pictures, etc. The content is displayed “**in a sequential arrangement**” because, under the proposed combination as fully explained for claim 1, the content is presented in connection with the display of nodes presented in a specified order.

172. The content is also “**derived via a communications network from a plurality of resources,**” as recited. As explained, under the proposed combination with Fox, the content is derived from a number of resources on the Web, such as

web pages, images and other resources identified by Uniform Resource Locators (URLs) that identify their locations on the Internet (“**a communications network**”).

- (a) **“identifying at least one resource from a plurality of accessible resources via said communications network, wherein each of said resources is associated with a corresponding icon displayed on a first section of said user’s terminal;” (Claim 10[a])**

173. As fully explained for claim 1[a], the information management part 6 in Shimizu system holds information about individual nodes, including an entity information pointer slot 63 that points to the resource associated with the node. (Shimizu, Fig. 6 & 5:55-59 (“The entity information pointer slot 63 retains the pointer pointing to the entity information to be displayed by the node in question. The entity information thus pointed to may be a character string, image information, motion pictures or sounds.”).) Shimizu thus discloses “**identifying,**” through the node information as discussed above, “**at least one resource from a plurality of accessible resources.**” Under the proposed further combination with Fox, as explained for claim 1[a], each of these resources is accessible “**via said communications network,**” *i.e.*, via the Internet by using its respective URL. Under the proposed combination, “**each of said resources is associated with a corresponding icon displayed on a first section of said user’s terminal.**” With respect to Shimizu, Figure 8 discloses a user interface for showing icons reflecting

the order of nodes:

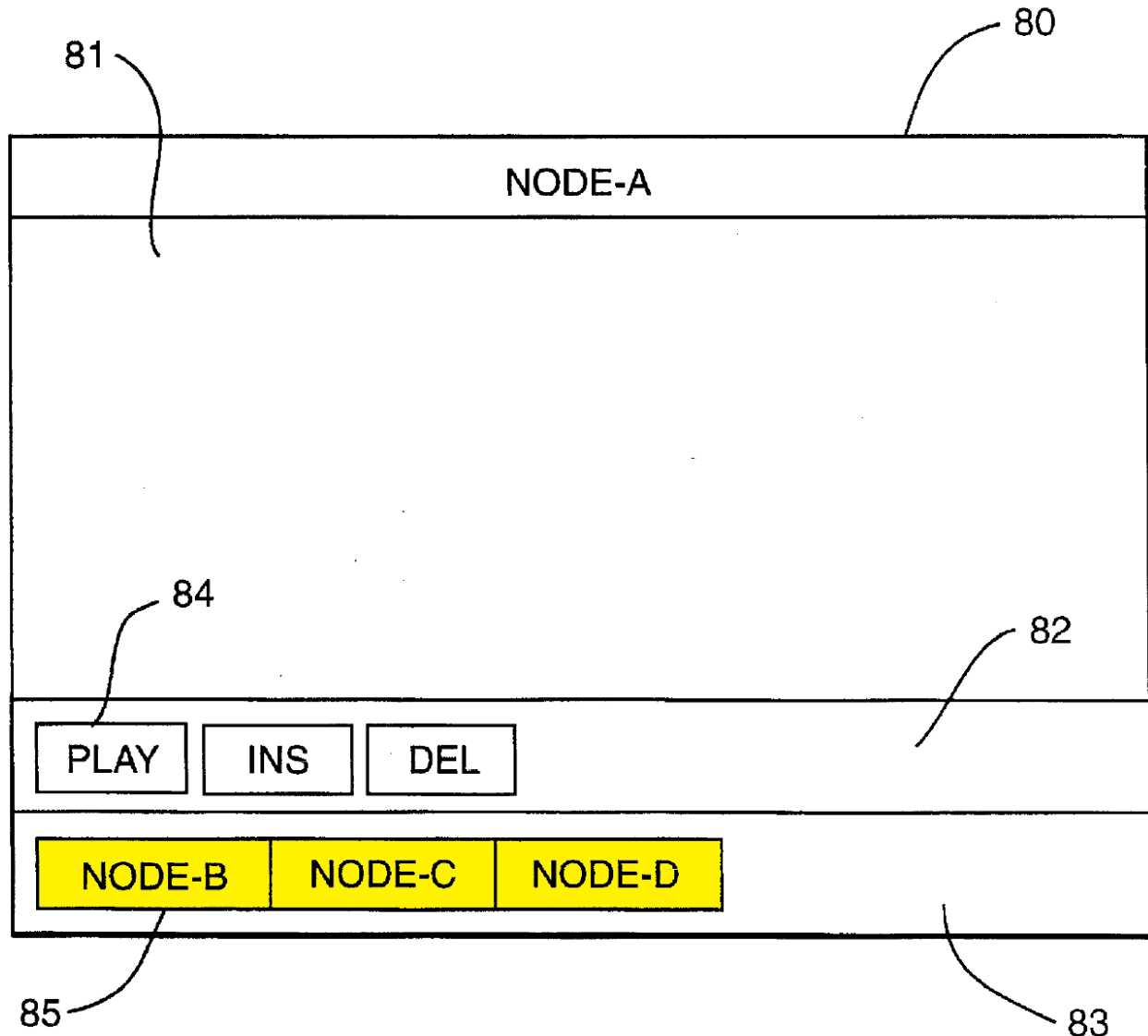


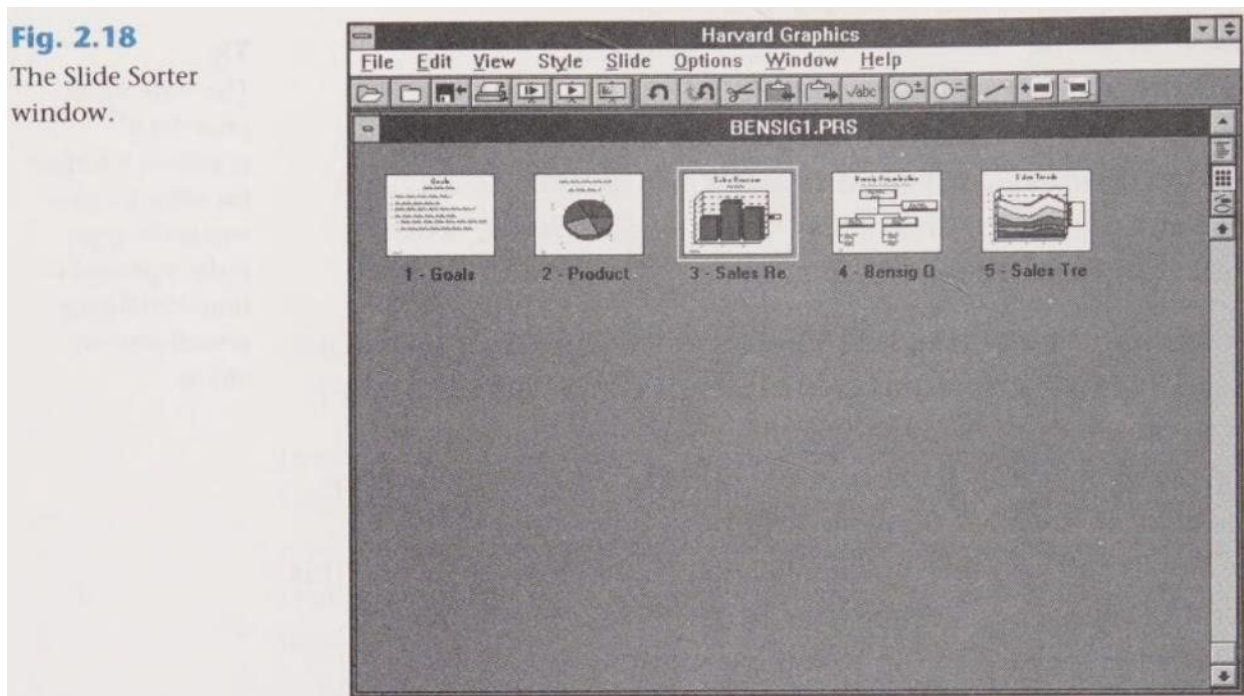
FIG. 8

(Shimizu, Fig. 8 (highlighting added).) With reference to Figure 8, Shimizu explains that “[t]he order display area **83** displays, in a left-justified manner and according to the link information order, icons **85** representing each of the nodes linked starting

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

from the current node. The icons **85** show in a list format the links connected to the current node and the order in which these links are established.” (Shimizu, 7:30-35.) In this example, therefore, NODE-A is linked to NODE-B, NODE-C, and NODE-D, in that order. (Shimizu, 7:36-39.)

174. Benedict discloses similar functionality for displaying icons for nodes. More specifically, Benedict discloses a “Slide Sorter” feature that displays icons, one icon for each slide in a presentation:



(Benedict, p.44, Fig. 2.18; *see also id.*, p.43 (“Figure 2.18 shows the Slide Sorter displaying five slides. The Slide Sorter represents these slides as scaled down images.”), pp.247-248 (“Even though most text is too small to read, one advantage

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

of using the Slide Sorter is that you can see all of the information on each slide... and evaluate the slide with a global view of the information.”).) Under the proposed Shimizu-Benedict-Fox combination, therefore, the system would have been adapted to include a user interface that represents the nodes of the show structure. Accordingly, under the proposed combination, each of the resources is “**associated with a corresponding icon,**” the icon corresponding to a scaled down image as disclosed in Benedict or one of icons **85** as disclosed in Shimizu. Each icon is also “**displayed on a first section of said user’s terminal,**” under the proposed combination, because it would have been displayed in a particular portion of the user’s screen. In Figure 2.18 of Benedict above, for example, each of the icons are displayed as a rectangle in a particular part of the screen for displaying slides for the currently loaded presentation (BENSIG1.PRS). In Figure 8 of Shimizu shown above, each icon is displayed within order display area **83** on the user’s terminal.

175. I provided a complete detailed description of the motivation to combine Shimizu, Benedict and Fox in my discussion of claim 1[a] above, and those motivations continue to apply here. But additional motivations also exist as applicable to this claim limitation. For example, an ordinarily skilled artisan would have been motivated to apply the icon display and Slide Sorter user interface teachings of Shimizu and Benedict to the proposed combination because they offer

a convenient way to visualize the sequence of nodes and manipulate their order. Benedict provides an express motivation to combine by explaining that the Slide Sorter “enables you to view many slides simultaneously” (Benedict, p.43), thus allowing the user to more quickly review the nodes. Conceptually speaking, the Slide Sorter is a computer-based implementation of the longstanding technique, well-known in the art and the general public, of organizing photographic slides by selecting and ordering them for presentation using a photographic 35mm slide projector. (Sagman, p.306 (“The most common use of the Slide Sorter view is to move slides to new positions in the presentation, the same way you’d sort 35-mm slides on a light table before placing them in a slide projector.”).) An ordinarily skilled artisan would thus have regarded the Slide Sorter interface in Benedict and the icon view in Shimizu as appropriate and user-friendly ways of allowing users to organize and order the nodes in the show structure.

176. An ordinarily skilled artisan would also have appreciated that the icon view enhances user experience by providing a “bird’s eye” view of an overall presentation, thus enabling far more efficient editing and manipulation. For example, the Slide Sorter interface in Benedict enables the user to easily reorder slides by simply moving a slide to a new position using simple drag-and-drop techniques. (Benedict, p.44.) Shimizu similarly discloses that the order of the nodes

reflected by the link information can be changed by switching icon positions using the user interface (Shimizu, 7:43-48.) The proposed combination would thus have provided a clear advantage over a user interface that could only represent one slide on the screen at a given time.

(b) **“creating a show structure of nodes, each node identifying a corresponding resource by recording a sequence of said resources by dragging and dropping said icons on a second section of said user’s terminal;”
(Claim 10[b])**

177. The first part of this limitation is substantially similar to claim 1[a] (**“creating a show structure of nodes, each node identifying a resource from a plurality of accessible resources”**). As explained for claim 1[a], the proposed combination discloses **“creating a show structure of nodes, each node identifying a corresponding resource,”** because it discloses creating a set of nodes that define their associated content and their links to other nodes. As fully discussed above, the node and link information created in Shimizu identifies a set of nodes, the content associated with each node, and the links that specify the order in which the nodes are displayed. It would also have been obvious based on the further combination with Fox, as explained, that the resource could have been a web page, image and other Web resource on the Internet identified by a URL.

178. Claim 10[b] continues by specifying that the show structure of nodes is

created by “**recording a sequence of said resources by dragging and dropping said icons on a second section of said user’s terminal.**” The Slide Sorter feature in Benedict, as mentioned, includes a drag-and-drop user feature that allows the user to order the slides in a presentation to create the desired sequence:

You move slides in the Slide Sorter by selecting and dragging the slide to a new position. When you select a slide, a highlight box appears behind the slide, indicating that it is active. As you drag the slide, a *slide placeholder*, the image of a hand that points to the new location, appears between slides to indicate the new location of the slide (see fig. 2.19). When you release the mouse button, the slide appears in its new location, and the other slides adjust to make room for the relocated slide.

(Benedict, p.44 (italics in original); *see also id.*, p.43 (“You can change the order of your slides by clicking and dragging a slide to a new location.”); *see also id.*, p.443 (“If you change the order of the slides, Harvard Graphics updates the links automatically.”).)

179. Shimizu similarly discloses similar functionality for changing the order of nodes and creating new nodes at particular positions in the show structure. With respect to the ability to relocate nodes, Shimizu discloses, in reference to icons **85** shown in Figure 8 discussed above, that “a mouse may be operated (in conjunction

with a graphical user interface function) to switch the positions of the nodes B and C (i.e., node C placed to the left of node B) in the left justified order. This operation changes accordingly the order of the link information specific to the node A.” (Shimizu, 7:43-48.) Shimizu also discloses that the user can press the “INS” button shown in Figure 8, which allows the user to “select[] the node candidate for insertion from the insertable node candidate menu.” (Shimizu, 9:14-18, Fig. 12.) If the user selects a new NODE-E for insertion, for example, the user can “designate[] the position into which to insert which also allows the user to “designate the position into which to insert the selected node E.” (Shimizu, 9:20-23.) Although Shimizu does not expressly state that drag-and-drop techniques are used to perform these operations, it would have been obvious to implement the node repositioning and new node insertion functionalities of Shimizu via drag-and-drop applied to the displayed node icons, as drag-and-drop techniques were well-known as confirmed by Benedict and further discussed below in the “rationale and motivation to combine” section provided below.

180. The proposed combination therefore discloses and renders obvious creating a show structure of nodes by “**recording a sequence of said resources by dragging and dropping said icons,**” as claimed. This is because, under the proposed combination, each “slide” icon in the Slide Sorter, or node icon in Shimizu,

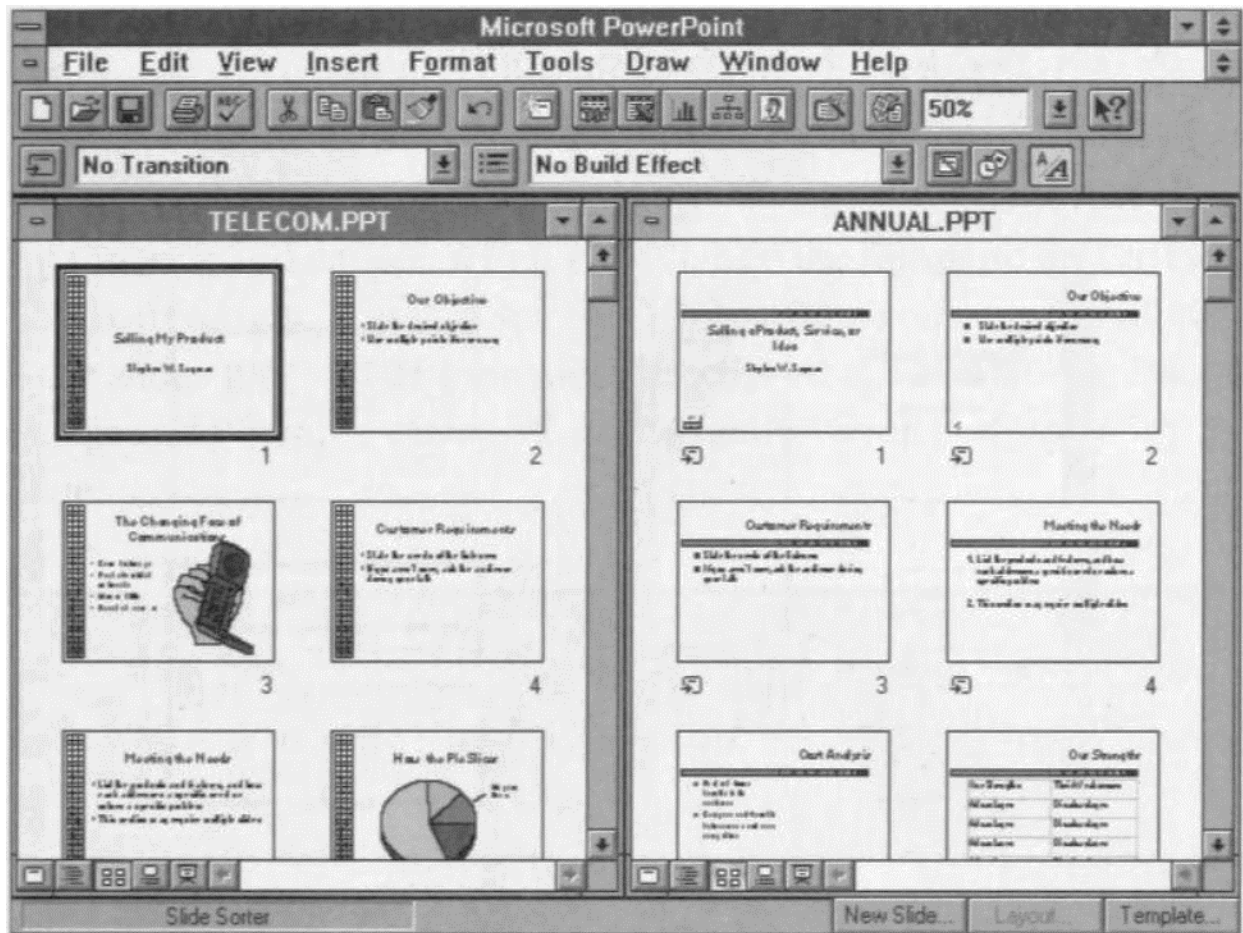
would have represented a node in the combined system, each node having one or more corresponding resources. A drag-and-drop feature would have predictably enabled the system under the proposed combination to “**record a sequence**” of nodes “**by dragging and dropping said icons,**” because the sequence of nodes would have been created and/or modified in response to drag-and-drop operations on the icons as described. Each node, as explained, is associated with one or more resources.

181. I note that it is possible that Patent Owner will argue that Shimizu and Benedict do not render obvious “**dragging and dropping said icons on a second section of said user’s terminal,**” based on the idea that the icons 85 in Shimizu and in the Slide Sorter appear within a single section of the user’s display. I would disagree with this argument because, as explained, the drag-and-drop interface in Benedict allows the user to drag a slide icon from its original location on the screen (“**a first section**”) and move it to another position (“**a second section**”). Similarly for Shimizu, each icon corresponding to a node corresponds to a distinct section of the screen. The claim does not appear to require “**a second section**” that is physically separated or divided from “**a first section.**” I also note that Shimizu discloses a “node candidate menu” for inserting new nodes as discussed (9:11-37), and an ordinarily skilled artisan would have understood that the menu would have

understood the menu to represent a different section of the screen than display area
83 that lists the node icons.

182. Nevertheless, even if the claim were interpreted in a narrow fashion that Patent Owner argued is not disclosed by Shimizu or Benedict, such a narrow reading of this limitation would have been obvious based on the further combination with **Sagman**. Sagman describes a software program known as Microsoft PowerPoint that provided functionality similar, in some respects, to the Harvard Graphics product described above. PowerPoint allowed users to create a presentation including a series of slides. (Sagman, pp.6-7.) Sagman discloses a “Slide Sorter” feature similar to the Harvard Graphics “Slide Sorter,” but with an additional feature – the ability to show two separate side-by-side presentations on the screen such that slides can be moved or copied using drag-and-drop:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1



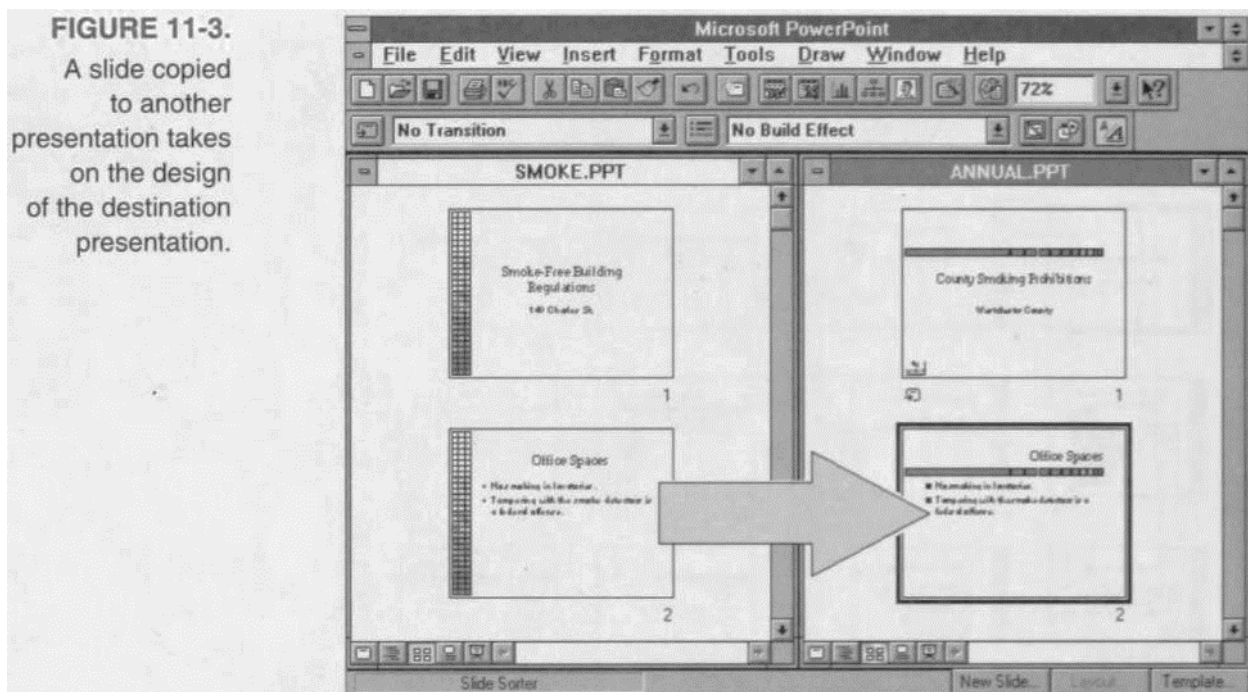
(Sagman, p.311.) The screen display above shows two PowerPoint presentations (TELECOM.PPT and ANNUAL.PPT) displayed side-by-side in two windows, with icons representing the individual slides in each presentation. Sagman explains that, using standard drag-and-drop techniques, the user can drag a slide from one presentation into the other:

After the presentations are arranged side by side, you can move a slide from one presentation to another by clicking on a slide in one presentation and dragging it to the other presentation. The moved slide

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

takes on the design of the destination presentation's template. To copy a slide rather than move it, hold down the Ctrl key as you drag the slide to the destination presentation. In Figure 11-3 on the next page, the bottom slide has been copied from one presentation to another. Note its appearance in each presentation.

(Sagman, p.313.) Figure 11-3, referenced above, shows the result of copying a slide from one presentation into another:



(Sagman, Fig. 11-3, p.312.) In this example, the slide “Office Spaces” was copied from SMOKE.PPT into the second position of the ANNUAL.PPT presentation. As shown, the slide in the target presentation retains the same content as the original slide but takes on the design of the target presentation.

183. *Rationale and Motivation to Combine (Shimizu, Benedict and Fox with Sagman):* It would have been obvious to combine the system based on Shimizu, Benedict, and Fox, with the above-described disclosures from Sagman. This would have predictably resulted in the Shimizu-Benedict-Fox system in which the **“show structure of nodes”** would have been created by dragging-and-dropping icons representing nodes from an existing show structure (in **“a first section of the user’s terminal”**) into the show structure being created (to **“a second section of the user’s terminal”**). This would thus have predictably resulted in **“recording a sequence of said resources by dragging and dropping said icons on a second section of said user’s terminal,”** using these techniques, as the user would have dragged the icons to the desired position in the show structure being created.

184. Sagman is an analogous reference in the same field of techniques for providing organized or sequenced presentation of content, and would have been reasonably pertinent to a number of problems facing the inventor. For example, Sagman would have been reasonably pertinent to the problem of how to design a user interface to organize and record a sequence of node (each node having corresponding resources), and how to use known drag-and-drop techniques to allow the user to carry out these functionalities using conventional user interface techniques. Sagman also discloses techniques for pausing the display of an

automatic slide presentation (Sagman, p.395, Table 16-1), which would have been reasonably relevant to other problems facing the inventor, such as how to implement the pausing features recited in other challenged claims.

185. An ordinarily skilled artisan would have been motivated to adapt this feature from Sagman to the Shimizu-Benedict-Fox combination in order to improve user experience. This is because dragging and dropping nodes (slides) from one presentation into another would have facilitated a more rapid creation of the claimed **“show structure of nodes.”** Sagman also provides an express motivation to combine by explaining that this drag-and-drop technique “lets you transfer an existing slide to a new presentation rather than recreate its content.” (Sagman, p.312.) This would have enabled the user to more rapidly create a new show structure of nodes by moving or copying existing nodes from another show structure into the show structure being created.

186. Like Sagman, Benedict discloses the ability to move or copy slides from one presentation to another but uses Cut, Copy, and Paste functions through on-screen menus instead of drag-and-drop techniques. (Benedict, pp.252-255.) An ordinarily skilled artisan would have been motivated to improve Benedict by using drag-and-drop to perform this function. The ability to drag-and-drop objects was a standard feature of conventional graphical user interfaces and employed in many

applications such as the Microsoft Office suite and Harvard Graphics as explained. Sagman further confirms that it was well-known to drag-and-drop objects between different windows on the same screen. Drag-and-drop techniques also had advantages over other known user interface techniques (such as pop-up or pull-down menus). For example, drag-and-drop techniques allowed users to move on-screen objects using gestures that more closely mimic how human beings manipulate objects in the real world, which made drag-and-drop techniques exceedingly easy to learn and use and a particularly desirable choice for user interface design. An ordinarily skilled artisan would have been well aware of these benefits.

187. An ordinarily skilled artisan would have had at least a reasonable expectation of success, and could have implemented the proposed combination using conventional programming techniques. Techniques for creating and displaying icons on a user interface, and for allowing the user to manipulate and reposition those icons using drag-and-drop techniques, was well-known and a standard fixture of popular operating systems by the mid-1990s, such as Microsoft Windows, IBM OS/2, Apple Macintosh OS, the X Window system available for Unix, among others. These operating systems also offered standard and well-documented APIs for accessing these drag-and-drop features and incorporating them into applications. The disclosures of Benedict and Sagman would have provided more than sufficient

details to have enabled an ordinarily skilled artisan to implement the user interface features I am citing for this claim limitation, without undue experimentation.

- (c) **“without requiring user input, automatically accessing each of said resources identified by each of said nodes in accordance with said show structure;” (Claim 10[c])**

188. This limitation is substantially similar to claim 1[b] (**“without requiring user input, automatically accessing a plurality of said resources each of said resources identified by each of said nodes”**), discussed in detail above. As explained, under the proposed combination, the resources would have been accessed and presented on the user’s computer in continuous sequential fashion, one after another, without requiring any input from the user beyond an initial click of a PLAY button or other control that commences the node display process. The resources would also have been accessed **“in accordance with said show structure,”** because the resources are identified by the node information and are presented in the sequence specified by the corresponding link information.

- (d) **“automatically retrieving a content corresponding to each of said accessed resources; and” (Claim 10[d])**
- (e) **“displaying each of said retrieved contents automatically in accordance with said show structure.” (Claim 10[e])**

189. These limitations add little to claim 10[c], as the claimed automatic

retrieval and display of content occur as part of the process of automatically accessing it. (Shimizu, 8:49-50 (“Step 107 accesses and reads the entity information linked by the entity information pointer of the node information structure... When the entity information (character string, image, sounds, motion pictures) is acquired from the node information entity management part 8, the content of the acquired information is reproduced and displayed in the node content display area **81** of the window **80** by applying the program which is associated with the pointer in the display procedure slot **62** of the node information structure **9**.”).) The proposed combination therefore discloses “**automatically retrieving a content corresponding to each of said accessed resources,**” *i.e.*, automatically retrieving the node content, and “**displaying each of said retrieved contents automatically in accordance with said show structure,**” *i.e.*, automatically displaying the retrieved content in the order as specified in the show structure.

2. Claim 11

190. Claim 11 is substantially similar to claim 2, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 11	Claim 2
<u>The method in accordance with claim 10, wherein said step of creating further comprises the step of providing an interactively variable duration information, representing the duration within which a corresponding content to said resource is being displayed.</u>	<u>The method in accordance with claim 1, wherein said step of creating further comprises the step of providing a duration information, representing the duration within which a corresponding content to said resource is being displayed.</u>

191. As shown, other than depending from a different independent claim, claim 11 substantially similar to claim 2. Claim 11 adds that the duration information is “**interactively variable**,” but as explained for claim 1 [c], the duration information under the proposed combination satisfies this requirement.

3. Claim 12

192. Claim 12 depends from claim 10, and further recites “**the method in accordance with claim 10, wherein said step of accessing further comprises the step of downloading a file from a server residing within said communications network.**” As explained for claim 4, under the proposed combination, it would have been obvious that the acquisition and retrieval of the node content would have included at least downloading a file from a server. Under the further combination with Fox, as explained for claim 4, the web browser on the user computer would have downloaded a file (*e.g.* an HTML file, an image file, etc.) from a server on the Internet as identified by its corresponding URL.

**D. Ground 3: Claims 13-20 Are Obvious Over Shimizu in view of
Benedict, Fox, Sagman, and Filepp**

1. Claim 13

193. Claim 13 depends from claim 12, and further recites “**the method in accordance with claim 12 further comprising the step of receiving a dynamic content from a predetermined server.**” As explained for claim 4, under the proposed combination with Fox, it would have been obvious for web pages received by the web browser under the proposed combination to have further included content generated on-the-fly by the server (using CGI scripts), such as HTML or image files. (Fox, p.391, p.423 (“CGI scripts are used for doing all of the ‘cool’ stuff on the Net. There are... sites that create graphic images on-the-fly.”).) Fox provides an example in which a CGI script was used to generate an image for a web page representing the current image captured by a camera. (Fox, p.424, Fig. 15-2.) It therefore would have been obvious that the web browser would have received “**a dynamic content from a predetermined server,**” *i.e.*, content generated in response to a server identified by the URL. This qualifies as “**dynamic content**” because it was generated at the time it was requested, as opposed to static content that was generated before the request was received. An ordinarily skilled artisan would have been motivated to implement the proposed combination to receive dynamic content

because such content is likely to be more interesting to the user as compared to common (static) content shown to all users. It would have been obvious to use dynamic content in a wide variety of contexts, such as news, financial information, weather, and other areas where information regularly changes.

194. The claimed dynamic content would also have been obvious in view of **Filepp**, which provides a further disclosure and motivation to combine as to this claim limitation. Filepp discloses an “interactive computer system that enables a user to display desired information, such as news, financial and cultural information, and perform desired transactional services, such as banking and shopping, through any of a plurality of types of personal computers.” (Filepp, Abstract.) Filepp was originally assigned to Prodigy Services Company, an early provider of interactive on-line services that competed with companies such as America Online (AOL) before widespread adoption of the Web.

195. The interactive system in Filepp displays “pages” to the user, with each page divided into a series of partitions as shown in Figure 3a:

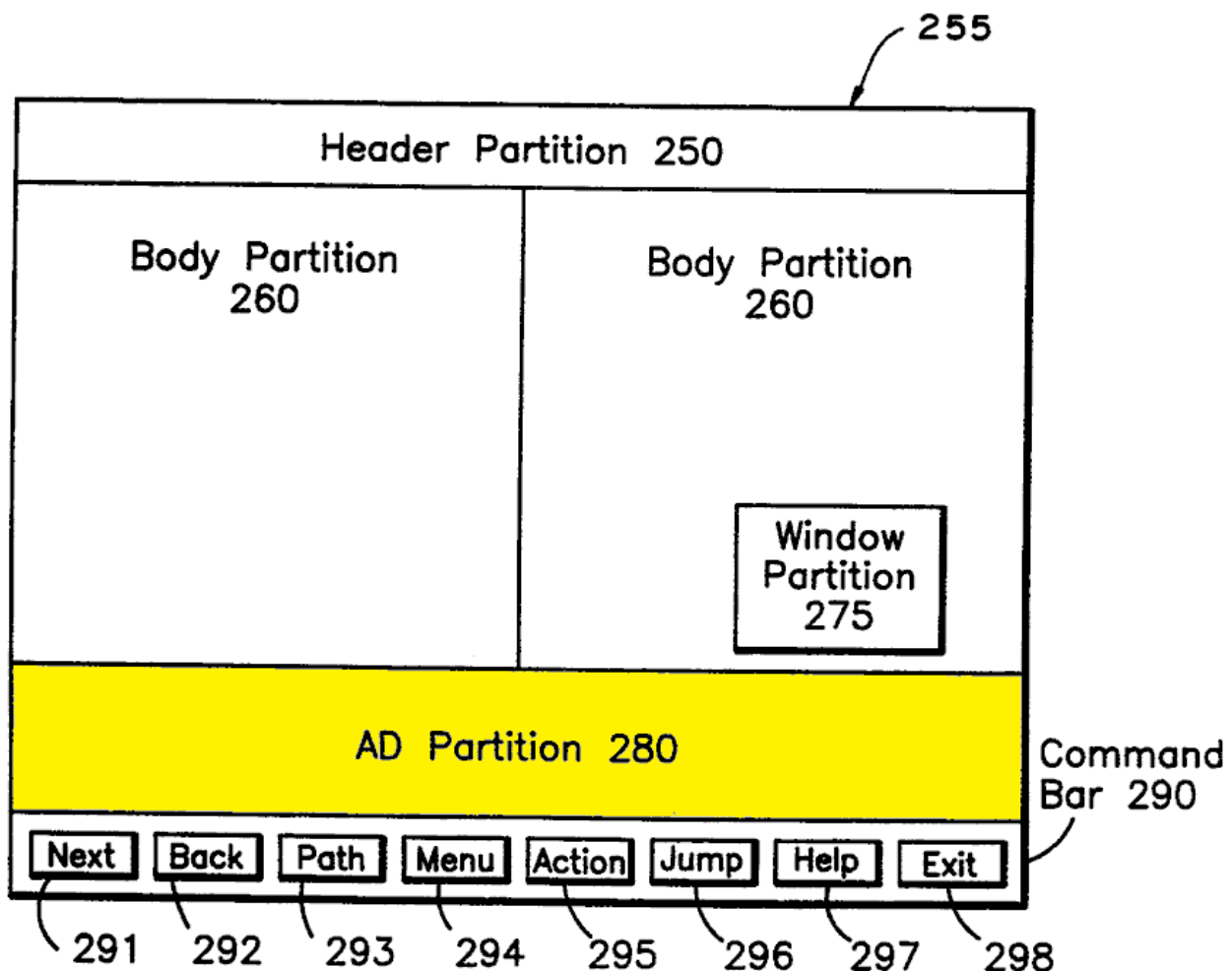


FIG. 3a

(Filepp, Fig. 3a.) “With reference to FIG. 3a, each page **255** is formatted into page partitions **250**, **260**, **280**, and **290** (not to be confused with applications partitions).”

(Filepp, 8:68-9:3.) Header partition **250** “typically conveys information on the page’s topic or sponsor,” and body partitions **260** and window partition **275** “gives the information and transactional content of the page.” (Filepp, 9:10-16.) AD

partition **280**, as its name implies, displays advertisements provided over the network for display to the user. (Filepp, 9:27-30, 9:30-34 (“Advertisements **280** may be presented to the user on an individual basis from queues of advertisements that are constructed off-line by business system **130**, and sent to file server **205** where they are accessible to each RS **400**.”), 12:64-66.)

196. Filepp discloses “**receiving a dynamic content from a predetermined server**” because the advertisements received by the system are targeted based on specific information about the user, such its past activities, demographic information, ZIP code and other targeting criteria:

Individual queues of advertisements are constructed based upon data collected on the partitioned applications that were accessed by a user, and upon events the user generated in response to applications. The data are collected and reported by RS **400** to a data collection co-application in file server **205** for later transmission to business system **130**. In addition to application access and use characteristics, a variety of other parameters, such as user demographics or postal ZIP code, may be used as targeting criteria. From such data, queues of advertisements are constructed and targeted to either individual users or to sets of users who fall into certain groups according [sic; to] such parameters.

(Filepp, 9:35-47.) As noted previously, these advertisements are accessible to the user computer (RS **400**) from file server **205**. (Filepp, 9:30-34.)

197. Filepp thus discloses another example of “**receiving a dynamic content from a predetermined server,**” *i.e.*, an advertisement selected for the user based on their past activity, demographic, or other targeting criteria. Such an advertisement qualifies as “**dynamic content**” because it was selected specifically for the requesting user based on its characteristics. Under the combination of Filepp with Fox, as noted above, advertising content could also have been generated on-the-fly, *e.g.*, an advertising image generated by a CGI server script as disclosed in Fox.⁵ (Fox, pp.423-424 (“CGI scripts are used for doing all of the ‘cool’ stuff on the Net. There are... sites that create graphic images on-the-fly.”), Fig. 15-2.)

198. ***Rationale and Motivation to Combine (with Filepp)***: It would have been obvious to combine Shimizu, Benedict, Fox, and Sagman with Filepp, in the manner described above. The combination would have predictably resulted in the system also receiving “**a dynamic content from a predetermined server,**” such as a dynamic advertisement targeted to the user based on its characteristics as disclosed in Filepp.

⁵ In my opinion, the term “**dynamic content**” does not require creation of new content for the user – it also encompasses delivery of pre-existing content dynamically selected for the user. But as explained in the text, even if the claimed dynamic content requires newly-created content, this was disclosed by and obvious in view of the teachings of Fox.

199. Filepp is an analogous reference in the field of interactive services for displaying content to users, and would have been reasonably pertinent to problems facing the inventor. For example, Filepp would have been pertinent in understanding how to select dynamic content for the user based on its user profile, including advertising content – a feature of the alleged invention. ('451, *e.g.*, 8:41-59.)

200. An ordinarily skilled artisan would have been motivated to combine for a simple reason – to make money. It was well-known long before the Internet that companies paid to display advertisements for their products or services across all forms of media (*e.g.* television, radio, newspapers, magazines, billboards, etc.). An ordinarily skilled artisan would have been motivated to adapt the Shimizu-Benedict-Fox-Sagman system to also display advertisements as disclosed in Filepp. For example, the revenue earned from such advertisements could have been used to defray the costs of developing and deploying the system or for other purposes.

201. An ordinarily skilled artisan would also have been motivated to provide “**dynamic**” content, such as targeted advertisements, because it was well-known that advertising targeted to the user based on their characteristics was more likely to be interesting to the user when compared to advertising delivered without regard to the viewer or audience. In the example in Filepp, advertisements can be selected based on (among other things) the user’s ZIP code (Filepp, 9:35-47), which would have

allowed the system to target products and/or services specific to that geographic region, thus increasing the utility and value of the advertisements.

202. An ordinarily skilled artisan would have had at least a reasonable expectation of success and could have implemented the proposed combination using conventional programming techniques. Fox itself, as explained, confirms that enabling a server to select and transmit content dynamically was a well-known use of CGI scripts. It would have been trivial to adapt the proposed combination to provide dynamic content such as targeted advertising. The disclosures of the prior art cited above would have provided more than sufficient details to have enabled an ordinarily skilled artisan to implement the proposed combination, without undue experimentation.

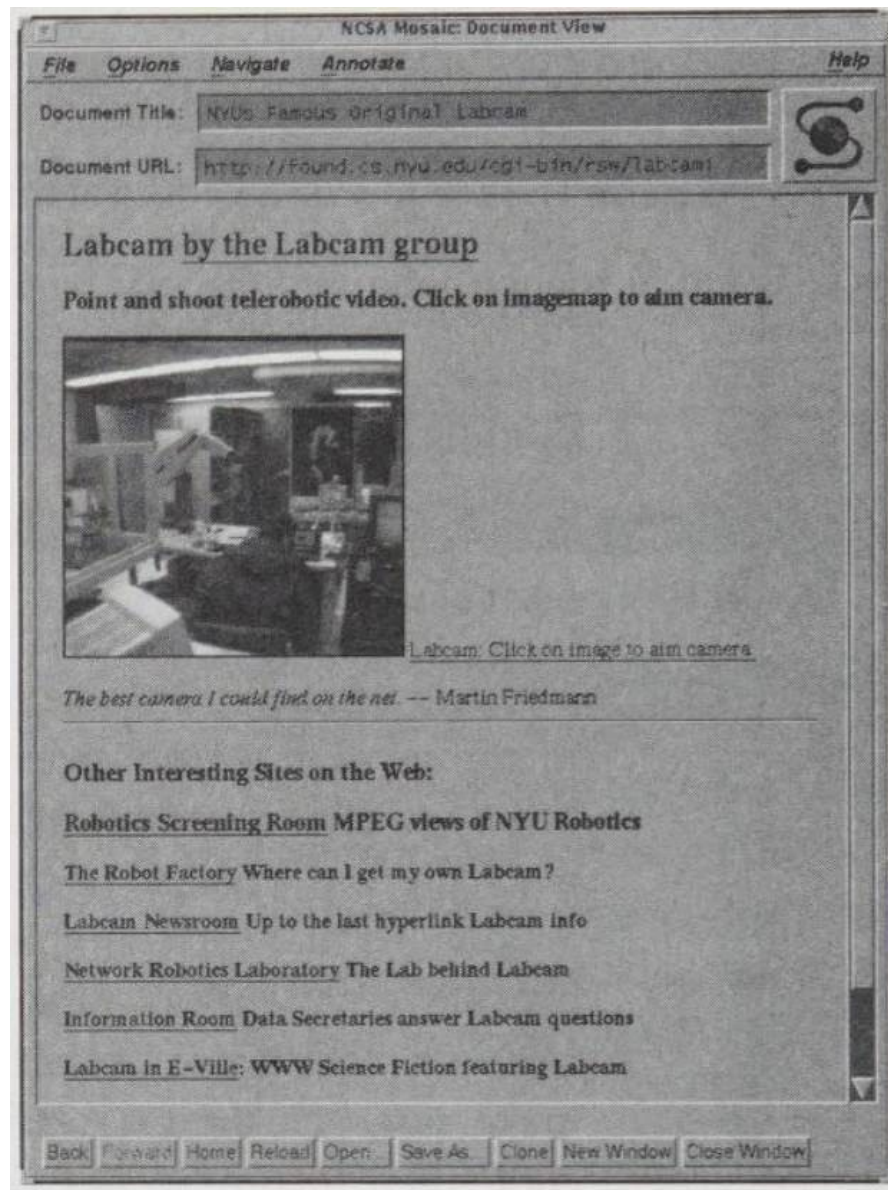
2. Claim 14

203. Claim 14 depends from claim 13, and further recites **“the method in accordance with claim 13 further comprising the step of displaying a persistent content window for receiving said dynamic content within said persistent content window.”** The ’451 patent does not expressly define **“persistent content window,”** but provides examples such as “a small window on the user’s screen” that can display content provided by a server, “such as advertising or user specific messages.” (’451, 8:44-46.) I am informed that in previous litigation involving the

'451 patent, a district court construed "**persistent content window**" as "*a designated window for displaying content to a user.*"

204. As explained for claim 13, it would have been obvious that a server under the proposed combination would have provided dynamic content. It would also have been obvious to display that content in a persistent content window on the user's display screen. For example, as explained previously, Fox provides an example in which a CGI script dynamically generates an image for a web page representing the current image captured by a camera. (Fox, p.424, Fig. 15-2.) Figure 15-2 shows that this dynamic image is presented in a fixed portion of the screen:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1



(Fox, p.424, Fig. 15-2.) In this example, the current image from the camera (in this case the inside of a lab) is presented within a designated window within the web browser. Fox thus discloses and renders obvious the receipt of dynamic content within a “**persistent content window**,” as claimed.

205. It is possible that Patent Owner will argue that “**persistent content**

window” requires a physically separated or partitioned area of the display. The ’451 patent does not define the word “**window**” or provide an example of the claimed persistent content window. A person of ordinary skill in the art would have understood “**window**” to merely refer to an area or portion of the display screen that can contain specific information. (See Microsoft Press Computer Dictionary (2d ed. 1994), EX1020, pp.417 (“**window** In applications and graphical interfaces, a portion of the screen that can contain its own document or message.”).)

206. But in the event the term “**window**” is construed narrowly, the claimed persistent content window would also have been obvious in view of **Filepp**.

207. As noted above, the interactive system in Filepp displays “pages” to the user, with each page divided into a series of partitions as shown in Figure 3a:

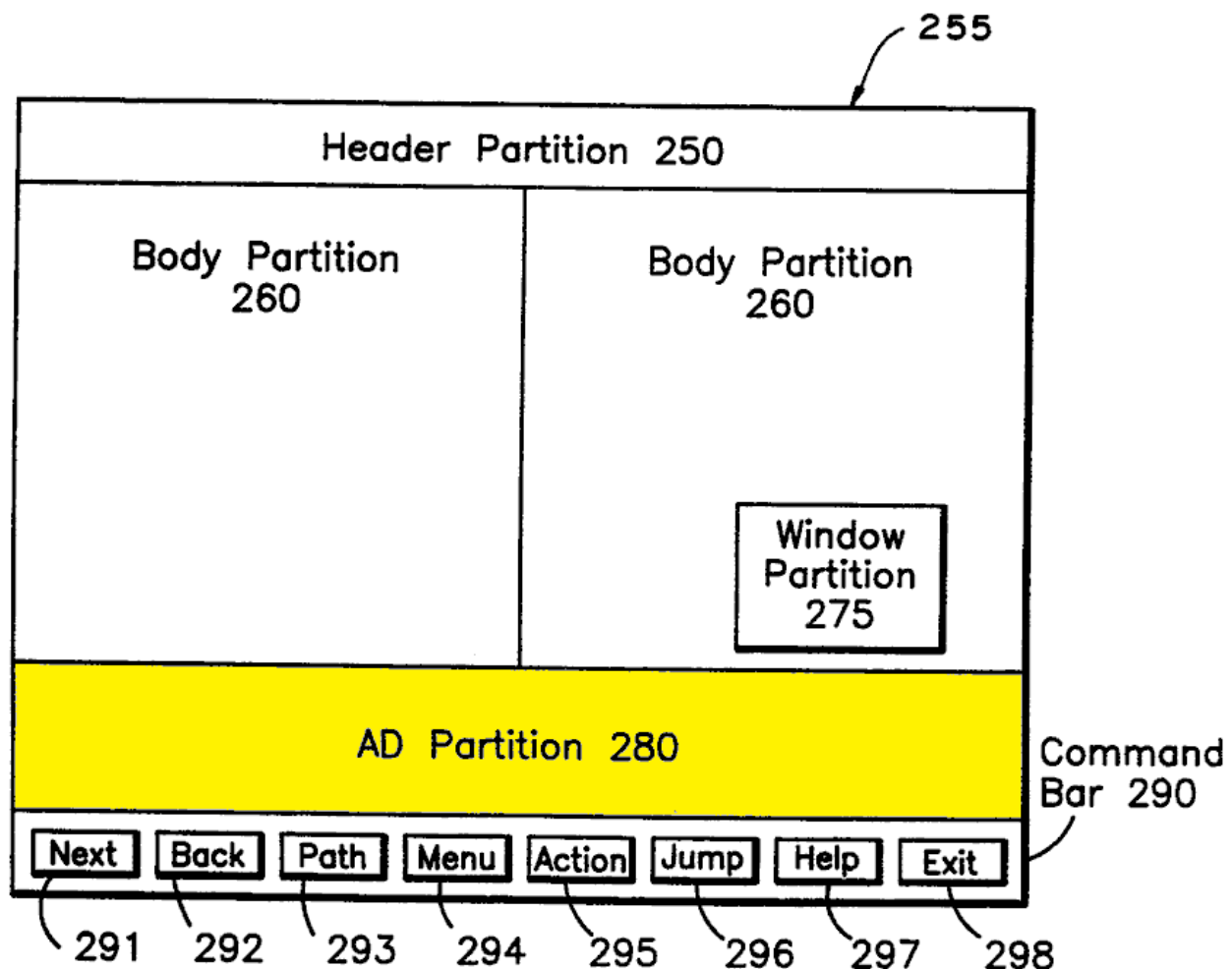


FIG. 3a

(Filepp, Fig. 3a (highlighting added).) “With reference to FIG. 3a, each page **255** is formatted into page partitions **250**, **260**, **280**, and **290** (not to be confused with applications partitions).” (Filepp, 8:68-9:3.) Header partition **250** “typically conveys information on the page’s topic or sponsor,” and body partitions **260** and window partition **275** “gives the information and transactional content of the page.”

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

(Filepp, 9:10-16.) AD partition **280**, as its name implies, displays advertisements provided over the network. (Filepp, 9:27-30, 9:30-34 (“Advertisements **280** may be presented to the user on an individual basis from queues of advertisements that are constructed off-line by business system **130**, and sent to file server **205** where they are accessible to each RS **400**.”), 12:64-66.) For purposes of this Declaration, I have identified AD partition **280** as “a **persistent content window**,” which provides a designated window on the screen devoted to the display of a particular type of content (*i.e.* advertising). This is consistent with the ’451 patent which, as noted, provides an example of a persistent content window as “a small window on the user’s screen” that can display content “such as advertising or user specific messages.” (’451, 8:44-46.) The AD partition **280** is also “**persistent**” because it appears in a designated and consistent place on the screen and its contents are separate and independent of the contents of other partitions. (Filepp, *e.g.*, 9:27-47, 10:30-36.) Each partition in Filepp also has a particular program associated with it (a “TBOL program”) that has access to the partition’s data – but not the data of other partitions. (Filepp, 39:23-36, 44:59-45:4.) It therefore would have been obvious that the content of AD partition **280** in Filepp could have remained on the screen even as the content of other partitions on the screen changed.

208. It therefore would have been obvious to combine Shimizu, Benedict,
154

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Fox, and Sagman with Filepp, in the manner described above, predictably resulting in the system in which the dynamic content is received and displayed by a “**persistent content window**,” as disclosed in Filepp. Under the proposed combination, therefore, the dynamic content received from a server would have been displayed in a fixed and dedicated partition on the user’s display device. An ordinarily skilled artisan would have been motivated to provide this functionality to improve user experience by locating placing advertising information in a consistent part of the screen, thus enhancing the user’s ability to perceive it.

209. As noted for claim 13, an ordinarily skilled artisan would have been motivated to combine, and could have implemented this combination using conventional programming techniques. An ordinarily skilled artisan would have been motivated to provide a persistent content window in order to improve user experience. An ordinarily skilled artisan would have appreciated that a persistent content window would have provided greater consistency and uniformity to a graphical user interface. It was well-known to ordinarily skilled artisans that organizing different types of information on the screen by displaying them in designated and consistent locations would have allowed the user to more quickly locate information of interest. Using the example in Filepp above, AD partition **280** provides a consistent location for advertising information on the screen, allowing a

visual tool for the user to quickly identify such information and distinguish it from other types of information presented on the screen. The ability to partition a user display into separate windows or sections was an elementary feature of graphical user interfaces (such as Microsoft Windows), and within the Web context, the ability to create a persistent content window within a page was a built-in feature of the HTML frames (<FRAMESET>) feature that was a standard available in standard HTML before the priority date. An ordinarily skilled artisan would have found the disclosures of the prior art more than sufficient to implement the claimed persistent content window, within the proposed combination, without undue experimentation.

3. Claim 15

210. Claim 15 depends from claim 13, and further recites “**the method in accordance with claim 13 further comprising the step of inserting resource identifiers interspersed within said show structure relating to said dynamic content.**” Claim 15 adds nothing of patentable significance as it merely identifies known well-known functionality of using URLs (“resource identifiers”) to identify Web resources.

211. Fox explains that “[a] *URL (or Universal Resource Locator)* is the address of a document, file, or other Internet resource... The World Wide Web gives every article, newsgroup, database, computer, file, or other Internet fragment its very

own URL.” (Fox, p.13 (*italics in original*).) To include a graphic image on a web page, for example, the HTML of the web page may include a URL identifying the file for the image. (Fox, pp.326-327.) Fox provides the following exemplary HTML: (Fox, p.327.) This exemplary HTML tells the web browser to retrieve the graphical image file identified by the URL, in this case, a file called “Jupiter.gif” from the “astronomy/planets” directory at the “www.smartpants.edu” server on the Web. The URL in this case provides an example of a “**resource identifier**.”

212. Interspersing of URLs to identify Web resources was basic knowledge of ordinarily skilled artisans. Under the proposed combination, therefore, it would have been obvious to “**insert[] resource identifiers interspersed within said show structure relating to said dynamic content.**” As explained for claim 1[a], the show structure corresponds to a set of nodes and their links or connections to other nodes, based on the combination of Shimizu, Benedict and Fox. And in the context of Web resources, as noted, these nodes would have identified web resources (*e.g.*, pages, images, etc.) via URLs. It therefore would have been obvious to insert (“**intersperse**”) URLs within the show structure to identify the dynamic content. (Fox, p.391, p.423 (“CGI scripts are used for doing all of the ‘cool’ stuff on the Net. There are... sites that create graphic images on-the-fly.”).) Fox provides an example

of a CGI script used to generate a dynamic image for a web page representing the current image captured by a camera. (Fox, p.424, Fig. 15-2.) The URLs for these resources (“**resource identifiers**”) would have been “**interspersed within said show structure**” because, under the proposed combination, they would have been stored within the show structure, *e.g.*, via the entity information pointer slot **63** of Shimizu as fully described for claim 1[a].

4. Claim 16

213. Claim 16 depends from claim 15, and further recites “**the method in accordance with claim 15 wherein said dynamic content is an advertising message.**” As explained for claims 13 and 14, under the further combination with Filepp, it would have been obvious that the dynamic content would have been an advertising message.

5. Claim 17

214. Claim 17 depends from claim 15, and further recites “**the method in accordance with claim 15 further comprising the step of providing said dynamic content in accordance with said user’s profile.**” As explained for claim 13, under the further combination with Filepp, targeted advertisements dynamically selected for the user (“said dynamic content”) were based on specific information about the user such past activities, demographic information, ZIP code and other

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

targeting criteria:

Individual queues of advertisements are constructed based upon data collected on the partitioned applications that were accessed by a user, and upon events the user generated in response to applications. The data are collected and reported by RS **400** to a data collection co-application in file server **205** for later transmission to business system **130**. In addition to application access and use characteristics, a variety of other parameters, such as user demographics or postal ZIP code, may be used as targeting criteria. From such data, queues of advertisements are constructed and targeted to either individual users or to sets of users who fall into certain groups according [sic; to] such parameters.

(Filepp, 9:35-47.) This renders obvious “**providing said dynamic content in accordance with said user’s profile,**” as claimed, because the targeted advertisements (“**dynamic content**”) is provided in accordance with the user’s interests (based on the previous information they accessed) and characteristics (such as demographics, ZIP code, and other criteria). This is consistent with the ’451 patent, which identifies similar information as examples of user profile information. (’451, 8:58-62 (“Such dynamic content is usually based on, among other things, the user’s profile. For example, some of the factors relating to a user profile, which are considered include a particular sports team preference; a demographic; or a pre-set profile as specified in the software.”).)

6. Claim 18

215. Claim 18 is substantially similar to claim 5, which was discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 18	Claim 5
<u>The method in accordance with claim 17 further comprising the step of storing said content for each resource accessed.</u>	<u>The method in accordance with claim 4 further comprising the step of storing said data file for each resource accessed.</u>

216. As shown, other than depending from different claims, claim 18 is substantially similar to claim 5, with the sole difference being that claim 18 recites storing said “**content**” whereas claim 5 recites formatting said “**data file.**” There is no meaningful difference as the content recited claim 18 under the proposed combination is stored in the data files as recited in claim 5.

7. Claim 19

217. Claim 19 is substantially similar to claim 7, which was discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 19	Claim 7
<u>The method in accordance with claim 18 wherein said step of displaying further comprises the step of</u>	<u>The method in accordance with claim 6 wherein said step of displaying further comprises the step of reformatting said</u>

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Claim 19	Claim 7
<u>reformatting said content in accordance with a predetermined arrangement.</u>	<u>data files in accordance with a predetermined arrangement.</u>

218. As shown, other than depending from different claims, claim 19 is substantially similar to claim 7, with the sole difference being that claim 19 recites reformatting said “**content**” whereas claim 7 recites formatting said “**data files.**” There is no meaningful difference as the content recited claim 19 under the proposed combination is stored in the data files as recited in claim 7.

8. Claim 20

219. Claim 20 is substantially similar to claim 8, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 20	Claim 8
<u>The method in accordance with claim 18 further comprising the step of displaying control buttons for controlling the display of said contents.</u>	<u>The method in accordance with claim 7 further comprising the step of displaying control buttons for controlling the display of said data files.</u>

220. As shown, other than depending from different claims, claim 20 is substantially similar to claim 8, with the sole difference being that claim 20 recites display of said “**contents**” whereas claim 8 recites display of said “**data files.**” As explained for claim 19, there is no meaningful difference as the contents recited

claim 20 are stored in the data files as discussed for claim 8.

E. Ground 4: Claims 22-25 Are Obvious Over Shimizu in view of Benedict, Fox, and Guinan

1. Independent Claim 22

221. For reference, claim 22 recites:

22. A method for transmitting in a sequential arrangement to a user's computer, content derived via a communications network from a plurality of resources the method comprising the steps of:

- [a] performing an on-line search, in response to which a plurality of search results and their corresponding URL resources are received;
- [b] automatically creating and storing a show structure of nodes, each node identifying a resource from a plurality of accessible resources via said communications network wherein said show structure is created in response to said search results received in response to said on-line search;
- [c] without requiring user input accessing each of said resources identified by each of said nodes;
- [d] retrieving a content corresponding to each of said accessed resources; and
- [e] automatically delivering said content to said user for display on said user's computer in accordance with said show structure.

(’451, 19:61-20:13 (Claim 22; bracketed notation (e.g., [a]) added).) Many of the limitations of claim 22 are substantially similar to limitations recited in claims 1 and 10 that were covered above. In the interests of brevity and consistency, therefore, I will refer back to my previous analysis where appropriate.

222. As I will show, the primary difference between claim 22 and the other independent claims addressed above is in how nodes are identified for creating the claimed “**show structure of nodes.**” In claim 22, as shown below, these nodes are identified based on the results of an on-line search.

223. The preamble recites “[a] method for transmitting in a sequential arrangement to a user’s computer, content derived via a communications network from a plurality of resources...” It is unclear if the preamble recites a claim limitation, but even if it does, it is disclosed by the prior art.

224. I note that the preamble of claim 22 is substantially identical to the preamble of claim 10, except that claim 22 recites “**transmitting**” content to a user’s computer whereas claim 10 recites “**displaying**” content on a user’s terminal. This presents no material difference because, as explained above, under the proposed combination content is transmitted to the user’s computer over a network in order to be displayed. The proposed combination thus renders obvious the preamble.

(a) **“performing an on-line search, in response to which a plurality of search results and their corresponding URL resources are received;” (Claim 22[a])**

225. Shimizu and Benedict do not expressly disclose performing an on-line search to identify resources. I am informed that a district court in prior litigation involving the '451 patent construed **“on-line search”** as *“Internet search.”*

226. Claim 22[a] would have been obvious in view of Fox and Guinan, which together confirm that performing an on-line search of hypertext resources was well-known to ordinarily skilled artisans. As Fox explains:

Search Indexes

Hypertext makes it easy to get lost. That's part of the appeal. You can start out reading about Sperm Whales and end up reading about the Prince of Wales. But what if you want to find pages dealing with a specific topic?

Luckily, the Web is full of indexes. An index is a special Web page that allows you to type in a search term or keyword. The index then spits out a list of Web pages matching your search. You can then click on any page that catches your interest and transfer there automatically...

Many indexes allow you to search the full contents of thousands of Web pages. Some indexes have a series of forms or menus you can customized, allowing you to find exactly what you want.

(Fox, pp.10-11.) Fox identifies more than 10 existing search indexes, including the

Lycos and WebCrawler search engines that were popular by the mid-1990s. (Fox, pp.11-12, Table 1-1.) But Fox does not describe how results from an on-line search are used beyond being displayed by the user. I have therefore cited Fox in combination with Guinan, which discloses a technique for (a) allowing a user to enter a natural language search query and then (b) searching for hypertext nodes in response to the query to create a “guided tour” for the user. (Guinan, pp.124-125.)

227. Guinan explains “[i]n using any hypertext system an end-user or browser will encounter many technical problems that have been well-documented elsewhere,” such as “disorientation and the quality of the information retrieved by the user from the hypertext.” (Guinan, p.122, §1.) This problem with hypertext was well-known and, as acknowledged in Fox, one of the key reasons the industry adopted on-line searching in the Web context as discussed above. (Fox, pp.10-11 (“Hypertext makes it easy to get lost... [W]hat if you want to find pages dealing with a specific topic? Luckily, the Web is full of indexes.”).)

228. Guinan states that, to address problems of hypertext systems:

We propose a method for dynamically generating a guided tour through a hypertext in response to a user’s query which will plan a tour through a hypertext covering all the nodes the system believes relevant to the user’s information need as expressed in a natural language query, which eases disorientation problems by using the guided tour metaphor... and

which takes advantage of typing of information links in the hypertext to present the nodes on the tour in a logical fashion. Effectively we extend the guided tour metaphor by including the information retrieval features... and we add in logical sequencing of nodes on the tour to complete the process.

(Guinan, p.123, §1 (Introduction).)

229. The first step of Guinan's proposed method is for the user to enter a "natural language" search query. (Guinan, p.124, §3.) The hypertext nodes are then searched for keywords matching the query, which are scored using a weighting algorithm. (*Id.*; *see also id.*, p.129, §6 ("*Finding information*: When the user enters a search query in the form of a natural language statement it is matched to the nodes in the hypertext using standard information retrieval techniques.") (italics in original).) Guinan explains that, in weighting hypertext nodes, search terms that appear less frequently throughout the collection are given more weight using known TF-IDF techniques,⁶ and as such, hypertext nodes containing more of these less

⁶ Guinan uses a weighting technique referred to as TF-IDF, an acronym for term frequency-inverse document frequency. (Guinan, p.124.) TF-IDF is a well-known technique for assessing the importance of terms appearing in documents. The basic idea behind TF-IDF is that terms appearing less frequently in documents are more likely to be important to the documents than terms that appear more frequently. TF-IDF will thus assign higher weights to these infrequently appearing terms.

frequent terms are given higher weights. (*Id.*) Hypertext nodes are also weighted based on the number of connections they have to other nodes. (*Id.*) Nodes whose weights satisfy a threshold value are included in the guided tour. (Guinan, p.125 (“All nodes whose weight is greater than or equal to this cut-off or threshold value would be selected for inclusion in the tour.”).)

230. Once matching hypertext nodes have been identified, a guided tour is assembled. “At this stage, the nodes for the guided tour have been dynamically chosen in response to a user’s query and some form of logical ordering has to be imposed on the selected tour nodes.” (Guinan, p.125, §3.) Guinan discloses an ordering procedure that takes into account linking relationships between the matching nodes. (*Id.*) “This procedure is followed until every node selected for inclusion in the tour has been positioned correctly relative to each other node in the tour.” (*Id.*) Guinan thus discloses using a search to identify hypertext nodes for creating a guided tour for presenting those nodes.

231. ***Rationale and Motivation to Combine (Shimizu, Benedict and Fox with Guinan)***: I already provided a full analysis of the manner and motivation to combine with respect to Shimizu, Benedict, and Fox in my discussion of claim 1 above, and that analysis applies here. With respect to claim 22[a], it would have been obvious to combine these references with Guinan, predictably resulting in the

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Shimizu-Benedict-Fox hypertext system adapted to use an “**on-line search**” of the Internet to identify a series of hypertext nodes.

232. These nodes, as explained in the next limitation, are then used to create the claimed “**show structure of nodes**” under the proposed combination.

233. Guinan does not expressly disclose that the hypertext nodes in the guided tour are located using “**on-line search**” to the extent this term is interpreted as requiring an Internet search. And because Guinan does not disclose the use of the Web, it also does not disclose that the search results are accompanied with “**corresponding URL resources.**” But these details would have been trivial and obvious in view of Fox which, as mentioned above, expressly discloses the use of on-line search indexes accessible over the Web. As noted below in my discussion of the motivations to combine, moreover, Guinan was expressly designed to solve problems with hypertext systems – and the Web is a hypertext system. It would also have been obvious that the search results obtained by Guinan, under the combination with Fox, would have been accompanied by the URLs associated with the located hypertext nodes. This is because, as was elementary knowledge to ordinarily skilled artisans, resources on the Web are identified using URLs. (Fox, p. 13 (“A *URL* (or *Universal Resource Locator*) is the address of a document, file, or other Internet resource... The World Wide Web gives every article, newsgroup, database,

computer, file, or other Internet fragment its very own URL.”).) It thus would have been obvious to adapt the hypertext searching and guided tour techniques of Guinan, predictably resulting in an “**on-line search**” that receives hypertext nodes as search results as well as their corresponding “**URL resources**,” which includes at least their URLs.

234. Guinan is an analogous reference in the same field of providing organized or sequenced presentation of hypertext content, and would have been reasonably pertinent to a number of problems facing the inventor. For example, Guinan would have been reasonably pertinent to the problem of how to use searching techniques to create a list of hypertext nodes for creating a show structure.

235. The motivation to combine would have been straightforward. Guinan provides an express motivation to combine by explaining that, using her techniques for creating a guided tour based on a search, “[u]sers don’t need to worry about getting lost or confused in the hypertext as all they need to do is follow the tour which is generated in response to their query.” (Guinan, p.129, §6.)

236. This motivation is also expressly echoed in Fox, which also identifies the use of searching techniques to help users locate specific information and avoid getting lost. (Fox, pp.10-11 (“Hypertext makes it easy to get lost... [W]hat if you want to find pages dealing with a specific topic? Luckily, the Web is full of

indexes.”).) The fact that Guinan and Fox both relate to hypertext systems (the Web being a hypertext system as discussed above), and that both address a common problem and identify search as the solution, reinforces their combinability and provides a further motivation to combine.

237. An ordinarily skilled artisan would thus have been motivated to use the searching techniques in Guinan and Fox to more easily identify relevant hypertext nodes for the user. These nodes, as explained below, are in turn used to create the claimed “**show structure of nodes**” that is delivered and displayed to the user under the proposed combination. As a further motivation to combine, I note that the “guided tour” of hypertext nodes in Guinan is directly analogous to the show structure of nodes under the proposed combination. It therefore would have been obvious to use the search results provided by Guinan – which Guinan placed into an ordered and sequenced guided tour – as the basis for creating the claimed show structure of nodes under the proposed combination

238. An ordinarily skilled artisan would also have been motivated to adapt the teachings of Guinan to an Internet search index that identifies search results with URLs, as disclosed in Fox. An ordinarily skilled artisan would have found this to have been a natural adaptation considering that, as Fox confirms, the Web is an example of a hypertext-based system. (Fox, p.9 (“The World Wide Web uses

hypertext to link all of its pages together.”).) An ordinarily skilled artisan would have been motivated to adapt Guinan for the Web to take advantage of the commercial growth and growing popularity of the Web, as fully explained for claim 1, and to expand the audience for her guided tour the universe of content that could be incorporated into the tour.

239. An ordinarily skilled artisan would have had at least a reasonable expectation of success and could have implemented the proposed combination using conventional programming techniques. The ability to search through hypertext documents was well-known, as evidenced by Guinan and the many commercially-available examples of Internet search indexes in Fox. Keyword searching techniques were well-known decades before the priority date, with the TF-IDF term weighting techniques in Guinan dating back to at least the 1970s. And as explained for claim 1, the use of URLs to identify resources on the Web was well-known. The disclosures of Fox and Guinan would have provided more than sufficient details to have enabled an ordinarily skilled artisan to implement the details I have cited for this claim limitation, without undue experimentation.

(b) **“automatically creating and storing a show structure of nodes, each node identifying a resource from a plurality of accessible resources via said communications network wherein said show structure is created in response to said search results received in response to said on-line search;” (Claim 22[b])**

240. The first part of claim 22[b] is substantially similar to claim 1[a] (**“creating a show structure of nodes, each node identifying a resource from a plurality of accessible resources”**). As explained for claim 1[a], the proposed combination discloses **“creating a show structure of nodes, each node identifying a corresponding resource,”** because it discloses creating a set of nodes that define their associated content and their links to other nodes. As fully discussed above, the node and link information created in Shimizu identifies a set of nodes, the content associated with each node, and the links that specify the order in which the nodes are displayed. It would also have been obvious based on the further combination with Fox, as explained, that the resource could have been a web page, image and other resource identified by a URL and accessible over the Internet (**“said communications network”**). With respect to the act of **“creating”** the claimed show structure, it would have been obvious to an ordinarily skilled artisan that the show structure in Shimizu was **“creat[ed],”** as it must have been brought into existence in order to perform the node display and other functions described in

Shimizu. As explained below, the show structure would have been created and stored automatically under the further combination with Guinan. And with respect to “**storing**” the claimed show structure, Shimizu explains that the node and link information, as well as the content resources themselves, are stored. (Shimizu, *e.g.*, 7:66-8:10, 6:39-43.) It would also have been obvious that the show structure would have been stored, as it must have been stored in either volatile or non-volatile memory in order to be used (for example, to display the contents as described in Shimizu).

241. It would also have been obvious to create and store the show structure “**automatically**” and “**in response to said search results received in response to said on-line search.**” As explained for claim 22[a], Guinan discloses automatically constructing a guided tour of hypertext nodes in response to a search query. It would also have been obvious that this occurs “**automatically**” because Guinan requires no user action to build the guided tour based on the search results. It therefore would have been obvious that to provide the guided tour created by Guinan, which identifies the hypertext nodes and specifies the order in which they will be presented, to the system of Shimizu to use as node and link information for the claimed “**show structure.**” Although the user enters a search query, this is a separate and unclaimed act that precedes the claimed step of creating of a show structure. I note that the

'451 patent similarly contemplates user action in initiating searches used to create show structures. ('451, 11:67-12:3 ("In accordance with one embodiment of the invention, the RoboGuide program may prepare various show structures based on a user's search specifications.").) Because the hypertext nodes and their links were identified via the guided tour in Guinan, under the proposed combination, the user would not need to take any action beyond entry of the initial search query. Under the proposed combination of Guinan with Shimizu, Benedict and Fox, the results obtained from the on-line search would have been used to identify nodes and automatically create and store the **"show structure of nodes."** The rationale and motivation to combine as to Guinan was provided for claim 22[a].

- (c) **"without requiring user input accessing each of said resources identified by each of said nodes;" (Claim 22[c])**
- (d) **"retrieving a content corresponding to each of said accessed resources; and" (Claim 22[d])**
- (e) **"automatically delivering said content to said user for display on said user's computer in accordance with said show structure." (Claim 22[e])**

242. These limitations are substantially similar to limitations already covered in claim 1 and 10, discussed in detail above, and the analysis above applies equally here. As explained, under the proposed combination, each of the resources

identified by each of the nodes would have been automatically accessed, retrieved, and delivered to the user's computer in continuous sequential fashion, one after another, without requiring any input from the user beyond an initial click of a PLAY button or other control to commence the node display process. It would have also have been obvious that, in order to display content as disclosed in Shimizu, Benedict, and Fox, that content would have been accessed and retrieved. The resources would also have been delivered "**in accordance with said show structure,**" because the resources are identified by the node information and are presented to the user's computer in the sequence specified by the corresponding link information.

9. Claim 23

243. Claim 23 is substantially similar to claim 2, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 23	Claim 2
<u>The method in accordance with claim 22, wherein said step of storing further comprises the step of providing a duration information, representing the duration within which a corresponding content to said resource is being displayed.</u>	<u>The method in accordance with claim 1, wherein said step of creating further comprises the step of providing a duration information, representing the duration within which a corresponding content to said resource is being displayed.</u>

244. As shown, other than depending from a different independent claim,

claim 23 substantially similar to claim 2. Claim 23 modifies the step of “**storing**” whereas claim 2 modifies the step of “**creating**,” but there is no material difference as the claimed “**show structure of nodes**” is both created and stored under the proposed combination, and that structure includes the claimed duration information.

10. Claim 24

245. Claim 23 is substantially similar to claim 12, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 24	Claim 12
<u>The method in accordance with claim 22, wherein said step of accessing further comprises the step of downloading a file from a server residing within said communications network.</u>	<u>The method in accordance with claim 10, wherein said step of accessing further comprises the step of downloading a file from a server residing within said communications network.</u>

246. As shown, other than depending from a different independent claim, claim 24 substantially identical to claim 12 (which in turn is similar to claim 4). Claim 24 is thus obvious for the same reasons as claims 12 and 4 discussed above.

11. Claim 25

247. Claim 25 is substantially similar to claim 13, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 25	Claim 13
<u>The method in accordance with claim 24 further comprising the step of transmitting to said user a dynamic content.</u>	<u>The method in accordance with claim 12 further comprising the step of receiving a dynamic content from a predetermined server.</u>

248. As shown, other than depending from a different claim, claim 25 substantially identical to claim 13. The only material difference is that claim 25 recites “**transmitting to said user**” whereas claim 13 recites “**receiving... from a predetermined server**” a dynamic content. These claims merely represent the flip sides of the same process; under the proposed combination, a predetermined server transmits the dynamic content, which is received by the user computer.

F. Ground 5: Claims 26-32, 35-36 Are Obvious Over Shimizu in view of Benedict, Fox, Sagman, Guinan, and Filepp

249. Claims 26-36 are similar to claims already addressed above, for which I have further cited the Filepp reference.

1. Claim 26

250. Claim 26 is substantially similar to claim 14, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Claim 26	Claim 14
<u>The method in accordance with claim 25 further comprising the step of formatting said dynamic content for displaying within a persistent content window on said user's terminal.</u>	<u>The method in accordance with claim 13 further comprising the step of displaying a persistent content window for receiving said dynamic content within said persistent content window.</u>

251. As shown, other than depending from different claims, claim 26 substantially similar to claim 14. The primary difference is that claim 26 recites the step of “**formatting**” the dynamic content “**for displaying**” within a persistent content window, whereas claim 14 recites that the persistent window receives the dynamic content. I do not perceive any difference material to my analysis, as the step of displaying dynamic content in the persistent content window includes formatting that content for that display. As explained above for claim 14, the claimed “**persistent content window**” is disclosed and rendered obvious based on Fox and Filepp. Filepp itself discloses formatting as part of the process of displaying page content into its respective partition. (Filepp, 8:64-9:2 (“Applications, i.e., information events, are composed of a sequence of one or more pages opened at screen **414** of monitor **412**... With reference to FIG. 3a, each page **255** is formatted into page partitions **250**, **260**, **280**, and 290...”).) Under the proposed combination, it would have been obvious that the dynamic content (*e.g.* dynamic HTML or image content as explained for claim 14) would have been formatted in order to be

displayed within the persistent content window.

252. To the extent the “**formatting**” limitation of claim 26 requires some kind of visual transformation of the dynamic content, as explained for claim 7, Fox discloses the ability of the web browser to size or position content (such as text or graphic files) based on formatting parameters specifying size, location, alignment, etc., based on standard HTML tags. (Fox, p.282 (“Perhaps the most typically used style of text alignment is centering. Centered text is flashy, professional, and easy to read.”), p.292 (“If you want, you can determine the exact width [of a line] in pixels... Or even more easily, you can just determine what percentage of the screen the line should take up.”); *see also id.*, p.347 (“It’s a good idea to change an image’s size if it seems out of place, or unwieldy. You should resize images to give your Web page some uniformity.”), 333 (“Netscape lets you solve this problem with the HEIGHT and WIDTH attributes. You specify the image’s size in pixels, as follows: Netscape then uses this information to predraw the image box.”), p.335 (“If you don’t feel like calculating out the exact size of a scaled image’s height and width in pixels, you can have Netscape do the work for you. Simply specify the HEIGHT or WIDTH as a percentage of the available screen. This way, the image always appears relative to the rest of the Netscape page.”), pp.328-330 (discussing image alignment

techniques).) Claim 26 is therefore obvious.

2. Claim 27

253. Claim 27 is substantially similar to claim 15, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 27	Claim 15
<u>The method in accordance with claim 26 further comprising the step of inserting resource identifiers interspersed within said show structure relating to said dynamic content.</u>	<u>The method in accordance with claim 13 further comprising the step of inserting resource identifiers interspersed within said show structure relating to said dynamic content.</u>

254. As shown, other than depending from different claims, claim 27 is identical to claim 15. Claim 27 is therefore obvious based on the same analysis.

3. Claim 28

255. Claim 28 is substantially similar to claim 16, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 28	Claim 16
<u>The method in accordance with claim 27 wherein said dynamic content is an advertising message.</u>	<u>The method in accordance with claim 15 wherein said dynamic content is an advertising message.</u>

256. As shown, other than depending from different claims, claim 28 is

identical to claim 16. Claim 28 is therefore obvious based on the same analysis.

4. Claim 29

257. Claim 29 is substantially similar to claim 17, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 29	Claim 17
<u>The method in accordance with claim 28 further comprising the step of providing said dynamic content in accordance with said user's profile.</u>	<u>The method in accordance with claim 15 further comprising the step of providing said dynamic content in accordance with said user's profile.</u>

258. As shown, other than depending from different claims, claim 29 is identical to claim 17. Claim 29 is therefore obvious based on the same analysis.

5. Claim 30

259. Claim 30 is substantially similar to claim 18, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 30	Claim 18
<u>The method in accordance with claim 29 further comprising the step of storing said content for each resource accessed.</u>	<u>The method in accordance with claim 17 further comprising the step of storing said content for each resource accessed.</u>

260. As shown, other than depending from different claims, claim 30 is identical to claim 18. Claim 30 is therefore obvious based on the same analysis.

6. Claim 31

261. Claim 31 is substantially similar to claim 19, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 31	Claim 19
<u>The method in accordance with claim 30 wherein said step of delivering further comprises the step of reformatting said content in accordance with a predetermined arrangement.</u>	<u>The method in accordance with claim 18 wherein said step of displaying further comprises the step of reformatting said content in accordance with a predetermined arrangement.</u>

262. As shown, other than depending from different claims, claim 31 is substantially identical to claim 19. The sole difference is that claim 31 modifies the step of “**delivering**” whereas claim 19 modifies the step of “**displaying**,” but this difference is immaterial as the delivery of the content in claim 22 is “**for display on said user’s computer.**” Claim 31 is obvious for the same reasons as claim 19.

7. Claim 32

263. Claim 32 is substantially similar to claim 20, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 32	Claim 20
<u>The method in accordance with claim 31 further comprising the step of transmitting information to said user for displaying control buttons for controlling the display of said contents.</u>	<u>The method in accordance with claim 18 further comprising the step of displaying control buttons for controlling the display of said contents.</u>

264. As shown, other than depending from different claims, claim 32 is substantially similar to claim 20 (which in turn is similar to claim 8). The sole difference is that claim 32 further recites “**transmitting information to said user**” for displaying the claimed control buttons, but this difference is immaterial. It would have been obvious that (as described for claim 8) that information would have been transmitted to the user to facilitate the display of the claimed control buttons. I note that the claim does not require that the claimed information be transmitted over a network. In any case, under the further combination with Fox as explained for claim 8, it would have been obvious for the control buttons to be part of the web page downloaded (and thus transmitted to) to the user over the Internet. (Fox, p.238 (“You can include these navigation controls at the top or bottom of every page.”).)

8. Claim 35

265. Claim 35 depends from claim 32, and further recites “**the method in accordance with claim 32 further comprising the step of inserting advertisement displays between display of said contents corresponding to each**

of said nodes.” This limitation adds little to what was already discussed. I note that claim 35 depends (through several other intermediate claims) on claim 28, which recites that “**said dynamic content is an advertising message.**” As explained for claims 13-14 and 16, under the proposed combination, advertisements would have been displayed in a “**persistent content window**” on the screen. Figure 3a of Filepp shows an example of a persistent content window in the form of AD partition 280 that displays advertisements to the user:

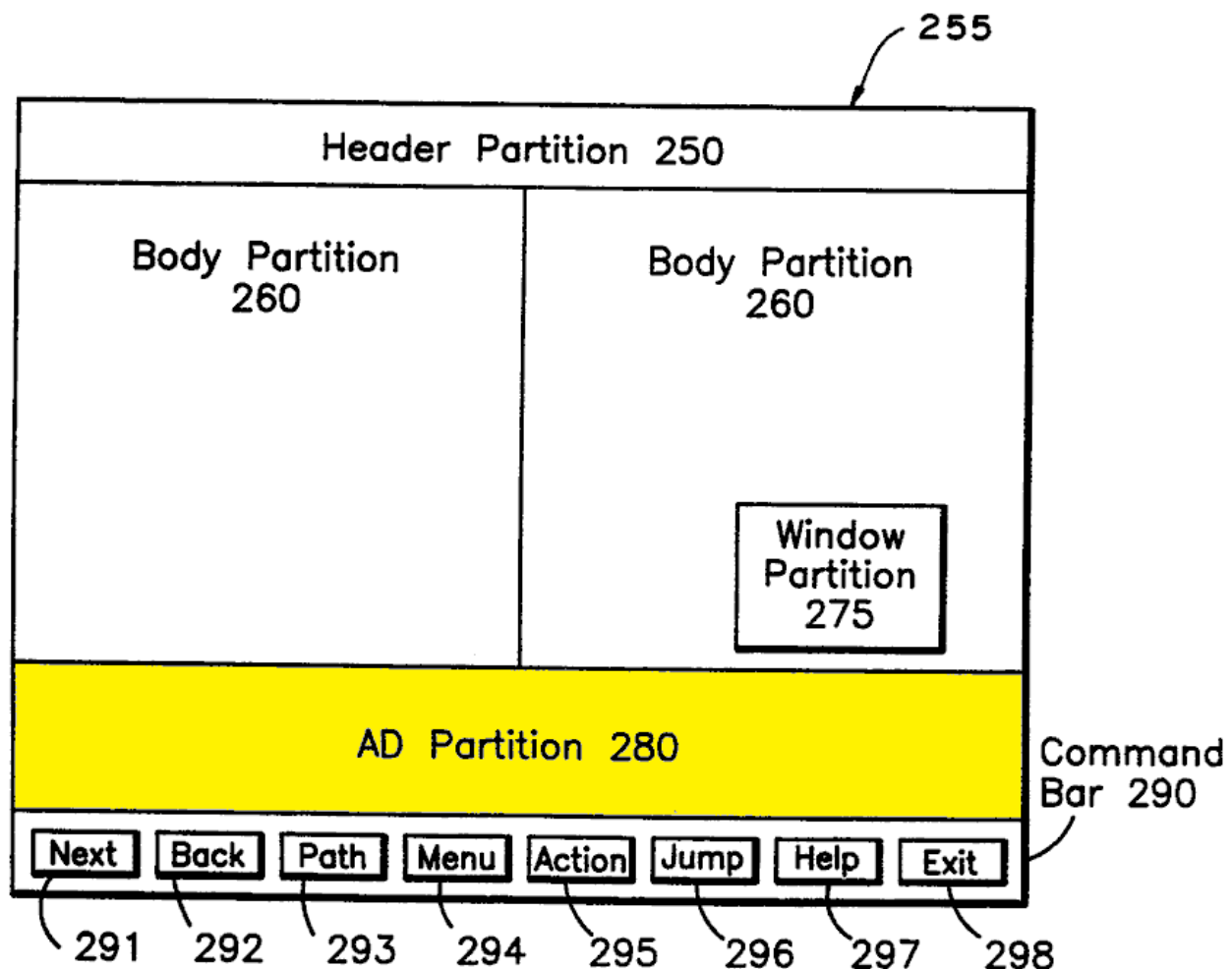


FIG. 3a

(Filepp, Fig. 3a (highlighting added).) As explained above, Filepp discloses that AD partition **280** can display advertisements based on a queue of advertisements assembled based on information about the user. (Filepp, 9:30-47.) It therefore would have been obvious to insert “**advertisement displays between display of said contents corresponding to each of said nodes,**” by displaying advertising

alongside the contents of the nodes (displayed, for example, in body partition **260** shown above). Under the proposed combination, as the system moves from one node to the next, advertisement displays would have been inserted on the screen between the display of the node content as shown in Figure 3a.

9. Claim 36

266. Claim 36 is substantially similar to claim 17, discussed above. This claim is also similar to claim 29 addressed above. The following side-by-side table shows the two claims (with common language shown in underlining): goo

Claim 36	Claim 17
<u>The method in accordance with claim 35</u> further comprising the step of selecting <u>said</u> advertisement displays based on <u>said users' profiles</u> .	<u>The method in accordance with claim 15</u> further comprising the step of providing <u>said</u> dynamic content in accordance with <u>said user's profile</u> .

As shown, other than depending from different claims, claim 36 is substantially similar to claim 17. The key difference is that claim 36 recites “**selecting said advertisement displays**” whereas claim 17 recites “**providing said dynamic content**.” This difference is immaterial because, as explained in claim 17, the dynamic content has been mapped to advertisement displays. I also note that claim 36 recites “**users' profiles**” (plural), which appears to be a typographical error as the antecedent basis for user (in claim 22) merely refers to “**a user**” (singular). Claim

36 is nevertheless obvious because targeted advertising, under the proposed combination, could also have been delivered to multiple users who accessed the system based on their respective user profiles.

G. Ground 6: Claims 37-38 Are Obvious Over Shimizu in view of Benedict, Fox, and Robertson

267. Claim 37 is similar to claim 1 with the exception that it recites a “**multidimensional**” show structure of nodes, and further recites that “**at least two of said nodes are spanned concurrently,**” and that “**said at least two nodes are displayed during an overlapping time period.**” The claimed multidimensional show structure is disclosed by Benedict and Fox, and the additional spanning and display limitations are obvious in further view of Robertson. For all other limitations, as shown below, the analysis from claim 1 applies.

1. Independent Claim 37

268. For reference, claim 37 recites:

37. A method for displaying on a user’s computer, content derived from a plurality of resources in an organized arrangement comprising the steps of:

[a] creating a multidimensional show structure of nodes, each node identifying a resource from a plurality of accessible resources;

[b] without requiring user input accessing each of said resources identified

by each of said nodes wherein at least two of said nodes are spanned concurrently;

- [c] automatically displaying a content corresponding to each of said resources in accordance with said show structure, wherein contents of said at least two nodes are displayed during an overlapping time period.

(’451, 20:65-21:10 (Claim 37; bracketed notation (e.g., [a]) added).) Many of the limitations of claim 27 are substantially similar to limitations in claims 1 that were covered above. In the interests of brevity and consistency, therefore, I will refer back to my previous analysis where appropriate.

269. The preamble of claim 37 recites “[a] **method of displaying on a user’s computer, content derived from a plurality of resources in an organized arrangement....**” It is unclear if the preamble recites a claim limitation, but even if it does, it is disclosed by the prior art. I note that the preamble of claim 37 is identical to the preamble of claim 1. The proposed combination thus renders obvious the preamble for the reasons explained for claim 1.

- (a) **“creating a multidimensional show structure of nodes, each node identifying a resource from a plurality of accessible resources;” (Claim 37[a])**

270. Claim 37[a] is identical to claim 1[a] with the exception that 37[a] requires a **“multidimensional”** show structure of nodes. With respect to the

remaining aspects of claim 37[a], they are rendered obvious by Shimizu, Benedict and Fox as explained for claim 1[a].

271. Turning to the “**multidimensional**” requirement, the ’451 patent does not expressly define multidimensional describes it as generally referring to a show structure in which multiple paths exist through the plurality of nodes. (’451, 2:65-3:6, 6:65-7:12.) I am informed in previous litigation, the district court defined “**multidimensional show structure of nodes**” as a show structure of nodes “*in which the nodes are presented concurrently for at least some portion of the show.*” This requirement largely restates the existing language of claim 37[c], and as shown below, is rendered obvious by the proposed combination. For purposes of my Declaration, it is unnecessary to provide a definitive construction of “**multidimensional show structure of nodes,**” as the prior art discloses the district court’s construction or any other reasonable construction. I will first explain how the proposed combination discloses a show structure of nodes in which multiple paths exist through the plurality of nodes, and then, will explain how the proposed combination discloses that nodes are presented concurrently as specified by the construction adopted by the district court.

272. First, as explained for claim 1[a], the combination of Shimizu, Benedict, and Fox discloses a show structure of nodes because the node and link

information in Shimizu identifies a set of nodes, the content associated with each node, and the links that specify the order in which the nodes are displayed. Under the further combination with Fox, as explained, each resource could have been identified by a URL pointing to the location of the resource on the Internet.

273. The “**multidimensional**” requirement is rendered obvious in view of Benedict and Fox, which make clear that the nodes in the proposed combination need not be linked solely in single sequential fashion – there could be multiple different paths connecting the nodes. As explained in Benedict, each slide can include hyperlinks to jump to other slides in the presentation:

HyperShow goes beyond the special effects of ScreenShow in presenting information. HyperShow keys and buttons enable you to control the flow of your presentation spontaneously, without editing the original slide order. With HyperShow links, you can jump back to a previous slide to reinforce a point, for instance, or you can skip ahead to slides that are more relevant to an idea or question from the audience. The capability of viewing slides in any sequence turns an order-driven presentation into a presentation driven by information. Best of all, you can tailor the information to accommodate the interests and focus of your audience.

(Benedict, pp.435-36.)

274. The capability to view nodes (which in Benedict are represented as

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

slides) “in any sequence” was basic and well-known functionality of hypertext systems such as the Web. It was well-known that each page in a collection of Web pages could have had hyperlinks to other pages in the collection, allowing the user to navigate the set of pages out of order. For example, Fox discloses a NYC Nite Life web site having five pages, each containing a hyperlink to the other four pages:

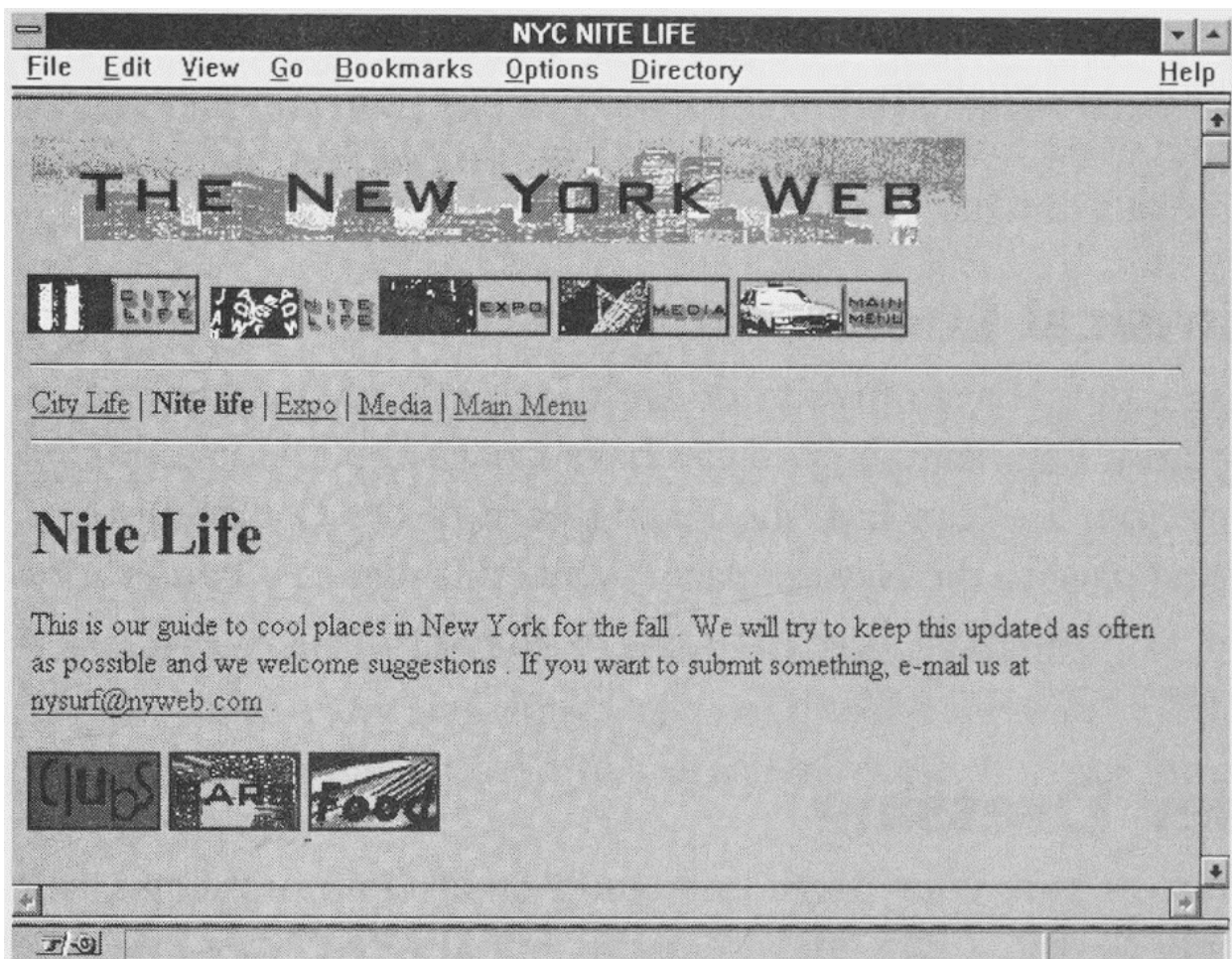


Figure 9-3 The New York Web, where each Web page contains links to other major sections

(Fox, p.239, Fig. 9-3.) Figure 9-3 shows an example in which “[a] set of custom icons at the top of the screen leads to

the City Life, Nite Life, Expo, Media,

or Main Menu section. This same

menu is printed at the top of every

page.” (Fox, p.238.) These links are

also shown under the icons as textual

hyperlinks (as indicated by the

underlining). In the example of Figure 9-3, the user can reach any one of the five

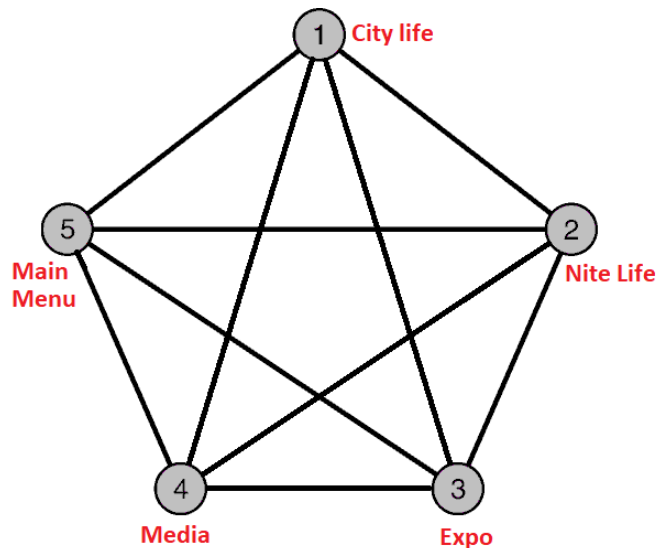
web pages in the set (*i.e.*, City Life, Nite Life, Expo, Media, or Main Menu) from

any other page. In the example above, the “Nite Life” page contains links to the

other four pages, and the same holds true for each of those other pages. Each page

therefore contains links to each other page (as shown in the figure at right that I

created for illustration).



275. It therefore would have been obvious to implement the proposed

combination such that the “**show structure of nodes**” as described for claim 1[a]

was also “**multidimensional**,” in the sense that multiple paths can exist through the

plurality of nodes. This would have entailed nothing more than utilizing known

hyperlinking capabilities as disclosed in Benedict and Fox such that each individual

node in the show structure could have included links to at least two other nodes. During operation of the system based on the proposed combination, therefore, the nodes would have been navigated and displayed either (a) automatically using the default path specified by the node and link information in Shimizu, as explained for claim 1[a], or (b) by utilizing hyperlinks within the nodes to skip to one or more other nodes thus changing the path.

276. I provided a full explanation of the motivation to combine Shimizu, Benedict and Fox for claim 1[a], which applies here. An ordinarily skilled artisan would have been further motivated to implement the proposed combination to provide a “**multidimensional**” show structure in order to enhance flexibility and the user’s experience. The ability to follow multiple different paths to navigate through a set of information was one of the key features that motivated the creation of hypertext systems. Benedict provides an express motivation to provide this functionality by explaining that HyperShow links “enable you to control the flow of your presentation spontaneously, without editing the original slide order.” (Benedict, pp.435-36.) Benedict touts “[t]he capability of viewing slides in any sequence” as a clear benefit because it “turns an order-driven presentation into a presentation driven by information,” and lets the user “tailor the information to accommodate the interests and focus of your audience.” (*Id.*) Fox provides a similar

express motivation by describing consistent hyperlinked navigation tools on each page, such as the example in Figure 9-3 above, as a way to achieve consistency and improve user experience. (Fox, pp.238-239.) An ordinarily skilled artisan would have appreciated that providing for multiple paths through the nodes have provided a useful enhancement to the proposed combination based on Shimizu which, as explained for claim 1, will by default automatically advance from one node next based on the sequential one-node-at-a-time path established by the node and link information. The ability to modify the path through the nodes would have provided useful adaptability, for example, by allowing the system to skip past nodes that the user finds less relevant or interesting, thus allowing a more focused and useful presentation of content for the user.

277. An ordinarily skilled artisan would have had at least a reasonable expectation of success because, as noted, adapting the proposed combination to provide a “**multidimensional**” show structure of nodes would have required nothing more than the existing and built-in linking functionalities available on hypertext systems such as the Web.

278. As noted above, the district court’s construction of “**multidimensional show structure of nodes**” specifies a show structure of nodes “*in which the nodes are presented concurrently for at least some portion of the show.*” To the extent the

district court's construction is applied here, this limitation would have been obvious under two separate and independent theories.

279. First, as explained for 10[a], Figure 8 of Shimizu shows the display **81** of a current node (NODE-A) concurrently with icons **85** for the other nodes (NODE-B, NODE-C, NODE-D). (Shimizu, Fig. 8.) It would have been obvious that icons **85** could have displayed the contents of their underlying nodes. For example, as also explained for 10[a], the Slide Sorter of Benedict shows icons for individual slides, each icon being a scaled down version of its content. (Benedict, pp.43-44, Fig. 2.18.) It would have been obvious to adapt the approach in Benedict to display of icons **85** in Shimizu such that each icon **85** is accompanied with a smaller representation of its corresponding content. The following figure that I created for illustration shows one exemplary way in which a combination of Shimizu and Benedict could have been implemented:

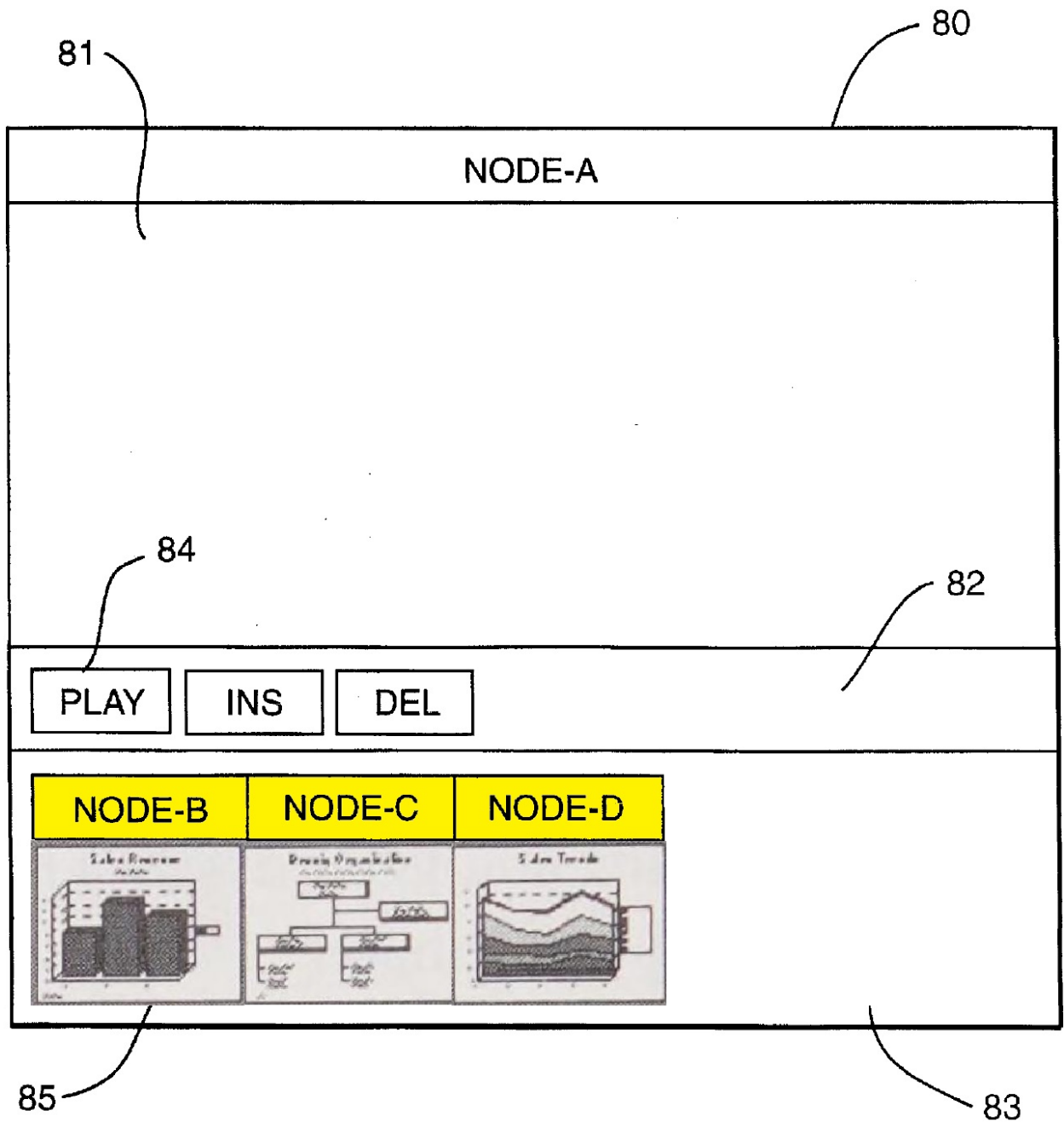


FIG. 8

(Shimizu, Fig. 8 (highlighting added, with scaled down images added).)

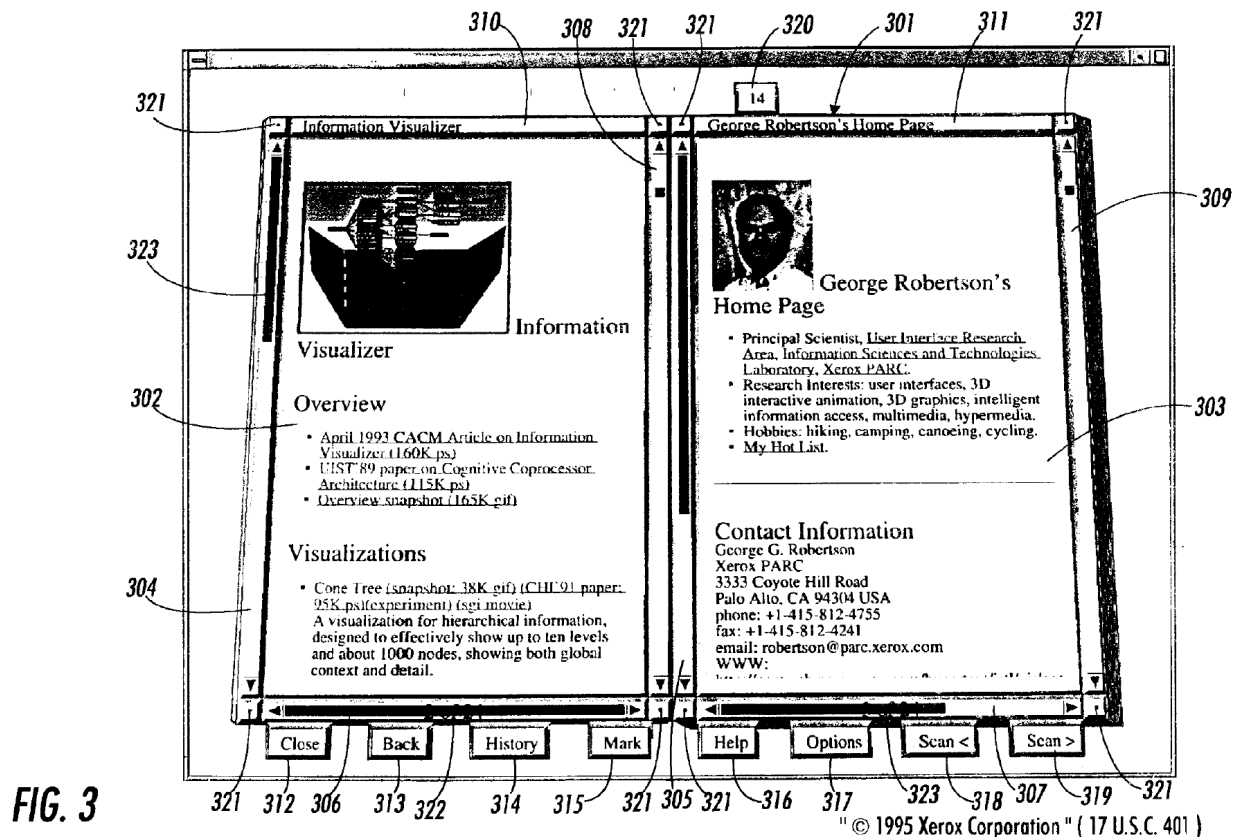
280. An ordinarily skilled artisan would have been motivated to implement

the proposed combination in this manner to more efficiently use screen space by showing contents from multiple nodes simultaneously, making it easier for the user to quickly review and visualize multiple nodes. This is because the smaller versions in Benedict would have benefitted the system of Shimizu by providing the user with a valuable visual cue and preview regarding content of subsequent nodes in the show structure (such as to better inform the user regarding node navigation, ordering and deletion).

281. Second, under a separate theory, the district court's construction would have been obvious in further view of **Robertson**. Robertson discloses a system for displaying lists of linked documents, such as web pages or other types of linked documents in a hypertext system. (Robertson, 3:24-39.) In one embodiment, Robertson presents a list of web pages by displaying each web page in the list as a "page" of a virtual book. (Robertson, Abstract, 2:14-25.) "The list of pages is displayed in a book metaphor, termed WebBook, on a computer controlled display system... When displayed, two active pages are displayed at one time." (Robertson, 2:24-28; *see also id.*, 4:14-19 ("The currently preferred embodiment of the present invention presents the list of pages to a user in a book metaphor hereinafter referred to as a WebBook. The book metaphor is useful since it conveys a relationship between the different pages. Each page in the WebBook represents a page in the

list.”).

282. Figure 3 of Robertson shows an exemplary WebBook view showing two web pages visible in a side-by-side fashion:



(Robertson, Fig. 3.) As explained in Robertson:

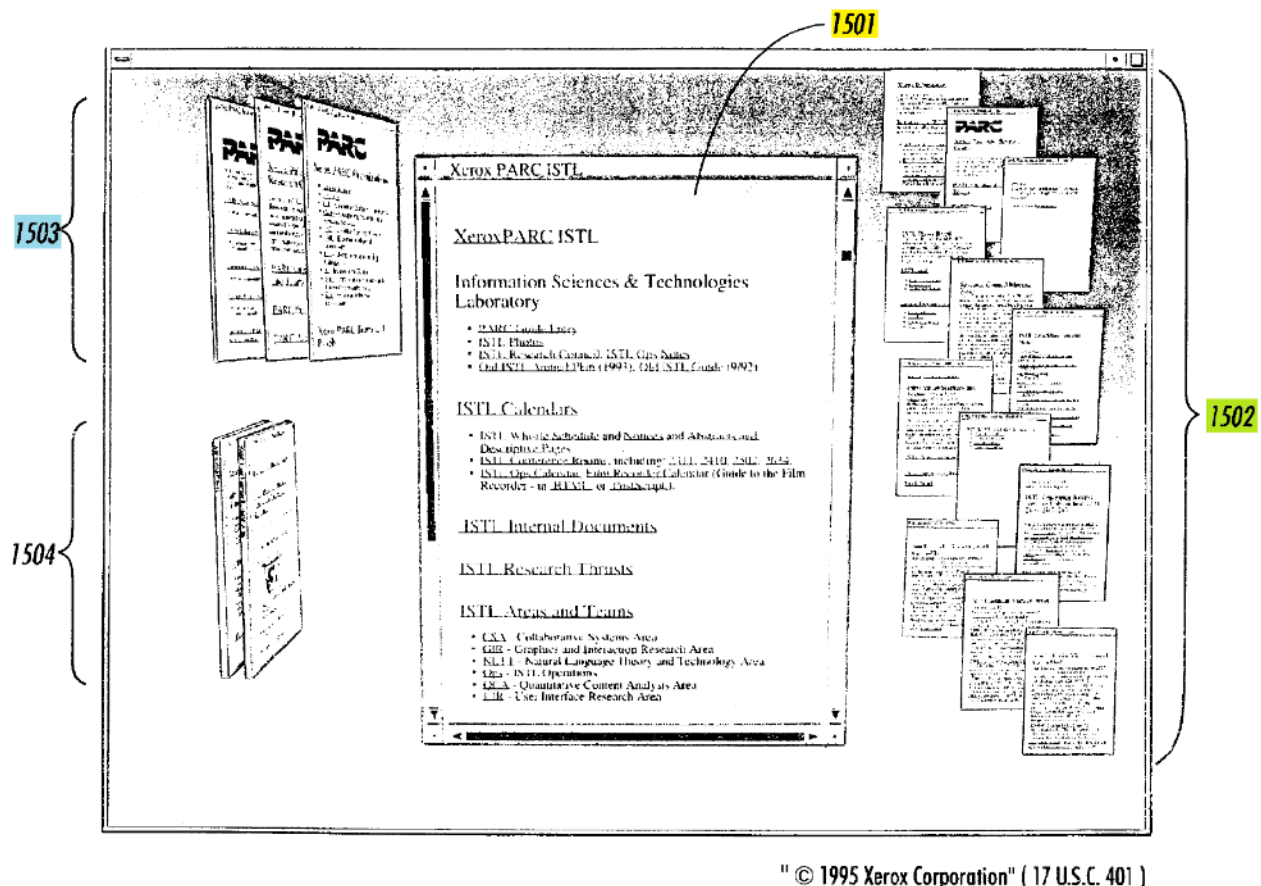
FIG. 3 is an illustration of a WebBook as displayed on a display. Referring to FIG. 3, the WebBook 301 is comprised of a first page 302 and a second page 303, each having an identical structure. The first page 302 is an instance of a left page and the second page 303 is an instance of a right page.

(Robertson, 6:15-19; *see also id.*, 6:28-30 (“The content and layout of the page is

determined by the HTML macros defining the page.”.) Robertson explains that, in order to display web pages in a WebBook, each Web page is converted into a page object. (Robertson, 7:65-66.) “Each page is provided in the order in which it is to appear in the WebBook. Accordingly, each page object will represent either a left page or a right page in the WebBook.” (Robertson, 7:66-8:2.) The “Scan >” button **319** in Figure 3 causes the pages in the WebBook to advance forward automatically after a period of time. (Robertson, 6:45-46, 9:5-6.) Robertson thus discloses an embodiment in which content corresponding to at least two nodes is automatically accessed and displayed at the same time.

283. But Robertson is not limited to simultaneously accessing and displaying just two pages. Another embodiment depicted in Figure 15 shows an enhancement in for viewing images representing many or even all of the pages of a WebBook:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1



(Robertson, Fig. 15 (highlighting added).)

284. Figure 15 shows the WebBook as providing “a feature of a complete Web Browsing system.” (Robertson, 10:15-16.) Area 1501 in the center shows the current web page viewed by the user. (Robertson, 10:17-19.) Area 1503 to the left “contain[s] pages which have been viewed by the user.” (Robertson, 10:22-24.) Area 1502 on the right contains “a collection of ‘thumbnail’ images of pages,” which have “not yet been viewed by the user.” (Robertson, 10:19-22.) “As the user views pages and skips to different pages, the prior current page is moved to the area 1503.”

(Robertson, 10:24-26.) “It should be noted that the collection of pages, i.e. the current page 1501 and the pages in the areas 1502 and 1503 themselves may be pages from a WebBook.” (Robertson, 10:26-28.)⁷

285. The screen display in Figure 15 shows an example of displaying 16 WebBook pages simultaneously (thumbnails of three previously-viewed pages in area 1503, one current page in area 1501, and thumbnails of 12 not-yet-viewed pages in area 1502). It would have been obvious to a person of ordinary skill in the art that, using the techniques of Robertson and depending on the number of pages in a WebBook, all of the pages of the book could have been displayed to the user concurrently or during an overlapping time period.

286. ***Rationale and Motivation to Combine (Shimizu, Benedict, and Fox with Robertson):*** I provided a full explanation of the motivation to combine Shimizu, Benedict, and Fox above. With respect to this aspect of claim 37, it would have been obvious to combine those references with Robertson. The combination would have predictably yielded a show structure of nodes “*in which the nodes are presented concurrently for at least some portion of the show,*” as required by the

⁷ Area 1504 (not highlighted in Figure 15) is a bookshelf area where WebBooks are maintained and may be accessed by the user. (Robertson, 10:29-32.)

district court's claim construction, because contents of multiple nodes would have been displayed at the same time based on the teachings of Robertson. I note that the claim language itself (claim 37[c], discussed below) only requires concurrent display of "**at least two of said nodes**," but it would have been obvious under the proposed combination to simultaneously display contents of many or all nodes in the claimed show structure as depicted in Figure 15 above. This requirement is satisfied by display of the current web page and thumbnails of the previously-reviewed and not-yet-seen nodes under the proposed combination because the thumbnail representations as shown in Figure 15 show miniaturized versions of the nodes.

287. Robertson is an analogous reference in the same field of techniques for providing organized or sequenced presentation of content, and would have been reasonably pertinent to a number of problems facing the inventor. For example, Robertson would have been reasonably pertinent to the problem of how to access and display multiple nodes concurrently, and would have provided useful teachings about how to effectively present linked web pages to the user.

288. An ordinarily skilled artisan would have been motivated to combine in the manner described above in order to improve user experience. As noted previously, Shimizu itself already discloses display of a node and icons **85** for each other linked node in the set of nodes. (Shimizu, Fig. 8, 7:30-35.) An ordinarily

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

skilled artisan would have been motivated to further enhance the proposed combination based on Robertson, as it explains that organizing information using a “book metaphor” was a known way to present information to the user. (Robertson, 1:66-2:3 (“The present invention uses a book metaphor to enable a user to browse through a list of web pages. The use of a book metaphor on computer based systems is known in the art. Such a book metaphor has been used as an aid for organizing related files and/or applications.”).) An ordinarily skilled artisan would have appreciated that the “book metaphor” of Robertson has a clear advantage of leveraging the user’s familiarity with real-world books to present an intuitive user interface for viewing linked webpages. As an additional advantage and benefit, the techniques in Robertson can display two or more web pages at the same time, thus more efficiently utilizing screen space and improving the user’s ability to review and visualize the actual content associated with the nodes.

289. An ordinarily skilled artisan would have had at least a reasonable expectation of success. Robertson itself confirms that web-based techniques, web browsers, and the ability to browse information using a “book metaphor,” were already known to persons of ordinary skill in the art and commercially available. (Robertson, 1:10-2:11.) The techniques in Robertson that I cite for claim 37 could have been implemented using conventional programming techniques, and the

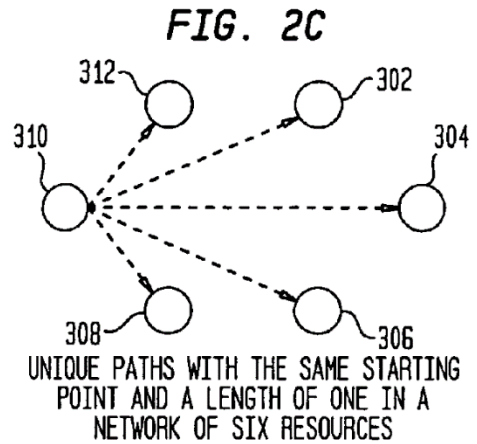
disclosures of Robertson would have provided more than sufficient detail to have enabled an ordinarily skilled artisan to implement the cited teachings applied to the proposed combination, without undue experimentation.

(b) **“without requiring user input accessing each of said resources identified by each of said nodes wherein at least two of said nodes are spanned concurrently;”
(Claim 37[b])**

290. Claim 37[b] is substantially identical to claim 1[b], except that claim 37[b] adds **“wherein at least two of said nodes are spanned concurrently.”** With respect to the other portions of claim 37[b], as explained for claim 1[b], under the proposed combination, the resources in each of the nodes would have been automatically accessed and presented on the user’s computer for the specified duration before advancing to the next node.

291. With respect to the requirement that **“at least two of said nodes are spanned concurrently,”** this is simply a byproduct of the **“multidimensional”** feature of the show structure under the proposed combination, described above.

292. The '451 patent does not expressly define what it means for nodes to be “**spanned concurrently**,” but provides a simple illustration with reference to Figure 2C (at right): “Fig. 2c illustrates a show structure that spans multiple paths from the same starting node such as node 310 to each one of the remaining nodes 312, 302, 304, 306, and 308.” ('451, 7:13-16.) “Thus, a show in accordance with the structure presented in FIG. 2c simultaneously presents the contents identified by nodes 312, 302, 304, 306 and 308, after the content identified by node 310 is presented.” ('451, 7:16-19.)



293. Figure 2C provides a highly simplified example in which nodes 302, 304, 306, and 308 are “**spanned concurrently**” because each of these nodes forms a separate simultaneous path from node 310. I am informed that in a previous litigation involving the '451 patent, the district court construed “**at least two of said nodes are spanned concurrently**” as “*nodes are accessed such that the content corresponding to at least two nodes is accessed at the same time*,” which is broader than the specific example discussed above. The district court’s construction is satisfied for the reasons explained for claim 37[a], because under the proposed combination, the content corresponding to at least two of said nodes are accessed

(and displayed) at the same time.

294. The claimed spanning of at least two nodes would have been obvious under the proposed combination because, as explained for claim 37[a], the show structure would have been capable of having multiple paths, with each node having links to two or more other nodes. Using the example from Figure 9-3 from Fox above, “Nite life” has a path to four other nodes (*i.e.* “City Life,” “Expo,” “Media,” “Main Menu”). A show structure based on these nodes and links discloses at least four nodes that are **“spanned concurrently.”** In fact, a show structure of nodes could satisfy the requirement of **“at least two of said nodes [being] spanned concurrently”** if it had a single node with links to two other nodes. With respect to the district court’s previous construction requiring that *“nodes are accessed such that the content corresponding to at least two nodes is accessed at the same time,”* this is also addressed in my discussion of claim 37[a] above as noted. Under the proposed combination with Robertson, in fact, it would have been obvious that the content corresponding to all of the nodes in the show structure would have been accessed at the same time.

- (c) **“automatically displaying a content corresponding to each of said resources in accordance with said show structure, wherein contents of said at least two nodes are displayed during an overlapping time period.”**
(Claim 37[c])

295. Claim 37[c] is substantially similar to claim 1[c], except that claim 37[c] adds **“wherein contents of said at least two nodes are displayed during an overlapping time period.”** With respect to the other portions of claim 37[c], as explained for claim 1[c], content corresponding to the resources associated with each node (*e.g.*, character string, image, sounds, motion pictures, etc.) is automatically displayed in accordance with the show structure for the specified duration.

296. With respect to the requirement that **“contents of said at least two nodes are displayed during an overlapping time period,”** this would have been obvious for the reasons discussed above. Under the proposed combination, therefore, **“contents of said at least two nodes are displayed during an overlapping time period,”** as claimed, because contents of at least two concurrently-spanned nodes would have been displayed at the same time. I note that the claim only requires display of **“contents of said at least two nodes,”** but it would have been obvious under the combination with Robertson to simultaneously display **“contents of”** every node in the claimed show structure as depicted in Figure 15 above. Claim 37 is a **“comprising”** claim that merely requires that contents **“of said**

at least two nodes” be displayed during an overlapping time period, and thus, does not exclude displaying additional nodes during that same time period. Moreover, claim 37[c] only requires display of “**contents of**” the at least two nodes, and does not require that the nodes be displayed on the screen in their entirety or in any particular size or arrangement. And under the further combination with Robertson, as explained, the contents of the at least two nodes would have been fully displayed as shown in Figure 3 of Robertson.

2. Claim 38

297. Claim 38 is substantially similar to claim 2, discussed above. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 38	Claim 2
The method in accordance with claim 37 further <u>comprising the step of providing a duration information, representing the duration within which a corresponding content to each of said resources is being displayed.</u>	The method in accordance with claim 1 , wherein said step of creating <u>further comprises the step of providing a duration information, representing the duration within which a corresponding content to said resource is being displayed.</u>

298. As shown, other than depending from a different independent claim, claim 38 substantially similar to claim 2. Claim 38 is arguably broader than claim 2 because it does not define “**the step of providing a duration information**” by

reference to the creating step, and this difference is immaterial for our purposes here.

Claim 38 is thus obvious for the same reasons as claim 2.

H. Ground 7: Claims 39-44 Are Obvious Over Shimizu in view of Benedict, Fox, and Miller

299. Independent claim 39 is substantially similar to claim 22 except that it adds an additional step of “**selecting a category from a predefined list of categories,**” and further requires an on-line search that is “**related to said category.**” The category-based limitations added by claim 39 are obvious based on the further disclosures in Miller. With respect to the other limitations, as shown below, the analysis from claim 22 applies here.

1. Independent Claim 39

300. For reference, claim 39 recites:

39. A method for displaying in a sequential arrangement to a user’s computer, content derived via a communications network from a plurality of resources the method comprising the steps of:

- [a] selecting a category from a predefined list of categories;
- [b] performing an on-line search related to said selected category, in response to which a plurality of search results and their corresponding URL resources are received;
- [c] automatically creating and storing a show structure of nodes, each node

identifying a resource from a plurality of accessible resources via said communications network wherein said show structure is created in response to said search results received in response to said on-line search;

[d] without requiring user input retrieving each of said resources identified by each of said nodes; and

[e] automatically delivering said content to said user for display on said user's computer in accordance with said show structure.

('451, 21:15-35 (Claim 39; bracketed notation (e.g., [a]) added).) Many of the limitations of claim 39 are substantially similar to limitations recited in claims 1 and 10 covered above. In the interests of brevity and consistency, therefore, I will refer back to my previous analysis where appropriate.

301. The preamble of claim 39 recites "[a] method for displaying in a sequential arrangement to a user's computer, content derived via a communications network from a plurality of resources...." It is unclear if the preamble recites a claim limitation, but even if it does, it is disclosed by the prior art.

302. I note that the preamble of claim 39 is substantially identical to the preamble of claim 10, except that claim 39 recites displaying content "**to a user's computer**" whereas claim 10 recites displaying content "**on a user's computer,**"

which presents no material difference. The proposed combination thus renders obvious the preamble.

(a) “selecting a category from a predefined list of categories;” (Claim 39[a])

303. As explained for claim 22[a], Fox describes various on-line search facilities available by the mid-1990s, including Yahoo – “the mother of all Web indexes.” (Fox, p.178.) Fox explains that “Yahoo has hundreds of links to pages in all of the following categories,” and then lists 19 categories including Art, Business, Computers, Society and Culture, among many others. (Fox, p.178.) Figure 8-2 of Fox shows a screen capture from Yahoo showing a list of subcategories within the “Society and Culture” category, each category containing documents:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

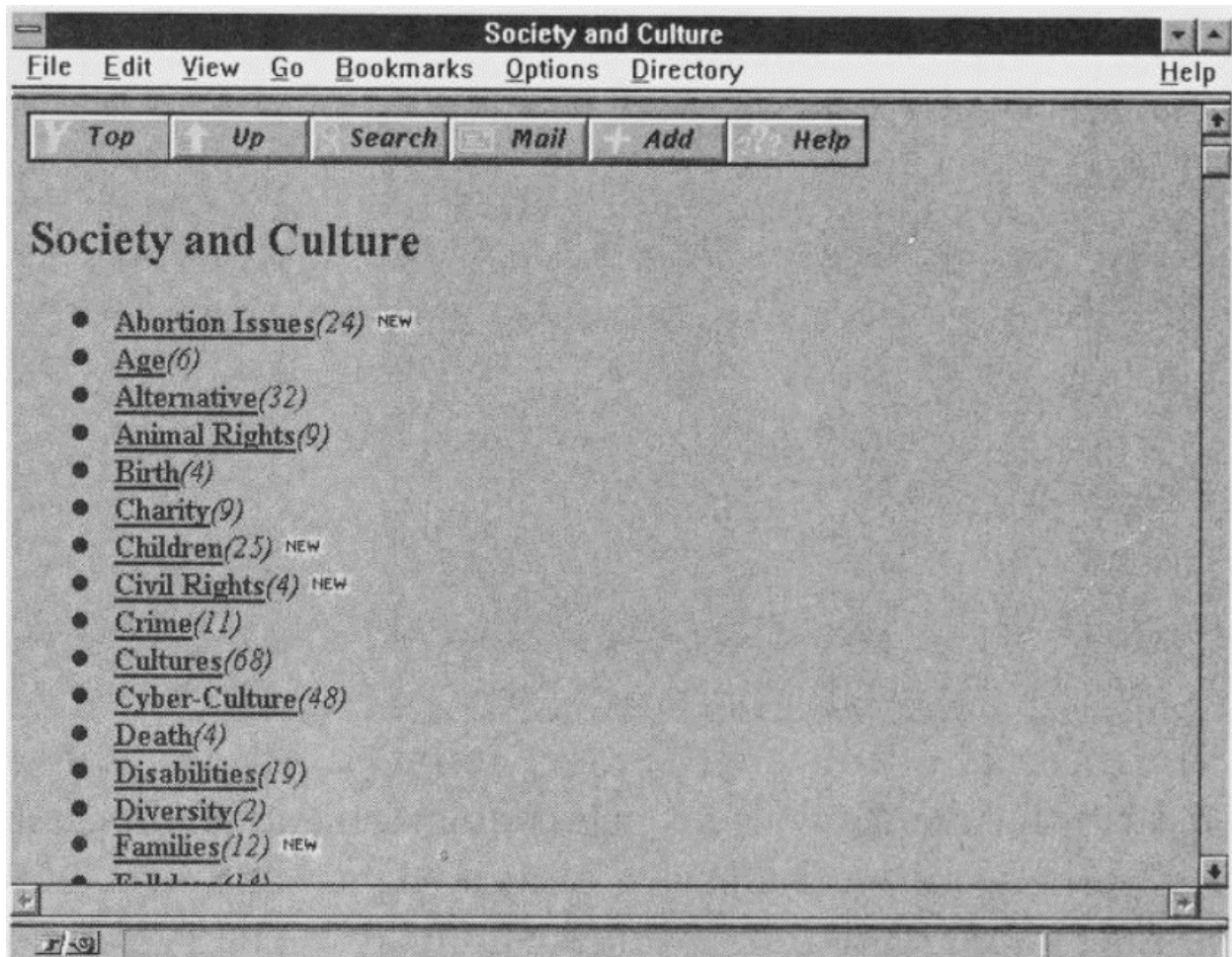


Figure 8-2 Yahoo

(Fox, p.179, Fig. 8-2.) But Fox does not disclose further details about Yahoo, such as the mechanism by which categories are selected for purposes of searching. I have therefore also cited to **Miller** for this limitation. As shown below, Miller discloses a technique for selecting categories (referred to in Miller as “sections”) within a predefined list of categories, which is then used to conduct a search.

304. As noted in **Part V.A.2** above, Miller describes various features of a

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

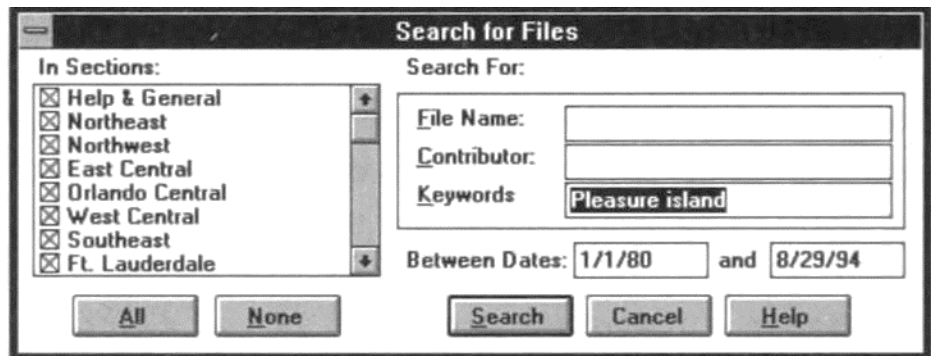
commercial on-line service known as CompuServe which was popular in the mid-1990s. (Miller, p.1.) CompuServe provided a feature known as “forums,” which were on-line communities where members could exchange information and messages about particular topics of interest. (Miller, p.29 (“You can find forums on just about any topic, from scuba diving to C programming.”); *id.*, p.30 (“A forum is like one of those after-school clubs you belonged to back in high school. A bunch of people with the same interests get together to talk and exchange information.”).) Forums included a message section, “where forum members exchange messages with each other,” and a file library, “where forum files are stored.” (Miller, p.34.)

305. Miller discloses that, “[i]n addition, every forum’s message sections and file libraries are comprised of multiple *sections*. Each section represents a more focused area of interest within the forum topic.” (Miller, p.35 (*italics in original*).) “For example, the Florida Forum has sections for Ft. Lauderdale, Tampa, St. Petersburg, and other specific locations.” (*Id.*) “The MacMillan Computer Publishing Forum (where you talk to Que [the publisher of Miller] has sections for Word Processing, Databases, Programming, and several other specific topics.” (*Id.*)

306. Miller also discloses features of CompuServe that allow users to search forums and sections within the forums. Figure 3.9 shows a user interface dialog box that allows a user to perform such a search:

Fig. 3.9

Use the Search for Files dialog box to search for specific forum files.



(Miller, p.42, Fig. 3.9; *see also id.*, p.42 (“A better way to find files... is to click the Search Libraries button in the toolbox. When you do this, a Search for Files dialog box appears (see fig. 3.9).”).) The dialog box, as shown, includes an “**In Sections:**” area that allows the user to select one or more sections to search. (Miller, p.42 (“Select the sections you wish to search.”).) “When you click the Search button, the forum’s libraries are searched for any and all files that match your search criteria.” (Miller, p.43; *see also id.*, p.419 (showing similar example).)

307. Miller thus discloses “**selecting a category from a predefined list of categories,**” *i.e.*, selecting at least one section (“**category**”) from a predefined list of sections as shown in Figure 3.9 above. A section in Miller qualifies as a “**category**” because it represents a class of items that have particular characteristics, *e.g.*, that relate to a particular topic of interest.

308. ***Rationale and Motivation to Combine:*** I already provided a full explanation of the rationale and motivation to combine Shimizu, Benedict, Fox, and

Guinan in my discussion of claim 22 above. It would have been obvious to adapt in the system based on the proposed combination to provide the further ability to select a category from a predefined list of categories, as disclosed in Miller.

309. Miller is an analogous reference in the same field of techniques for organizing and presenting content, and would have been reasonably pertinent to a number of problems facing the inventor. For example, Miller would have been reasonably pertinent to the problem of how to implement a searching feature that allowed searches within selected categories.

310. An ordinarily skilled artisan would have been motivated to combine in the manner described above in order to improve the user experience. The ability to specify particular categories for a search would have provided the clear benefit of allowing the user to focus a search on particular topics of interest, thus resulting in search results that are more relevant to the user. This would have been particularly useful in cases where the search could result in a potentially large number of results. Using the example of Figure 3.9, suppose the user was only interested in files in the Florida Forum specifically pertaining to Ft. Lauderdale. The user could select just the Ft. Lauderdale section (and deselect other sections) to search only for files within that category, thus avoiding the need to review numerous search results from other sections not relevant to the user.

311. An ordinarily skilled artisan would have had at least a reasonable expectation of success. The ability to organize information into categories, and to search within selected categories, would have been considered rudimentary and elementary tasks to ordinarily skilled artisans. The disclosures of Miller would have provided more than sufficient details to have enabled an ordinarily skilled artisan to adapt the features that I have cited above to the proposed combination, without undue experimentation.

(b) **“performing an on-line search related to said selected category, in response to which a plurality of search results and their corresponding URL resources are received;” (Claim 39[b])**

312. Claim 39[b] is substantially identical to claim 22[a], except that claim 39[b] requires an on-line search **“related to said selected category.”** All other aspects of claim 39[b] are obvious based on the proposed combination of Shimizu, Benedict, Fox, and Guinan as explained for claim 22[a].

313. With respect to requiring that the on-line search **“relate to said selected category,”** this would have been obvious in further view of Miller discussed above, which specifically discloses limiting the search to the selected categories. Under the proposed combination, therefore, the on-line search performed under the proposed combination as disclosed in Guinan and Fox (as explained for claim 22[a]) would

have been further enhanced to search for hypertext nodes within the selected category. As explained previously, this would have provided a distinct advantage of focusing the search on items more relevant to the user.

- (c) **“automatically creating and storing a show structure of nodes, each node identifying a resource from a plurality of accessible resources via said communications network wherein said show structure is created in response to said search results received in response to said on-line search;” (Claim 39[c])**

314. Claim 39[b] is identical to claim 22[b], and is thus obvious for the reasons stated previously.

- (d) **“without requiring user input retrieving each of said resources identified by each of said nodes; and” (Claim 39[d])**
- (e) **“automatically delivering said content to said user for display on said user’s computer in accordance with said show structure.” (Claim 39[e])**

315. These limitations are substantially similar to limitations already covered for claim 1, 10, and 22, discussed in detail above, and the analysis above applies equally here. As explained, under the proposed combination, the resources and content identified by each of the nodes would have been automatically accessed, further filter search results retrieved, and delivered to the user’s computer and displayed for the specified duration. The content would also have been delivered

“in accordance with said show structure,” because the resources are identified by the node information and presented to the user’s computer in the specified sequence.

2. Claim 40

316. Claim 40 depends from claim 39, and further recites **“the method in accordance with claim 39, wherein prior to said step of automatically delivering the method further comprises the step of filtering the retrieved plurality of said resources to determine which of the retrieved resources are to be displayed.”**

This limitation is obvious based on Guinan.

317. Guinan, as explained above, assigns a weight to each matching hypertext node and then filters out nodes that fall below a minimum cut-off or threshold value. (Guinan, p.125 (“All nodes whose weight is greater than or equal to this cut-off or threshold value would be selected for inclusion in the tour.”).) This clearly discloses the step of, prior to the automatically delivering step, **“filtering the retrieved plurality of said resources.”** This is because Guinan restricts (filters) the resources that will be included in the tour presented to the user.

318. I note that my mapping of claim 39[b], which recites **“performing an on-line search,”** does not render claim 40 superfluous. Guinan performs a search by identifying hypertext nodes based on the user’s query. It is true that Guinan performs a further step of filtering out search results that fall below the threshold

value, but that is a subsequent step performed after search results have been obtained. (Guinan, p.125 (“Once each of the nodes in the hypertext have been assigned a utility score based on the user’s query, it is necessary to establish a cut-off point (*u*) below which all nodes whose utility score is less than are discounted from inclusion in the tour.”).) This additional step of removing search results that fall below the threshold or cut-off value is not required by claim 39[b]. Accordingly, the additional filtering in Guinan is, for purposes of claim 39[b], an unclaimed additional functionality. And that additional functionality renders obvious claim 40 as discussed.

319. It therefore would have been obvious, under the proposed combination, to filter the retrieved plurality of resources to determine which ones will be delivered to the user, which would obviously have occurred “prior to said step of automatically delivering,” as claimed.

3. Claim 41

320. Claim 41 depends from claim 39, and further recites “**the method in accordance with claim 39, further comprises the step of automatically replacing each of said displayed content to said user with a following retrieved resource, after a specified period of time.**” This limitation was already covered in the discussion of claims 1 and 2, above. As explained, under the proposed combination, displayed content is automatically replaced “with a following retrieved resource”

because, when the system advances to next node in the show structure of nodes after a particular duration, at the displayed content is replaced with content associated with the next node. (Shimizu, 8:67-9:4 (“In step 105, the next node information is reproduced in accordance with the next link information. The processing above is carried out so that the node C is displayed followed by node B and node D, in accordance with the order of the link information.”), 5:11-14 (“This makes it possible to build with ease an application program illustratively causing the display of the nodes to be switched in a specific order and at predetermined intervals.”).)

4. Claim 42

321. Claim 42 depends from claim 39, and further recites “**the method in accordance with claim 39, further comprising the step of allowing an option to bookmark each of the resources being displayed for later viewing.**” Claim 42 covers nothing of patentable significance as it merely recites well-known bookmarking features of web browsers. Fox confirms that conventional web browsers allowed users to bookmark web pages displayed for later viewing. (Fox, p.138 (“In your journeys around the Web, you’re bound to come across a page or two that you absolutely adore. Whenever you find a page you’re likely to use again, Netscape makes it easy to place a bookmark. Simply select Bookmarks, Add Bookmark, or press CTRL-A.”), pp.64-65 (discussing similar bookmark option for

Lynx web browser), pp.96-97 (SkipKnot web browser).)

322. Under the proposed combination based on Fox, as explained above, it would have been obvious that the displayed “**resources**” could have been web pages identified by URLs. It therefore would have been obvious, based on the built-in features of conventional web browsers, to “**allow an option to bookmark each of the resources being displayed for later viewing.**” Under the proposed combination, therefore, the option would have been available to the user when the resources associated with the nodes are being displayed by the web browser. An ordinarily skilled artisan would have been motivated to implement this feature in order to gain the benefits of bookmarking as described in Fox, *e.g.*, the ability to keep track of a selected web page for later viewing.

5. Claim 43

323. Claim 43 depends from claim 39, and further recites “**the method in accordance with claim 39, further comprising the step of allowing an option to bookmark each of the resources being displayed for printing a selected subset of said retrieved resources.**” As explained for claim 42, it would have been obvious to allow an option to bookmark each of the resources being displayed. Conventional web browsers also allowed users to print out web pages being displayed. (Fox, *e.g.*, p.58, Table 3-1 (“Print the page to disk, e-mail, or a printer.”),

p.108 (“Print: Prints out the current Web page.”), p.117 (“If you have a printer attached to your PC, you can easily create a hard copy of any Web page. Simply select File, Print.”), p.145 (“If you have a printer attached to your computer, you can easily generate a hard copy of the current Web page. In most cases, all text and graphics go to the printer, so your Web page looks the same on paper as it did on the screen.”).) It thus would have been obvious that a user could have bookmarked a particular page being displayed (as discussed for claim 42), in order to print it out. This would have predictably resulted in “printing a selected subset of said retrieved resources,” by printing a particular web page which is a subset of the plurality of web pages identified in the show structure of nodes. An ordinarily skilled artisan would have been motivated to implement this feature in order to gain the benefits of obtaining a hardcopy version of a desired web page.

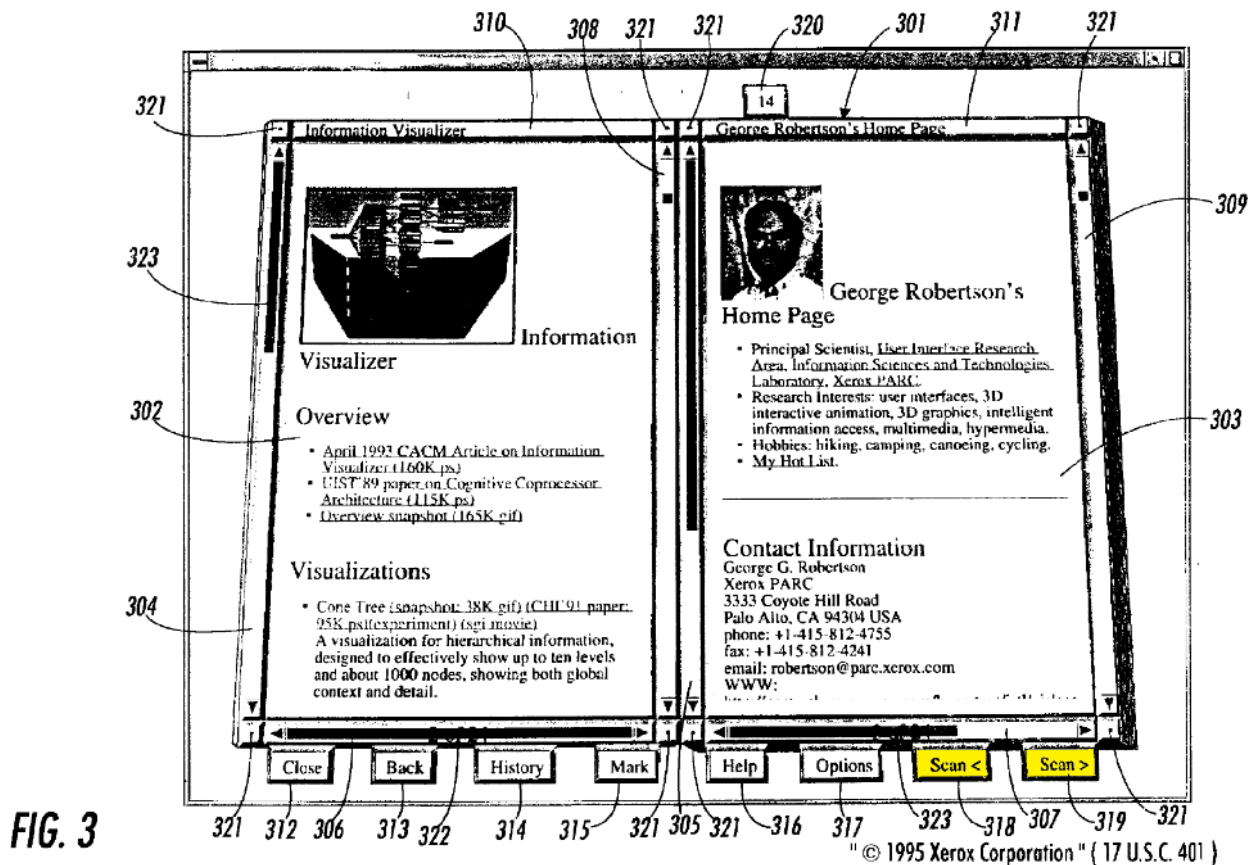
6. Claim 44

324. Claim 44 depends from claim 39, and further recites “**the method in accordance with claim 39, wherein each of said retrieved resources is a World Wide Web page.**” Under the proposed explained and based on Fox, as explained above, it would have been obvious that each of the retrieved resources is a webpage identified a URL.

I. Ground 8: Claim 9 is Obvious Over Shimizu in view of Benedict, Fox, Robertson, and Sagman

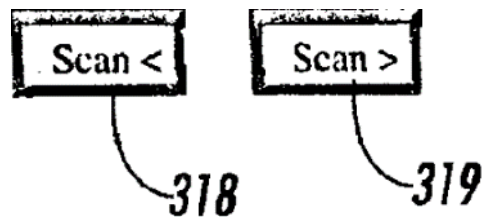
325. Claim 9 depends from claim 8 and further recites “**the step of receiving control button commands for pausing said display of said data files, and for reversing the sequence of said display of said data files and for fast forwarding said display of said data files.**” As explained for claim 8, the proposed combination of Shimizu, Benedict and Fox render obvious control buttons for controlling the display of data files. But these references do not appear to disclose the more specific control button commands in claim 9. I have therefore cited these references in further combination with Robertson and Sagman, which were also discussed above.

326. As explained, Robertson discloses a system for presenting a list of web pages as a virtual book referred to as a WebBook. Figure 3 of shows an exemplary WebBook showing two web pages visible in a side-by-side fashion:



(Robertson, Fig. 3 (highlighting added).)

327. The key features of Robertson for purposes of claim 9 relate to the buttons on the bottom of Figure 3 that allow the user to control the display of web pages. (Robertson, 6:30-57.) More specifically, the two buttons on the far right side shown in highlighting, “Scan <” 318 and “Scan >” 319 (shown at right), allow the user to automatically advance backward or forward, respectively, through the web pages in a WebBook. “The button 318 labeled



Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

‘Scan <’ provides for automatic scanning of pages to the lower numbered pages. The button **319** labeled ‘Scan >’ provides for automatic scanning of pages to the higher numbered pages.” (Robertson, 6:43-46.) “The scan operation will cause the pages to turn at a predetermined rate.” (Robertson, 9:5-6.)

328. Robertson also makes clear that the rate at which the pages will advance using the scan operation can be adjusted by the user. Figure 13 discloses a pop-up window **1300** user interface for adjusting this rate. (Robertson, 9:45-46.)

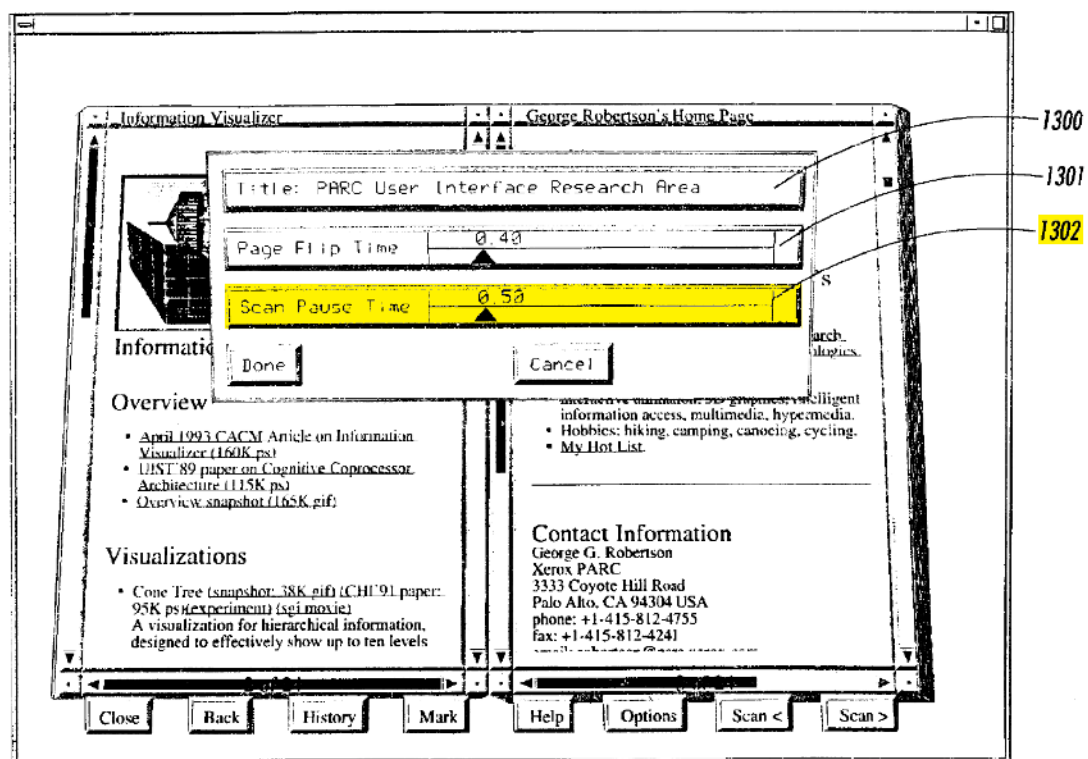


FIG. 13

" © 1995 Xerox Corporation" (17 U.S.C. 401)

(Robertson, Fig. 13 (highlighting added).) User interface bar **1302** in Figure 13 shown in highlighting allows the user to vary the scan pause time, which in this case

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

is currently set to 0.50. (Robertson, 9:49-50, 9:57-58.) “The scan pause time refers to a pause in between page turns during a scan operation.” (Robertson, 9:50-51.) The user can decrease or increase the scan pause time by moving the indicator to the left or right using a dragging operation. (Robertson, 9:53-58.)

329. Under the further combination with Robertson, therefore, it would have been obvious to receive control button commands for “**reversing the sequence of said display of said data files and for fast forwarding said display of said data files,**” as claimed. As I explained previously, Robertson is an analogous reference.

330. With respect to **reversing** the sequence, the “**Scan <**” **318** control button, as noted, causes the WebBook to automatically present web pages in reverse order. (Robertson, 6:43-45 (“The button **318** labeled ‘Scan <’ provides for automatic scanning of pages to the lower numbered pages.”).) It therefore would have been obvious to adapt the system based on the proposed combination to receive a control button command that reverses the sequence of automatic display of data files associated with the nodes in the claimed show structure of nodes. An ordinarily skilled artisan would have been motivated to provide this feature to allow a user to reverse the order of presentation in the event the user wishes to revisit previously viewed content.

331. With respect to **fast forwarding**, the “**Scan >**” **319** control button, as

noted, causes the WebBook to automatically present the web pages in forward order. (Robertson, 6:45-46.) Benedict and Fox disclose similar functionality of automatically advancing to the next slide or web page after a particular duration selected by the user. But Robertson provides the *further* ability to shorten that duration using “scan pause time” control **1302** shown in Figure 13 above.

332. It therefore would have been obvious under the proposed combination to adapt the system based on the proposed combination to receive a control button command that reduces the amount of time a node is displayed before automatically advancing to the next node. For example, under the proposed combination, a control button command could have reduced the duration between nodes from 10 seconds to 1 second, or other desired value, predictably resulting in “**fast forwarding said display of said data files.**” An ordinarily skilled artisan would have been motivated to provide this feature to allow a user to speed up the advancement to subsequent nodes, for example, if the user already has familiarity with the node content.

333. For purposes of the fast forward limitation, the “**button**” corresponds in Robertson to the user interface bar **1302** in Figure 13 that allows the user to vary the scan time. This qualifies as a control “**button**” because the user has to click on the bar to change the setting using a dragging operation. (Robertson, 5:21-24, 9:53-57.) Alternatively, the control “**button**” could also correspond to the “Done” button

in Figure 13 because it accepts and applies the newly-specified scan time. In any case, it would have been obvious to adapt the proposed combination to include a control button that, when activated, causes the computer to receive a command that causes the fast forwarding to occur. User interface control buttons were common features of conventional graphical user interfaces, and the particular appearance and appearance of the control buttons under the proposed combination would have been a trivial and routine implementation detail for an ordinarily skilled artisan.

334. With respect to “**pausing said display of said data files,**” this would have been obvious in further view of Sagman. Sagman, as noted, discloses features of Microsoft PowerPoint which provides functionality similar to Harvard Graphics as described in Benedict. Similar to Benedict and Robertson, Sagman discloses the ability to automatically advance and display the next item after a user-specified duration. (Sagman, p.391 & Fig. 16-2.) Sagman further discloses a keyboard shortcut that performs the action of: “Pause/resume automatic show” (Sagman, p.395, Table 16-1). It therefore would have been obvious to adapt the proposed combination to provide a control button that causes receipt of a command that temporarily stops the system from automatically advancing to the next node after the specified duration, predictably resulting in “**pausing said display of said data files,**” as claimed. An ordinarily skilled artisan would have been motivated to

provide the pause feature to allow the user to pause the display of content.

335. I note that the pause feature in Sagman takes the form of a keyboard shortcut rather than a control button visible displayed on the user interface, but this presents no obstacle to obviousness. On-screen control buttons were standard fixtures of graphical user interfaces and regarded as substitutes for keyboard shortcuts. An on-screen button would also have offered an advantage – because it is visible on the screen, the button would have provided the user with a clear visual cue of the availability of its corresponding feature. Keyboard shortcuts, on the other hand, require that the user already know about the feature and its shortcut and thus may be used less frequently or reserved for more advanced users. In any case, an ordinarily skilled artisan would have found it obvious to provide the pause feature in the form of a control button, consistent with the control buttons disclosed in Robertson and in Benedict and Fox as discussed for claim 8, and would have been motivated to do so to achieve the user interface advantages discussed above.

336. An ordinarily skilled artisan would have had at least a reasonable expectation of success. Benedict and Fox already disclose the ability to advance to the next node within a specified duration. Adapting the proposed combination to pause or reverse the order of node presentation in response to control button commands would have been trivially simple to implement. An ordinarily skilled

artisan would have found the disclosures of Robertson and Sagman sufficiently detailed to implement the features discussed above without undue experimentation.

J. Ground 9: Claim 21 is Obvious Over Shimizu, Benedict, Fox, and Filepp, in further view of Robertson and Sagman

337. Claim 21 is substantially similar to claim 9, discussed in the preceding section. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 21	Claim 9
The method in accordance with claim 20 <u>further comprising the step of receiving control button commands for pausing said display of said content, and for reversing the [sic] said display of said content and for fast forwarding said display of said content.</u>	The method in accordance with claim 8 <u>further comprising the step of receiving control button commands for pausing said display of said data files, and for reversing the sequence of said display of said data files and for fast forwarding said display of said data files.</u>

338. As shown, other than depending from different claims, claim 21 is substantially similar to claim 9, with the primary difference being that claim 21 recites display of said “**content**” whereas claim 8 recites display of said “**data files.**” As explained for claim 19, there is no meaningful difference as the contents recited in 21 are stored in the data files as discussed for claim 9. I also note that claim 21 recites “**reversing the [sic] said display of said content,**” which appears to be missing a word; I have interpreted this clause as consistent with claim 9, *i.e.*,

“**reversing the sequence of said display of**” said content.

339. As explained for claim 9 above, therefore, claim 21 would have been obvious based on the prior art cited against claim 20, in further view of Robertson and Sagman. The analysis from claim 9 applies equally here.

K. Ground 10: Claims 33-34 Are Obvious Over Shimizu in view of Benedict, Fox, Guinan, Filepp, in further view of Robertson and Sagman

340. Claim 33 is substantially similar to claim 9, discussed in the preceding section. The following side-by-side table shows the two claims (with common language shown in underlining):

Claim 33	Claim 9
<u>The method in accordance with claim 32 further comprising the step of receiving control button commands for pausing said display of said content, and for reversing the said display of said content and for fast forwarding said display of said content.</u>	<u>The method in accordance with claim 8 further comprising the step of receiving control button commands for pausing said display of said data files, and for reversing the sequence of said display of said data files and for fast forwarding said display of said data files.</u>

As shown, other than depending from different claims, claim 33 substantially similar to claim 9. The sole difference is that claim 33 recites display of “**content**” whereas claim 9 recites display of “**data files,**” which is an immaterial distinction.

341. As explained for claim 9 above, therefore, claim 33 would have been obvious based on the prior art cited against claim 32, in further view of Robertson

and Sagman. The analysis from claim 9 applies equally here.

342. With respect to claim 34, it recites “**the method in accordance with claim 32 further comprising the step of monitoring said user’s instructions corresponding to interruption of said show structure within said duration.**” As explained for claim 9 above, under the further combination with Sagman, it would have been obvious to allow the user to pause the display of the show structure. The user activating the pause feature qualifies as a “**user’s instructions corresponding to interruption of said show structure,**” because the pause feature would have caused interruption of the automatic display of the show structure under the proposed combination. These instructions also occur “**within said duration,**” because the user would have activated the pause feature while viewing the content associated with a node before the expiration of the display duration associated with that node.

L. Ground 11: Claim 40 is Obvious Over Shimizu in View of Benedict, Fox, and Miller, in Further View of Schloss

343. Claim 40 depends from claim 39, and further recites “**the method in accordance with claim 39, wherein prior to said step of automatically delivering the method further comprises the step of filtering the retrieved plurality of said resources to determine which of the retrieved resources are to be displayed.**”

344. Claim 40, and claim 39 from which it depends, are obvious based on

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Shimizu in view of Benedict, Fox, and Miller. I have cited Schloss as an additional theory of obvious with respect to claim 40.

345. **Schloss** explains that, because of the growth of the Web, “millions of users began to have access to millions of documents through the World Wide Web,” including documents “contain[ing] text, graphics, audio, video, etc.” (Schloss, 1:30-35.) But this access to large amounts of information created a challenge – which continues to this day – of how to prevent users from seeing content deemed undesirable or inappropriate content. “In surfing the Web,” for example, “children may have easy access to inappropriate videos, graphical data and other related information.” (Schloss, 1:57-59.)

346. Schloss purports to address this problem through a technique that allows the client system to filter content downloaded from the Web to determine which content to display. The general technique is straightforward. A client system running a web browser requests and downloads content from a content server on the Web. (Schloss, 4:30-33 (“More specifically, a client system running a Web Browser request content from a content server **6** using a Hypertext Transfer Protocol (HTTP) request and receiving the content in a HTTP response.”).) The client can then communicate with an advisory server **20** that transmits “characterization data” about the content to the client, which the client uses to make a filtering decision:

The advisory servers maintain one or more knowledge bases **22** that characterize the content generated by one or more of the content servers **6**. In addition, the system includes one or more clients **25** (three shown) running a Web Browser that when set in an advisory mode, for each content request to the content servers **6**, requests characterization data from one or more of the advisor servers **20**. The advisory servers **20** generate the appropriate characterization data based upon the information stored in the knowledge base **22**, and transmit the characterization data to the client **25**. Upon receiving the characterization data, the client **25** utilizes the characterization data to determine whether to filter the content data transmitted by the content server **6**.

(Schloss, 4:59-5:5.) The characterization data provided by advisory server **20** can indicate, for example, that the content is inappropriate for minors. (Schloss, 7:6-8 (“For example, the return code may indicate that the URL contains sexually explicit material not suitable for minors.”), 10:17-29 (“Not Appropriate for minors – violent”).)

347. Schloss discloses an alternative “advisory mode” in which the web browser inhibits downloading of the content until after it receives the corresponding characterization data from advisory server **20**. (Schloss, 5:24-33.) In a still further embodiment, the filtering functions of the browser are performed by a proxy server

that sits between the client and advisory server **20**. (Schloss, 11:58-12:1.) Under this embodiment, the proxy server requests the characterization data from advisory server **20** about the content transmitted by content server **6**, and uses the characterization data to determine whether or not to transmit the content to the client. (Schloss, 11:61-12:4.) “Upon receiving the characterization data, the proxy server utilizes the characterization data to determine whether to filter the content data transmitted by the content server **6**.” (Schloss, 12:1-4.)

348. *Manner, Rationale and Motivation to Combine (Ground 7 Prior Art with Schloss)*: It would have been obvious to combine the prior art cited for Ground 7 with Schloss. The proposed combination would have predictably resulted in **“filtering the received the retrieved plurality of said resources to determine which of the retrieved resources are to be displayed,”** as disclosed in Schloss. This filtering would also have occurred **“prior to said step of automatically delivering,”** which refers to the step of claim 39 of **“automatically delivering said content to said user for display on said user’s computer.”** I note that the delivering step of claim 39 requires **“delivering said content to said user,”** and under the proposed combination, the filtering would take place “prior to” the resources being delivered (*i.e.*, displayed) to the user. Moreover, even if the claim required the filtering to take place before physical transmission of the resources to

the user computer, this would have been obvious based on the proxy server embodiment in Schloss. Under the proposed combination based on this embodiment, for example, the resources would have been retrieved by the proxy server but filtered at the proxy server before they were transmitted to the user's computer over the network. Under either theory, claim 40 is obvious.

349. Schloss is an analogous reference in the field of presentation of content over the Internet, and would have been reasonably pertinent to problems facing the inventor. For example, Schloss would have been pertinent to techniques for filtering content to determine whether or not to display it to the user.

350. An ordinarily skilled artisan would have been motivated to combine in the manner described above to provide a further level of control over the content display. As was well-known and confirmed in Schloss, the Internet can contain a wide variety of content including content (such as adult-oriented or violent content) that may be undesirable and/or unsuitable for the viewer. The filtering techniques of Schloss would have provided the user with an additional level of control to prevent the delivery and display of unwanted content to the viewer. This would have had obvious utility for parents seeking to prevent minors or children from viewing potentially harmful content.

351. An ordinarily skilled artisan would have had at least a reasonable

expectation of success. The techniques described in Schloss all rely on well-known techniques including requesting and retrieving Web content over the Internet using HTTP, interfacing with advisory and proxy servers using well-known protocols such as TCP/IP, and other techniques that would have been within the basic knowledge of an ordinarily skilled artisan. The Background in Schloss, moreover, confirms that databases that characterized the content of websites were known. (Schloss, 1:64-2:11.) The disclosures of Schloss would have provided more than sufficient details to have enabled an ordinarily skilled artisan to implement the features on which I rely for the proposed combination, without undue experimentation.

M. Ground 12: Claim 45 is Obvious Over Shimizu in view of Benedict, Fox, and Miller, in further view of Ackermann and Resource Kit

352. Claim 45 depends from claim 44, and further recites “**the method in accordance with claim 44, further comprising the step of displaying a schedule window, for displaying a set of chosen line-up of show structures and time of their display, each corresponding to one of said categories.**” Claim 44 is obvious based on Shimizu in view of Benedict, Fox, and Miller. The additional limitations of claim 45 are obvious in further view of Resource Kit and Ackermann.

353. **Ackermann** describes features of the popular UNIX operating system. (Ackermann, pp.11-12.) As was well-known, the term “command line” (sometimes

referred to as a “shell”) refers to an interface used by operating systems to receive text-based commands from the user. Command line interfaces existed (and continue to exist) for a number of popular operating systems including UNIX, Windows, and many others, and were (and are) often used by developers and administrators to automate and schedule computing tasks.

354. Ackermann describes a well-known functionality of UNIX known as “cron” (short for chronograph) and “crontab” that allowed users to create schedules for performing tasks at certain times. (Ackermann, pp.225-226.) As further explained in Ackermann:

Use the crontab command to install a list of commands that will be executed on a regular schedule. The commands are scheduled to run at a specified time (such as once a month, once an hour, once a day, and so on). The list of commands to be performed on the specified scheduled must be included in the crontab file, which is installed with the crontab command. Once you install the crontab file, cron reads and executes the listed commands at the specified times.

(Ackermann, p.226.)

355. A crontab file follows a straightforward structure. Each line of a crontab file specifies time information for a particular command:

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

Following is the syntax for the commands you enter in a file to be used by crontab. The first five fields are time-option fields (you must specify all five of these fields; use an asterisk * in a field if you want to ignore that field). Table 11.3 lists the time-field options available with crontab.

minute hour day-of-month month-of-year day-of-week command

(Ackermann, p.227.) Ackermann provides the following concrete example:

To sort a file named /usr/wwr/sales/weekly and mail the output to a user named twool at 7:30 each Monday, use the following entry in a file:

```
30 07 * * 01 sort /usr/wwr/sales/weekly |mailx -s"Weekly Sales" twool
```

This command specifies the *minute* as 30, the *hour* as 07, any day of month with the *, any month of year with another *, and the *day-of-week* as 01 (which represents Monday).

(Ackermann, p.227.)

356. In the exemplary crontab list entry above, the first five pieces of information (“30 07 * * 01”) specify timing information for a particular command, and everything on that line after and including “sort” is the complete command line string of the command to be executed at 7:30AM each Monday. Ackermann explains that, to create a crontab file, the user can simply use a standard text editor such as “the graphical Text Editor that comes with UnixWare.” (Ackermann, p.226; *see also id.*, p.228 (explaining how to edit crontab files using a text editor.)) It would have been obvious that, when creating or editing a crontab file using a text editor, the file – and its corresponding crontab entries – would have been displayed to the

user. It would also have been obvious that when this occurs within a graphical user interface (such as using the graphical text editor mentioned in Ackermann), the schedule in the crontab file would have been displayed in a **window**, *e.g.*, the window of the UnixWare graphical Text Editor utilized to view and edit the crontab file. Ackermann therefore discloses and renders obvious “**displaying a schedule window, for displaying a set of chosen line-up of [commands] and time of their [execution].**” Ackermann does not expressly disclose a schedule for display of “**show structures,**” but was obvious in further view of Resource Kit.

357. **Resource Kit** describes features of the Microsoft Office suite of applications, including Microsoft PowerPoint. Resource Kit explains that, like Harvard Graphics discussed in Benedict, PowerPoint supported self-running slide presentations that could be moved automatically from slide-to-slide without user intervention. (Resource Kit, pp.338-339.) Resource Kit also describes a way to launch PowerPoint using straightforward commands executed at the command line. As noted, the commands listed in a crontab file discussed in Ackerman corresponded to commands that could have been executed by manual entry by the user via a command line interface. As I will explain below, it would have been obvious to combine Ackerman with Resource Kit to create a for causing PowerPoint to present one or more presentations at specified times based on a schedule.

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

358. Resource Kit confirms that, as was well-known, Microsoft Office applications could be instructed to launch in a particular way using a command line interface. Resource Kit provides an example of a command line string that instructs PowerPoint to launch and automatically begin displaying a particular presentation:

```
C:\MsOffice\PowerPnt\Powerpnt.exe /s c:\My Documents\MyKiosk.ppt
```

(Resource Kit, p.340, ¶3.) This command line string has three components: (1) the file path for the executable program for PowerPoint (Powerpnt.exe), (2) a command line switch (/s) that tells PowerPoint to automatically begin displaying a particular presentation, and (3) the name and path of the file of the presentation to display (MyKiosk.ppt). (Resource Kit, p.340; *see also id.*, p.341 (“**PowerPoint Command-Line Startup Switches.... /s Starts PowerPoint in Slide Show view and runs the specification presentation or presentations**”) (boldface in original).) The command line string above, therefore, would have instructed PowerPoint to start and display to the user the presentation whose file is named “MyKiosk.ppt.”

359. *Manner, Rationale and Motivation to Combine (Ground 7 Prior Art with Ackermann and Resource Kit)*: It would have been obvious to combine the prior art cited for Ground 7 with Ackermann and Resource Kit. The proposed combination would have predictably resulted in performing a further step of “**displaying a schedule window,**” *i.e.*, a window of a graphical text editor for editing

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

a schedule such as a crontab file, “**for displaying a set of chosen line-up of show structures and time of their display,**” *i.e.*, a list of entries for specifying times when the software program for presenting node content would be executed, and identifying the show structure to display. Take the following crontab file that I created for illustration containing with three entries based on Ackermann and Resource Kit:

```
00 07 * * * C:\MsOffice\PowerPnt\Powerpnt.exe /s c:\My Documents\Bill.PPT
00 10 * * * C:\MsOffice\PowerPnt\Powerpnt.exe /s c:\My Documents\Sally.PPT
00 13 * * * C:\MsOffice\PowerPnt\Powerpnt.exe /s c:\My Documents\Lucy.PPT
```

This example shows a schedule with three entries specifying how and when PowerPoint (Powerpnt.exe) will be executed: (1) at 7:00AM each day to automatically display a presentation named Bill.PPT, (2) at 10:00AM to automatically display Sally.PPT, and (3) at 1:00PM to automatically display Lucy.PPT. This crontab file would have been displayed to the user in a window, for example, when creating or editing it using a graphical text editor. The example above merely provides an illustration to visualize the straightforward way in which these references can be combined. Under the proposed combination, of course, “Powerpnt.exe” would have been replaced with the name of the software application that displays the show structure of nodes. (Shimizu, 9:63-66 (“This makes it easy to create illustratively an application program allowing the display of node contents to be switched in a desired order and at predetermined intervals on the hypertext

apparatus.”.) A PowerPoint presentation file (.PPT) is directly analogous to a show structure of nodes because it identifies a series of nodes and their content for display in a sequential manner. Under the proposed combination, this would have been replaced in the crontab file above with the name of the computer file(s) specifying the show structure that will be automatically displayed to the user when the display application is invoked. I note that, under the proposed combination, an ordinarily skilled artisan could have used the crontab format discussed in Ackermann or any other appropriate format for specifying the schedule for executing the show structure display application.

360. Finally, under the proposed combination, each of these show structures would “**each [have] correspond[ed] to one of said categories,**” as claimed. As explained for claim 39, it would have been obvious under the proposed combination with Miller to create a show structure of nodes related to a selected category. Because show structures were created under the proposed combination based on category-based searches, in the first instance, it would have been obvious that each of the show structures would have corresponded to at least one category. I note that claim 45 does not require that the category itself actually be displayed in the schedule window, although this would have been obvious and beneficial in order to inform the user of the category to which each show structure in each line-up relates.

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

361. Resource Kit is an analogous reference in the same field of providing organized or sequenced presentation of content, and both references would have been reasonably pertinent to a number of problems facing the inventor including how to automatically commence display of a show structure of nodes. Ackermann, for its part, is an operating system reference book that would have been widely applicable and pertinent to a number of problems facing the inventor, including how to implement a system in which a user can display a chosen line-up of show structures and specify times for their display.

362. An ordinarily skilled artisan would have been motivated to combine in the manner discussed above. Resource Kit characterizes the ability to automatically launch PowerPoint to display a particular presentation as a useful feature, particularly for self-running presentations that may run in a public or unattended setting. (Resource Kit, p.338-339.) An ordinarily skilled artisan would have been motivated to combine in order to obtain the flexibility and benefit of being able to schedule a line-up of show structures of nodes to be displayed at desired times, thus obviating the need for a human operator to manually commence the display at the appropriate times. This would also have provided the benefit of preventing human error, as the more complex the schedule and line-up, the more likely a human operator without the benefit of such automated scheduling may select the wrong

show structure and/or cause it to be displayed at the wrong time.

363. An ordinarily skilled artisan would have had at least a reasonable expectation of success. The proposed combination above, in fact, is based on a combination of two well-known and off-the-shelf tools. The “cron” facility was well-known, fully documented, and source code for that tool (as with other parts of UNIX) was widely available. It would have been trivial for an ordinarily skilled artisan to either use cron, or implement its own scheduling facility having similar or equivalent functionality, to carry out the proposed combination. Additionally, the ability of applications to launch and automatically perform specified tasks, such as automatically play particular content, was a well-known implementation and use a command line interfaces and command line arguments. The disclosures of Resource Kit and Ackermann would have provided more than sufficient details to have enabled an ordinarily skilled artisan to implement the proposed combination as described, without undue experimentation.

VI. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS

364. As explained in **Part III.B**, I understand that “secondary considerations” may be used as indicia of nonobviousness, provided that any such consideration has a nexus to the invention claimed. I understand from Petitioners’ counsel that Patent Owner in the related district court litigation has not yet identified

Declaration of Christopher M. Schmandt in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 7,155,451 B1

any evidence related to secondary considerations with a nexus to the claimed invention. Nor am I aware of information, such as commercial success, unexplained results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others, suggesting that the claims addressed in this Declaration are not obvious. To the extent Patent Owner later provides information it claims relates to secondary considerations, I reserve the right to supplement my analysis and opinions to comment on it.

VII. CONCLUSION

365. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in this proceeding. If required, I will appear for cross-examination at the appropriate time. I reserve the right to offer opinions relevant to the invalidity of the challenged claims at issue and/or offer testimony in support of this Declaration.

366. I hereby declare that all statements made herein of my own knowledge are true and that all statements are made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Dated: February 21, 2023

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Chris Schmandt', written over a horizontal line.

Christopher M. Schmandt
Winchester, MA

EXHIBIT A

Christopher Schmandt

4 Longfellow Road
Winchester, MA 01890
+1-617-230-4257

Education

M.I.T., Master of Science, Visual Studies (Computer Graphics), 1980

M.I.T., Bachelor of Science, Computer Science, 1978

Professional Experience -- MIT

Media Laboratory, Principal Research Scientist, 1985-2018 (retired)
Director, Living Mobile Research Group (formerly Speech + Mobility)

Architecture Machine Group, Research Associate, 1980-1984

Architecture Machine Group, Research Assistant, 1979-1980

Architecture Machine Group, Graphics Programmer, 1977-1979

Departmental Undergraduate Research Opportunities Program Coordinator, 1984-2018

Laboratory Intellectual Property Committee 2001-2016, chair 2002-2009

Departmental Committee on Graduate Studies, 1996-2001, 2007-2018

Sponsored Research Activities

Alerting and Mobile Messaging, Digital Life Consortium, MIT Media Lab, 1997-2018

Acoustical Cues to Discourse Structure, National Science Foundation, Principal Investigator, 1995-1998

Parsing Radio, News in the Future Consortium, MIT Media Lab, 1993-1999

Desktop Audio, SUN Microsystems, Inc., Principal Investigator, 1989-1996

Voice Interaction in Hand Held Computers, Apple Computer, Principal Investigator, 1991-1993

Voice Interfaces for Network Services, AT&T, Principal Investigator, 1989-1991

Back Seat Driver, NEC, Principal Investigator, 1988-1991

Acoustic and Visual Cues for Speech Recognition, DARPA, co-Principal Investigator, 1986-1988

Personal Computers and Telephony, NTT Public Corporation, Principal Investigator, 1984-1989

Home Telecomputing, Atari, Inc., Principal Investigator, 1983

Testifying legal engagements within the last five years, representing party in italics

- *Microsoft v. SpeakWare*, 2019
- *Bumble Trading LLC v. Match Group LLC*, 2019
- *Tile Inc. v. Cellwitch Inc.*, 2019
- *Blackberry Ltd v. Facebook, Inc* 2020
- *EROAD Ltd v. PerDiem Co.*, 2020
- *Snap Inc. v. SRK Technology LLC*, 2020
- *Shopify Inc Ltd v. Express Mobile Inc.*, 2020
- *Express Mobile Inc. v Wix Ltd* 2020
- *Motorola Mobility LLC v. Ironworks patents, LLC.* 2021
- *Express Mobile Inc. v GoDaddy Inc* 2021
- *Bumble Trading LLC v. Kinectus, LLC.* 2021
- *Tile, Inc v. Linquet Technologies, Inc.* 2021
- *Facebook, Inc. v. Palo Alto Research Center, Inc.* 2021
- *Unified Patents, LLC v. Gesture Technology Partners, LLC.* 2021
- *Quantum Metric, Inc v. Content Square Isreal, Ltd.* 2021
- *Meta Platforms Inc. v. AlmondNet, Inc* 2022
- *Apple Inc. v. Zentian Limited* 2022

Professional Experience -- Intellectual Property

Co-inventor on U.S. Patents

- 5,177,685 for "Automobile navigation system using real time spoken driving instructions"
- 6,728,348 for "System for storing voice recognizable identifiers using a limited input device such as a telephone key pad"
- 6,937,986 for "Automatic dynamic speech recognition vocabulary based on external sources of information"
- 7,098,776 for "Methods and apparatus for vibrotactile communication"
- 7,392,280 for "Method for summarization of threads in electronic mail"
- 7,738,637 "Interactive voice message retrieval"
- 7,443,283 for "Methods and apparatus for connecting an intimate group by exchanging awareness cues and text, voice instant messages, and two-way voice communications"
- 7,865,560 for "System for summarization of threads in electronic mail"
- 8,121,653 for "Methods and apparatus for autonomously managing communications using an intelligent intermediary"
- 8,135,128 for "Animatronic creatures that act as intermediaries between human users and a telephone system"

Co-inventor on U.S. Patent Applications

- 20020076009 "International dialing using spoken commands"
- 20030023688 "Voice-based message sorting and retrieval method"
- 20030081738 "Method and apparatus for improving access to numerical information in voice messages"
- 20030144846 "Method and system for modifying the behavior of an application based upon the application's grammar"
- 20030158903 "Calendar bar interface for electronic mail interaction"

Publications

Field Study of a Tactile Sound Awareness Device for Deaf and Hard of Hearing Users ISWC 2020. (with D. Jain, B. Chiu, S. Goodman, L. Findlater, J. Froehlich)

SkinMorph: Texture-Tunable On-Skin Interface Through Thin, Programmable Gel ISWC 2018. (with Cindy Kao, M. Banforth, D. Kim)

Technical Interventions to Detect, Communicate, and Deter Sexual Assault. ISWC 2017. (with Manisha Mohan)

Exploring Interactions and Perceptions of Kinetic Wearables. DIS 2017. (with Cindy Hsin-Liu Kao, D. Ajilo, O. Anilonyte, A. Dementyev, I. Choi and S. Follmer)

Leveraging User-made Predictions to Help Understand Personal Behavior. MobileHCI 2017. (with Miriam Greis, Tilman Dangler and Albrecht Schmidt)

Rovables: On-Body Robots as Mobile Wearables. UIST 2016. (with Cindy Hsin-Liu Kao, A. Dementyev, I. Choi, D. Ajilo, M. Xu, and S. Follmer)

DuoSkin: Rapidly Prototyping On-Skin User Interfaces Using Skin-Friendly Materials. ISWC 2016. (with Cindy Hsin-Liu Kao, Christian Holz, Asta Roseway, and Andres Calvo)

Immersive Terrestrial Scuba Diving Using Virtual Reality (with Dhruv Jain, Misha Sra, Jingru Go, Rodrigo Margues, Raymond Wu and Justin Chiu) Proceedings, UIST 2016

Expanding social mobile games beyond the device screen (with Misha Sra) Journal of Personal and Ubiquitous Computing, 2015

NailO: Fingernails as an input surface (with Cindy Hsin-Liu Kao, Artem Dementyev, Joseph Paradiso) CHI 2015

Mugshots: A mug display for front and back stage social interaction in the workplace (with Cindy Hsin-Liu Kao) TEI (Tangible and Embedded Interfaces) 2015

Mime: compact, low power 3D gesture sensing for interaction with head mounted displays (with Andrea Colaco, Ahmend Kirmani, Hye Soo Yang, Nan-Wei Gong, and Vivek Goyal) Proceedings of UIST 2013.

Spotz: A location-based approach to self-awareness (with Misha Sra) Proceedings of Persuasive 2013.

Setting the stage for interaction: A tablet application to augment group discussion in a seminar class (with Drew Harry and Eric Gordon) Proceedings of CSCW 2012.

Indoor Location Sensing using Geo-Magnetism (with Jaewoo Chung, Matt Donahoe, Ig-Jae Kim, Pedram Razavai and Micaela Wiseman) Proceedings of International Conference on Mobile Systems, Applications, and Services (Mobisys) 2011.

My second bike: a TV-enabled social and interactive riding experience (with Jaewoo Chung, Kuang Xu, Andrea Colaco, and Victor Li) Proceedings of IEEE Communications and Networking Conference, Jan 2010.

Going my way?: User-aware route planner (with Jaewoo Chung) Proceedings CHI 2009.

Globetoddler: Designing for remote interaction between preschoolers and their traveling parents (with Paulina Modlitba) CHI 2008 Extended Abstracts

Are we there yet? - a temporally aware media player (with Matt Adcock and Jaewoo Chung), Australian User Interface Conference (AUIC) 2008

Physical embodiments for mobile communication agents (with Stefan Marti), UIST 2005

Giving the caller the finger: collaborative responsibility for cellphone interruptions (with Stefan Marti)
Extended Abstracts, CHI 2005

Active Messenger: email filtering and delivery in a heterogeneous network (with Stefan Marti) Human-Computer Interaction Journal (HCI) Volume 20 (2005)

WatchMe: communication and awareness between members of a closely-knit group (with Natalia Marmasse)
Proceedings of UbiComp 2004

An audio-based personal memory aid (with S. Vemuri, W. Bender, S. Tellex and B. Lassey) Proceedings of UbiComp 2004

Improving speech playback using time-compression and speech recognition (with Sunil Vemuri, Philip DeCamp, and Walter Bender) Proceedings of CHI 2004

Impromptu: managing networked audio applications for mobile users (with Kwan Lee, Jang Kim, and Mark Ackerman), Proceedings of MobiSys 2004

TalkBack: a conversational answering machine (with Vidya Lakshminpathy and Natalia Marmasse) Proceedings of UIST 2003

"ListenIn" to domestic environments from remote locations (with Gerardo Vallejo) Proceedings of the 2003 International Conference on Auditory Display (ICAD)

Safe & Sound: a wireless leash (with Natalia Marmasse) Extended Abstracts, Proceedings of CHI 2003

Mediated voice communication via mobile IP (with Jang Kim, Kwan Lee, Gerardo Vallejo, and Mark Ackerman), Proceedings of UIST 2002.

The Audio Notebook: Paper and pen interaction with structured speech (with Lisa Stifelman and Barry Arons), Proceedings of CHI 2001

Synthetic News Radio (with Keith Emnett) IBM Systems Journal, Vol. 39 Nos. 3-4, pp. 646-659, 2000.

Everywhere messaging (with Natalia Marmasse, Stefan Marti, Nitin Sawhney, and Sean Wheeler) IBM Systems Journal, Vol. 39 Nos. 3-4, pp. 660-677, 2000.

Location-aware information delivery with comMotion (with Natalia Marmasse), Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing, pp. 157-171, Springer, 2000.

Nomadic Radio: Scalable and contextual notification for wearable audio messaging (with Nitin Sawhney), Proceedings of CHI 1999.

Speaking and listening on the run: Design for wearable audio computing (with Nitin Sawhney), Proceedings of International Symposium on Wearable Computing, 1998.

Audio Hallway: A virtual acoustic environment for browsing, Proceedings of UIST 1998.

Dynamic Soundscape: Mapping time to space for audio browsing (with Minoru Kobayashi), Proceedings of CHI 1997.

CLUES: Dynamic personalized message filtering (with Matt Marx), Proceedings of CSCW 1996.

Using acoustic structure in a hand-held audio playback device (with Deb Roy), IBM Systems Journal, Vol 35, Nos. 3 and 4, 1996.

Mailcall: Message presentation and navigation in a nonvisual environment (with Matt Marx), Proceedings of CHI 1996

AudioStreamer: Exploiting simultaneity for listening (with Atty Mullins), short paper, CHI 1995.

Multimedia momadic services on today's hardware, IEEE Network, September/October 1994.

Putting people first: Specifying proper names in speech interfaces (with Matt Marx), proceedings of UIST 1994.

Chatter: A conversational learning speech interface (with E. Ly) AAAI Workshop on Intelligent Multi-Media Multi-Modal Systems, 1994.

Voice Communication with Computers: Conversational Systems. New York: Van Nostrand Reinhold. 1994.

Capturing, structuring, and representing ubiquitous audio (with D. Hindus and C. Horner), ACM Transactions on Information Systems, Vol. 11, No. 4, October 1993.

Speech Recognition Architectures for Multimedia Environments, (with E. Ly and B. Arons), Proceedings of the 1993 AVIOS Conference, September 1993.

Phoneshell: the Telephone as Computer Terminal, Proceedings of the ACM Multimedia Conference, August 1993.

Voicenotes: A Speech Interface for a Hand-Held Voice Notetaker (with L. Stifelman, B. Arons, and E. Hulteen), Proceedings of INTERCHI'93, April 1993.

From Desktop Audio to Mobile Access: Opportunities for Voice in Computing, book chapter in Advances in Human-Computer Interaction, Vol. 4, H.R. Hartson and D. Hix editors. 1992.

Ubiquitous Audio: Capturing Spontaneous Collaboration (with D. Hindus), Proceedings of CSCW'92, November 1992.

Integrating Audio and Telephony in a Distributed Workstation Environment (with S. Angebrannndt, R. Hyde, D. Luong, and N. Siravara), Proceedings of the Summer 1991 USENIX Conference, June 1991.

Augmenting a Window System with Speech Input (with M. Ackerman and D. Hindus), Computer, IEEE Computer Society, Vol. 23, No. 8, August 1990.

Observations on Using Speech Input for Window Navigation (with D. Hindus, M. Ackerman, and S. Manandhar), Proceedings, Human-Computer Interaction, Interact '90, IFIP, August 1990.

Phonetool: Integrating Telephones and Workstations (with S. Casner), Proceedings, GLOBECOM '89, IEEE Communications Society, November 1989.

Desktop Audio (with B. Arons), UNIX Review, October 1989.

Synthetic Speech for Real Time Direction-Giving (with J. Davis), IEEE Transactions on Consumer Electronics, IEEE, September 1989.

The Back Seat Driver: Real Time Spoken Driving Instructions (with J. Davis), Proceedings, IEEE Vehicle Navigation and Information Systems Conference, IEEE, Toronto, Canada, September 1989.

An Audio and Telephone Server for Multi-media Workstations (with M. McKenna), Proceedings, Second IEEE Conference on Workstations, IEEE, Palo Alto, CA., 1988.

Employing Voice Back Channels to Facilitate Audio Document Retrieval, Proceedings, ACM Conference on Office Information Systems (COIS), Santa Clara, CA, 1988.

Conversational Telecommunications Environments, Proceedings, Second International Conference on Human-Computer Interaction, 1987.

Understanding Speech Without Recognizing Words, Proceedings, American Voice Input/Output Society Conference, AVIOS, 1987.

A Robust Parser and Dialog Generator for a Conversational Office System (with B. Arons and C. Simmons), Proceedings, American Voice Input/Output Society Conference, AVIOS, Palo Alto, CA, 1987.

Integrated Messages and Network Services for a Personal Workstation, IEEE Workshop on Telematics and Message Handling Systems, IEEE, 1986.

Voice Interaction in an Integrated Office and Telecommunications Environment, Proceedings, American Voice Input/Output Society Conference, AVIOS, San Francisco, CA, 1985.

Voice Communication with Computers, book chapter in *Advances in Human-Computer Interaction*, H. R. Hartson ed., 1985.

Speech Synthesis Gives Voiced Access to an Electronic Mail System, *Speech Technology*, Vol. 2, No. 3, Aug/Sept 1984.

A Conversational Telephone Messaging System (with B. Arons), *IEEE Transactions on Consumer Electronics*, IEEE, Vol CE-30, August 1984.

Phone Slave: A Graphical Telecommunications Interface (with B. Arons), Proceedings, Society for Information Display International Symposium, SID, San Francisco, CA, June 1984.

Input/Display Registration in a Stereoscopic Workstation, *Displays*, April 1984.

Remote Access to Voice and Text Messages, Proceedings, American Voice Input/Output Society Conference, AVIOS, Washington D.C., 1984.

Fuzzy Fonts: Analog Models Improve Digital Text Quality, Proceedings, National Computer Graphics Association Conference, Chicago, IL, 1983.

Greyscale Fonts Designed From Video Signal Analysis, Society of Applied Learning Technology, Houston, TX, 1983.

Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station, Proceedings, ACM/SIGGRAPH, Detroit, MI, 1983.

A Programmable Virtual Vocabulary Speech Processing Peripheral (with W. Bender), Proceedings, American Voice Input/Output Society Conference on Voice Data Entry Systems Applications, AVIOS, 1983.

The Intelligent Voice Interactive Interface (with E. A. Hulteen), Proceedings, Human Factors in Computer Systems, National Bureau of Standards/ACM, Gaithersburg, MD, 1982.

Interactive Three-Dimensional Computer Space, Proceedings, SPIE Conference on Processing and Display of Three-Dimensional Data, SPIE, San Diego, CA, 1982, Vol. 367.

Speech Communications, a Systems' Approach, Proceedings, American Voice Input/Output Society Conference on Entry Systems Applications, 1982.

Voice Interaction: Putting Intelligence into the Interface, Proceedings, IEEE International Conference on Cybernetics and Society, IEEE, Seattle, WA, 1982.

The Intelligent Ear: A Graphical Interface to Digital Audio, Proceedings, IEEE International Conference on Cybernetics and Society, IEEE, Atlanta, GA, 1981.

Soft Typography, Information Processing 1980, IFIPS, S. Lavington ed., North-Holland Publishing Co., 1980.
