UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FACEBOOK, INC.,
Petitioner

v.

EXPRESS MOBILE INC.,
Patent Owner

U.S. Patent No. 7,594,168 B2
Issue Date: September 22, 2009

Title: Browser Based Web Site Generation Tool and Run Time Engine

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT 7,594,168 B2**

Table of Contents

Page

I. MANDATORY NOTICES UNDER §42.8(A)(1) 1
A. Real Party-In-Interest under §42.8(b)(1) 1
B. Related Matters under §42.8(b)(2) 1
C. Lead and Back-Up Counsel under §42.8(b)(3) 8
D. Service Information 9
II. FEE PAYMENT 9
III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D) 10
A. Grounds for Standing 10
B. Identification of Challenge and Statement of Precise Relief Requested 10
C. Considerations Under §§ 314(a) and 325(d) 10
IV. PATENT OVERVIEW 12
A. Level of Ordinary Skill 12
B. Specification 13
V. CLAIM CONSTRUCTION 15
VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE 18
A. Overview of Grounds 18
B. Ground 1: Obviousness of Claims 1-4 and 6 Over Reynolds, Lemay Web, Lemay Java, Perry, and Witkowski 20
1. Claim 1 20
(a) "a server comprising a build engine configured to:" (Claim 1[a]) 22
(b) "accept user input to create a web site, the web site comprising a plurality of web pages, each web page comprising a plurality of objects," (Claim 1[b]) 23

Table of Contents
(continued)

Page

(c) "accept user input to associate a style with objects of the plurality of web pages, wherein each web page comprises at least one button object or at least one image object, and wherein the at least one button object or at least one image object is associated with a style that includes values defining transformations and time lines for the at least one button object or at least one image object;" (Claim 1[c]).....30

(d) "and wherein each web page is defined entirely by each of the plurality of objects comprising that web page and the style associated with the object," (Claim 1[d]) 40

(e) "produce a database with a multidimensional array comprising the objects that comprise the web site including data defining, for each object, the object style, an object number, and an indication of the web page that each object is part of, and" (Claim 1[e]).....43

(f) "provide the database to a server accessible to [sic; a] web browser;" (Claim 1[f]) 57

(g) "wherein the database is produced such that a web browser with access to a runtime engine is configured to generate the web-site from the objects and style data extracted from the provided database." (Claim 1[g]).....59

2. Claim 2: "The system of claim 1, wherein one of said plurality of objects is a child, and wherein the build engine is configured to accept user input to associate a style with child button and child image objects.".....62

3. Claim 3: "The system of claim 2, wherein at least one of said styles includes values defining time lines for child button and child image objects.".....67

Table of Contents
(continued)

Page

4. Claim 4: "The system of claim 1, wherein at least one of said styles includes settings for multiple object states."....67

5. Claim 6: "The system of claim 1, where said data is stored as one or more of a Boolean an integer, a string, a floating point variables, or a URL." 68

C. Ground 2: Obviousness of Claims 1-4 and 6 Over Reynolds, Lemay Web, Lemay Java, Perry, Witkowski, and Coombs 73

1. Claim 1 73

2. Claim 2: "The system of claim 1, wherein one of said plurality of objects is a child, and wherein the build engine is configured to accept user input to associate a style with child button and child image objects."....78

3. Claim 3: "The system of claim 2, wherein at least one of said styles includes values defining time lines for child button and child image objects."....79

4. Claim 4: "The system of claim 1, wherein at least one of said styles includes settings for multiple object states."....79

5. Claim 6: "The system of claim 1, where said data is stored as one or more of a Boolean an integer, a string, a floating point variables, or a URL." 79

VII. CONCLUSION....79

Express Mobile, Inc., Exhibit 2009
Page 2009 - 4
IPR2022-00597, Google LLC. v. Express Mobile, Inc.

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 7,594,168 B2 to Steven R. Rempell (filed January 24, 2003, issued September 22, 2009) ("'168" or "'168 patent")
1002	Declaration of Vijay K. Madisetti, Ph.D. ("Madisetti")
1003	U.S. Patent No. 6,052,717 to Brian Reynolds et al. (filed October 23, 1996, issued April 18, 2000)
1004	Excerpts from Laura Lemay et al., *Teach Yourself Web Publishing with HTML 3.2 in a Week* (3d ed.1996)
1005	Excerpts from Laura Lemay et al., *Teach Yourself Java in 21 Days* (1996)
1006	Excerpts from Greg Perry, *C++ Programming 101* (1992)
1007	U.S. Patent No. 6,125,360 to Andrew Witkowski et al. (filed July 2, 1998, issued September 26, 2000)
1008	Excerpts from Jason Coombs et al., *Setting up an Internet Site for Dummies* (3d ed. 1998)
1009	Excerpts from *Microsoft Press Computer Dictionary* (3d ed. 1997)
1010	*Reserved*
1011	Prosecution History of U.S. Patent No. 7,594,168 B2
1012	Complaint for Patent Infringement (Dkt. 1) in *Express Mobile, Inc. v. Facebook, Inc.*, Case No. 6:20-cv-00803-ADA (W.D. Tex. Sept. 1, 2020)
1013	Scheduling Order (Dkt. 46) in *Express Mobile, Inc. v. Facebook, Inc.*, Case No. 6:20-cv-00803-ADA (W.D. Tex. April 6, 2021)

List of Exhibits

Exhibit No.	Description of Document
1014	Preliminary Infringement Contentions and Identification of Priority Dates, served February 5, 2021, in *Express Mobile, Inc. v. Facebook, Inc.*, Case No. 6:20-cv-00803-ADA (W.D. Tex.)
1015	*Reserved*
1016	Amended Order Construing Claims (Dkt. 81) in *X Commerce, Inc. d/b/a Magento, Inc. v. Express Mobile, Inc.*, No. 17-cv-02605 (N.D. Cal. Sept. 12, 2018)
1017	Oral Order (Dkt. 35) in *Express Mobile, Inc. v. eGrove Systems Corporation*, No. 1-17-cv-00703 (D. Del. Aug. 22, 2018.)
1018	Claim Construction Memorandum Opinion and Order (Dkt. 97) in *Express Mobile, Inc. v. Svanaco, Inc.*, No. 2-17-cv-00130 (E.D. Tex. Feb. 7, 2018)
1019	*Reserved*
1020	Order on Claim Construction (Dkt. 142) in *Shopify Inc. et al v. Express Mobile, Inc.*, No. 1-19-cv-00439 (D. Del. June 30, 2020)
1021	Memorandum Opinion (Dkt. 121) in *Express Mobile v. GoDaddy.Com, LLC*, No. 1-19-cv-01937 (D. Del. June 1, 2021)
1022	Request for Ex Parte Reexamination of U.S. Patent No. 7,594,168, Control No. 90/014,583 (September 25, 2020)
1023	*Reserved*
1024	Declaration of Sylvia Hall-Ellis, Ph.D.
1025	Joint Claim Construction Chart (Dkt. 23) in *Express Mobile, Inc. v. eGrove Systems Corporation*, No. 1-17-cv-00703 (D. Del. April 6, 2018.)

List of Exhibits

Exhibit No.	Description of Document
1026	Joint Claim Construction Brief (Dkt. 64) in *Express Mobile, Inc. v. GoDaddy.Com, LLC*, No. 1:19-cv-01937 (D. Del. March 19, 2021)
1027	Defendant's Amended List of Proposed Claim Constructions in *Express Mobile, Inc. v. Facebook, Inc. et al*, No. 6-20-cv-00803 (W.D. Tex. May 12, 2021)

I. MANDATORY NOTICES UNDER §42.8(A)(1)

A. Real Party-In-Interest under §42.8(b)(1)

Facebook, Inc. is the real party-in-interest to this *inter partes* review ("IPR") petition.

B. Related Matters under §42.8(b)(2)

Petitioner is unaware of any other IPR proceedings directed to the '168 patent. The '168 patent is the subject of a pending *ex parte* reexamination (Control No. 90/014,583), filed by third party Unified Patents, LLC. The '168 patent is also the subject of the following co-pending civil actions:

1. *Express Mobile, Inc. v. Facebook, Inc.*, No. 6-20-cv-00803 (W.D. Tex.)
2. *Express Mobile, Inc. v. Google LLC*, No. 6-20-cv-00804 (W.D. Tex.)
3. *Express Mobile, Inc. v. Salesforce.com, Inc.*, No. 3-20-cv-08461 (N.D. Cal.)
4. *Express Mobile, Inc. v. Booking.com B.V.*, No. 3-20-cv-08491 (N.D. Cal.)
5. *Express Mobile, Inc. v. SAP SE*, No. 3-20-cv-08492 (N.D. Cal.)
6. *Express Mobile, Inc. v. Oath Holdings Inc. f/k/a Yahoo!*, No. 3-20-cv-08321 (N.D. Cal.)
7. *Express Mobile, Inc. v. Adobe Inc. d/b/a Adobe Systems Inc.*, No. 3-20-cv-08297 (N.D. Cal.)
8. *Express Mobile, Inc. v. HubSpot, Inc.*, No. 1-20-cv-01162 (D. Del.)
9. *Express Mobile, Inc. v. Squarespace, Inc.*, No. 1-20-cv-01163 (D. Del.)
10. *Express Mobile, Inc. v. Microsoft Corp.*, No. 3-20-cv-06152 (N.D. Cal.)
11. *Express Mobile, Inc. v. Expedia, Inc.*, No. 6-20-cv-00801 (W.D. Tex.)

12. *Express Mobile, Inc. v. eBay Inc.*, No. 6-20-cv-00802 (W.D. Tex.)

13. *Express Mobile, Inc. v. Web.com Group, Inc.*, No. 3-20-cv-00839 (M.D. Fl.)

14. *Express Mobile, Inc. v. GoDaddy.Com, LLC*, No. 1-19-cv-01937 (D. Del.)

15. *Express Mobile, Inc. v. Wix.com, Ltd.*, No. 3-19-cv-06559 (N.D. Cal.)

16. *Shopify Inc. v. Express Mobile, Inc.*, No. 1-19-cv-00439 (D. Del.)

17. *X Commerce, Inc. d/b/a Magento, Inc. v. Express Mobile, Inc.*, No. 3-17-cv-02605 (N.D. Cal.)

The ’168 patent was also formerly the subject of the following civil actions:

1. *Express Mobile Inc. v. BigCommerce, Inc.*, No. 2-16-cv-00384 (E.D. Tex.) (terminated June 30, 2016)

2. *Express Mobile, Inc. v. Webs, Inc.*, No. 1-16-cv-00160 (E.D. Tex.) (terminated June 2, 2016)

3. *Express Mobile, Inc. v. Webs, Inc.*, No. 2-16-cv-00558 (E.D. Tex.) (terminated Jan. 9, 2017)

4. *Express Mobile, Inc. v. Jiva Infotech, Inc. d/b/a I95DEV*, No. 2-16-cv-00775 (E.D. Tex.) (terminated Jan. 4, 2017)

5. *Express Mobile, Inc. v. Amplifi Commerce, LLC*, No. 2-16-cv-00811 (E.D. Tex.) (terminated Jan. 4, 2017)

6. *Express Mobile, Inc. v. Weebly, Inc.*, No. 2-16-cv-00906 (E.D. Tex.) (terminated Oct. 28, 2016)

7. *Express Mobile, Inc. v. Digital Evolution Group, LLC*, No. 2-16-cv-00922 (E.D. Tex.) (terminated Feb. 27, 2017)

8. *Express Mobile, Inc. v. Astound Commerce Corporation*, No. 2-16-cv-00923 (E.D. Tex.) (terminated Jan. 4, 2017)

9. *Express Mobile, Inc. v. OSF Global Services Inc.*, No. 2-16-cv-00924

(E.D. Tex.) (terminated Jan. 4, 2017)

10. *Express Mobile, Inc. v. Blue Acorn, Inc.*, No. 2-16-cv-01411 (E.D. Tex.) (terminated Feb. 27, 2017)

11. *Express Mobile, Inc. v. Guidance Solutions, Inc.*, No. 2-16-cv-01412 (E.D. Tex.) (terminated May 12, 2017)

12. *Express Mobile, Inc. v. Coalition Technologies LLC*, No. 2-16-cv-01413 (E.D. Tex.) (terminated Feb. 9, 2017)

13. *Express Mobile, Inc. v. Lyons Holding Co. Inc.*, No. 2-16-cv-01414 (E.D. Tex.) (terminated Mar. 13, 2017)

14. *Express Mobile, Inc. v. Volusion, LLC*, No. 2-17-cv-00064 (E.D. Tex.) (terminated May 12, 2017)

15. *Express Mobile, Inc. v. WaveMaker, Inc.*, No. 2-17-cv-00065 (E.D. Tex.) (terminated Apr. 14, 2017)

16. *Express Mobile, Inc. v. Alpine Consulting, Inc.*, No. 2-17-cv-00126 (E.D. Tex.) (terminated June 14, 2017)

17. *Express Mobile, Inc. v. Forix LLC*, No. 2-17-cv-00127 (E.D. Tex.) (terminated May 17, 2017)

18. *Express Mobile, Inc. v. KTree Computer Solutions Inc.*, No. 2-17-cv-00128 (E.D. Tex.) (terminated June 13, 2017)

19. *Express Mobile, Inc. v. Optaros, Inc.*, No. 2-17-cv-00129 (E.D. Tex.) (terminated June 27, 2017)

20. *Express Mobile, Inc. v. Svanaco, Inc.*, No. 2-17-cv-00130 (E.D. Tex.) (terminated May 29, 2018)

21. *Express Mobile, Inc. v. BigCommerce, Inc.*, No. 2-17-cv-00160 (E.D. Tex.) (terminated June 5, 2018)

22. *Express Mobile, Inc. v. Alpine Consulting, Inc.*, No. 1-17-cv-03815 (N.D. Ill.) (terminated July 6, 2017)

23. *Express Mobile, Inc. v. Brainvire Infotech, Inc.*, No. 1-17-cv-00702 (D. Del.) (terminated Oct. 4, 2017)

24. *Express Mobile, Inc. v. eGrove Systems Corporation*, No. 1-17-cv-00703 (D. Del.) (terminated Feb. 28, 2019)

25. *Express Mobile, Inc. v. Icreon Tech, Inc.*, No. 1-17-cv-00704 (D. Del.) (terminated Oct. 27, 2017)

26. *Express Mobile, Inc. v. Mobikasa, LLC*, No. 1-17-cv-00705 (D. Del.) (terminated Oct. 4, 2017)

27. *Express Mobile, Inc. v. Salzer Technologies, Inc.*, No. 1-17-cv-00706 (D. Del.) (terminated Nov. 27, 2017)

28. *Express Mobile, Inc. v. ePages Inc.*, No. 1-17-cv-00707 (D. Del.) (terminated Oct. 12, 2017)

29. *Express Mobile, Inc. v. Webflow, Inc.*, No. 1-17-cv-00708 (D. Del.) (terminated Oct. 4, 2017)

30. *Express Mobile, Inc. v. AppGyver Inc.*, No. 1-17-cv-00710 (D. Del.) (terminated Sept. 1, 2017)

31. *Express Mobile, Inc. v. Domani Studios LLC*, No. 1-18-cv-00102 (D. Del.) (terminated Aug. 24, 2018)

32. *Express Mobile, Inc. v. Manifest LLC*, No. 1-18-cv-00103 (D. Del.) (terminated July 3, 2018)

33. *Express Mobile, Inc. v. SG Hosting Inc.*, No. 1-18-cv-00104 (D. Del.) (terminated Aug. 27, 2018)

34. *Express Mobile, Inc. v. Rockfish Interactive LLC*, No. 1-18-cv-00105 (D. Del.) (terminated Aug. 15, 2018)

35. *Express Mobile, Inc. v. WPEngine, Inc.*, No. 1-18-cv-00106 (D. Del.) (terminated Aug. 14, 2018)

36. *Express Mobile, Inc. v. Fulcrum Worldwide Inc.*, No. 1-18-cv-00154 (D. Del.) (terminated June 4, 2018)

37. *Express Mobile, Inc. v. FiloBlu USA Corp.*, No. 1-18-cv-00155 (D. Del.) (terminated June 22, 2018)

38. *Express Mobile, Inc. v. Axelerant Technologies, Inc.*, No. 1-18-cv-

00156 (D. Del.) (terminated July 5, 2018)

39. *Express Mobile, Inc. v. BigCommerce, Inc.*, No. 3-18-cv-03287 (N.D. Cal.) (terminated July 23, 2019)

40. *Express Mobile, Inc. v. Alloy Marketing and Promotions, LLC*, No. 1-18-cv-01166 (D. Del.) (terminated Aug. 24, 2018)

41. *Express Mobile, Inc. v. Big Spaceship LLC*, No. 1-18-cv-01167 (D. Del.) (terminated Jan. 29, 2019)

42. *Express Mobile, Inc. v. Blue State Digital Inc.*, No. 1-18-cv-01168 (D. Del.) (terminated Jan. 2, 2019)

43. *Express Mobile, Inc. v. Born Group, Inc.*, No. 1-18-cv-01169 (D. Del.) (terminated May 8, 2019)

44. *Express Mobile, Inc. v. Blackstone Technology Group Inc.*, No. 3-18-cv-04678 (N.D. Cal.) (terminated Oct. 25, 2018)

45. *Express Mobile, Inc. v. Code and Theory LLC*, No. 3-18-cv-04679 (N.D. Cal.) (terminated Apr. 22, 2019)

46. *Express Mobile, Inc. v. Globant, SA*, No. 3-18-cv-04681 (N.D. Cal.) (terminated Apr. 4, 2019)

47. *Express Mobile, Inc. v. Happiest Minds Technologies Pvt. Ltd.*, No. 3-18-cv-04683 (N.D. Cal.) (terminated Jan. 24, 2019)

48. *Express Mobile, Inc. v. Huge, LLC*, No. 3-18-cv-04687 (N.D. Cal.) (terminated June 17, 2019)

49. *Express Mobile, Inc. v. Pantheon Systems Inc.*, No. 3-18-cv-04688 (N.D. Cal.) (terminated Feb. 14, 2019)

50. *Express Mobile, Inc. v. SoftVision Inc.*, No. 3-18-cv-04693 (N.D. Cal.) (terminated Feb. 7, 2020)

51. *Express Mobile, Inc. v. Svitla Systems Inc.*, No. 3-18-cv-04694 (N.D. Cal.) (terminated Oct. 12, 2018)

52. *Express Mobile, Inc. v. DreamHost, LLC*, No. 1-18-cv-01173 (D. Del.) (terminated Aug. 13, 2019)

53. *Express Mobile, Inc. v. Hill Holliday LLC*, No. 1-18-cv-01174 (D. Del.) (terminated Aug. 5, 2019)

54. *Express Mobile, Inc. v. Hostway Services, Inc.*, No. 1-18-cv-01175 (D. Del.) (terminated Aug. 8, 2019)

55. *Express Mobile, Inc. v. iCrossing, Inc.*, No. 1-18-cv-01176 (D. Del.) (terminated Aug. 7, 2020)

56. *Express Mobile, Inc. v. Liquid Web, LLC*, No. 1-18-cv-01177 (D. Del.) (terminated Aug. 7, 2019)

57. *Express Mobile, Inc. v. Martin Retail Group, LLC*, No. 1-18-cv-01178 (D. Del.) (terminated Sept. 24, 2018)

58. *Express Mobile, Inc. v. Mondo International, LLC*, No. 1-18-cv-01179 (D. Del.) (terminated Dec. 6, 2018)

59. *Express Mobile, Inc. v. McCann Relationship Marketing, LLC d/b/a MRM McCann*, No. 1-18-cv-01180 (D. Del.) (terminated Aug. 5, 2019)

60. *Express Mobile, Inc. v. Namecheap, Inc.*, No. 1-18-cv-01181 (D. Del.) (terminated Sept. 16, 2019)

61. *Express Mobile, Inc. v. Publicis Hawkeye, Inc.*, No. 1-18-cv-01182 (D. Del.) (terminated Sept. 24, 2019)

62. *Express Mobile, Inc. v. Rackspace Hosting, Inc.*, No. 1-18-cv-01183 (D. Del.) (terminated May 29, 2019)

63. *Express Mobile, Inc. v. Gaddis Partners, Ltd. d/b/a T3*, No. 3-18-cv-05058 (N.D. Cal.) (terminated May 6, 2019)

64. *Express Mobile, Inc. v. The Stephenz Group Inc.*, No. 3-18-cv-05061 (N.D. Cal.) (terminated Feb. 6, 2019)

65. *Express Mobile, Inc. v. Add2Net, Inc.*, No. 2-19-cv-05091 (C.D. Cal.) (terminated Sept. 30, 2019)

66. *Express Mobile, Inc. v. Evolve Media, LLC*, No. 2-19-cv-05093 (C.D. Cal.) (terminated Mar. 16, 2020)

67. *Express Mobile, Inc. v. InMotion Hosting, Inc.*, No. 2-19-cv-05097

(C.D. Cal.) (terminated May 13, 2020)

68. *Express Mobile, Inc. v. Netlancers Inc.*, No. 2-19-cv-05102 (C.D. Cal.) (terminated Dec. 16, 2019)

69. *Express Mobile, Inc. v. Perficient, Inc.*, No. 2-19-cv-05105 (C.D. Cal.) (terminated June 13, 2019)

70. *Express Mobile, Inc. v. Sigma Infosolutions Inc.*, No. 2-19-cv-05106 (C.D. Cal.) (terminated June 13, 2019)

71. *Express Mobile, Inc. v. Possible Worldwide, LLC*, No. 2-19-cv-05110 (C.D. Cal.) (terminated Oct. 18, 2019)

72. *Express Mobile, Inc. v. Perficient, Inc.*, No. 8-19-cv-01175 (C.D. Cal.) (terminated Mar. 10, 2020)

73. *Express Mobile, Inc. v. Advantage AMP, Inc.*, No. 2-19-cv-05155 (C.D. Cal.) (terminated Jan. 24, 2020)

74. *Express Mobile, Inc. v. Contus Interactive, Inc.*, No. 3-19-cv-03350 (N.D. Cal.) (terminated Feb. 7, 2020)

75. *Express Mobile, Inc. v. e-Zest Solutions, Inc.*, No. 3-19-cv-03351 (N.D. Cal.) (terminated Nov. 18, 2019)

76. *Express Mobile, Inc. v. MH Sub I, LLC d/b/a Officite*, No. 3-19-cv-03352 (N.D. Cal.) (terminated Aug. 13, 2020)

77. *Express Mobile, Inc. v. Phase2 Technology LLC*, No. 3-19-cv-03353 (N.D. Cal.) (terminated Feb. 4, 2020)

78. *Express Mobile, Inc. v. R2I Holdings, LLC*, No. 3-19-cv-03355 (N.D. Cal.) (terminated Jan. 22, 2020)

79. *Express Mobile, Inc. v. Rishabh Business Solutions, Inc.*, No. 3-19-cv-03356 (N.D. Cal.) (terminated Mar. 13, 2020)

80. *Express Mobile, Inc. v. Rauxa Direct, LLC*, No. 3-19-cv-03357 (N.D. Cal.) (terminated Jan. 17, 2020)

81. *Express Mobile, Inc. v. Web.Com Group, Inc.*, No. 1-19-cv-01936 (D. Del.) (terminated July 28, 2020)

C. Lead and Back-Up Counsel under §42.8(b)(3)

Petitioner provides the following designation of counsel.

LEAD COUNSEL	BACK-UP COUNSEL
Heidi L. Keefe (Reg. No. 40,673) hkeefe@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington, DC 20004 Tel: (650) 843-5001 Fax: (650) 849-7400	Phil Morton (Reg. No. 57,835) pmorton@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004 Tel: (202) 728-7055 Fax: (202) 842-7899
	Andrew Mace (Reg. No. 63,342) amace@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (650) 843-5808 Fax: (650) 849-7400
	Chih Yun (Steve) Wu (Reg. No. 69,082) swu@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (650) 843-5665 Fax: (650) 849-7400
	Dustin Knight (Reg. No. 76,239) dknight@cooley.com COOLEY LLP ATTN: Patent Group

LEAD COUNSEL	BACK-UP COUNSEL
	1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (703) 456-8024 Fax: (703) 456-8100
	Mark R. Weinstein (Admission *pro hac vice* to be requested) mweinstein@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (650) 843-5007 Fax: (650) 849-7400

D. Service Information

This Petition is being served by Federal Express to the attorney of record for the '168 patent, Lowenstein & Weatherwax LLP, 1880 Century Park East, Suite 815, Los Angeles, CA 90067. Petitioner consents to electronic service at the addresses provided above for lead and back-up counsel.

II. FEE PAYMENT

Petitioner requests review of 5 claims, with a $41,500 payment.

III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D)

A. Grounds for Standing

Petitioner certifies that the '168 patent is available for IPR and that Petitioner is not barred or otherwise estopped.

B. Identification of Challenge and Statement of Precise Relief Requested

Petitioner requests IPR institution based on:

Ground	Claims	Basis under §103
1	1-4 and 6	Reynolds (EX1003), Lemay Web (EX1004), Lemay Java (EX1005), Perry (EX1006), and Witkowski (EX1007)
2	1-4 and 6	Reynolds, Lemay Web, Lemay Java, Perry, Witkowski, and Coombs (EX1008)

Submitted with this Petition is the Declaration of Vijay K. Madisetti, Ph.D., a qualified technical expert. (EX1002, ¶¶1-13.)

C. Considerations Under §§ 314(a) and 325(d)

Petitioner respectfully requests that the Board decline to exercise its discretion to deny institution under §§ 314(a) or 325(d).

§314(a): The litigation is in its early stages—Patent Owner commenced litigation on September 1, 2020 (EX1012, p.93), and served Infringement

Contentions on February 5, 2021 (EX1014). Claim construction briefing is still ongoing and the district court has not issued a claim construction order. Petitioner served invalidity contentions in the district court, but to mitigate concerns regarding duplicative efforts, Petitioner will not pursue invalidity of the '168 patent in the litigation based on any instituted ground in this proceeding.

Trial is tentatively scheduled for over a year from now, on September 12, 2022. (EX1013.) Petitioner intends to move to stay the litigation pending IPR.

Petitioner is mindful of the limited resources of the Board. This Petition only challenges the five claims asserted in the litigation and sets forth a streamlined and straightforward challenge.

§325(d): None of the references cited in the proposed grounds listed in Part **III.B** was cited or considered during prosecution, yet they disclose the very limitations the Examiner found to be missing from the prior. In allowing the claims, the Examiner stated that the prior art did not disclose "producing a database of the constituent objects of a plurality of webpages including styles that include transformations and timelines for those objects in the manner claimed such that the web sites can be created from the database." (EX1011, p.140.) As explained in **Part VI**, Petitioner cites Reynolds (EX1003) in combination with Lemay Web (EX1004) and Lemay Java (EX1005), which disclose and render obvious the claimed "produce a database" and "style that includes values defining transformations and time lines"

limitations, as well as the other limitations of the challenged claims. Additionally, the references relied on the proposed grounds in this Petition are different from the ones relied upon in the *ex parte* reexamination filed by third party Unified Patents LLC. (EX1022, pp.5-6 (citing U.S. Patent Nos. 6,415,316, 6,289,362, 6,178,432, 6,230,185, 6,292,827, 6,433,784, and publications by Mauer ("Course Development Environment for Hyperware"), Li ("Synchronization in WebSmart Multimedia Authoring System"), NetObjects ("NetObjects Fusion Version 2.0 User Guide")).)

IV. PATENT OVERVIEW

A. Level of Ordinary Skill

A person of ordinary skill in the art as of December 1999 would have possessed at least a bachelor's degree in software engineering, computer science, computer engineering, or electrical engineering with at least two years of experience in web-based software application development, including experience in developing software and systems for storing, retrieving, and transmitting information (such as text and images) over a computer network such as the Internet (or equivalent degree or experience). (EX1002, ¶17.) 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 in the fields listed above. Acquired as part of the person's basic computer education and/or experience, as the

'168 patent makes clear, a person of ordinary skill in the art would have had a working knowledge about various ways to implement websites and providing webpages to browsers upon request. (*Id.*)

B. Specification

The '168 patent is directed to "[a] method and apparatus for designing and building a web page." ('168, Abstract.) As the '168 patent explains, its web page editor includes a "browser based build engine including build tools" that "are operable to construct a single run time file and an associated database that describe, and when executed, produce the web page." (*Id.*) A completed web page is stored into a database, which is uploaded to a user's web site along with an "HTML Shell File" and a "runtime engine." ('168, 2:38-48, 5:47-53.) Stored in the database are the objects of the web pages, including settings on "transformation[s]" and "time line[s]" for the web page objects. ('168, 35:19-30 (describing "transformation settings and technology" and stating that "[a] transformation is defined as the changing of an object from one state to another based on a timer control, subject to user settings"), 36:20-35 (describing "text button and image time lines and technology" and stating that "[a] time line is an independent asynchronous process that defines the existence of a given text button or image object.").) Exemplary object transformations include "the instantaneous drawing of one object state while erasing the previous object state" and "the gradual fading out of the previous object

state, while, simultaneously, fading into the next object state." ('168, 35:25-30.) The specification also describes various "settings" for transformations of "text" and "image" objects of the web pages, such as "[w]hich image states should the transformation be from and into" and "[t]he delay before the next transformation." ('168, 35:48-67.) With respect to "time lines," the specification states that "an object time line can be created as an instance of a class, which has a threadable interface" and "its own data structures, which define the animations, and transitions associated with the time line definition." ('168, 36:26-30.) According to the specification, a "class" refers to a Java class that is compiled and "uploaded to the user's web site" along with the database and run time engine. ('168, 44:3-30 (explaining that "all necessary JAVA class files," the "customized run time engine file" and the "database file" are compressed into "JAR and CAB files" and "uploaded to the user's web site," which "completes the run generation processes").)

When viewing the web site, the browser calls the runtime engine to read the database and "draw" a web page, ('168, 5:53-58), which includes executing instances of Java classes such as an object time line ('168, 22:31-35). Figure 2 below illustrates the process of accepting user input, creating a database and other "run time files," and uploading those files to a website.

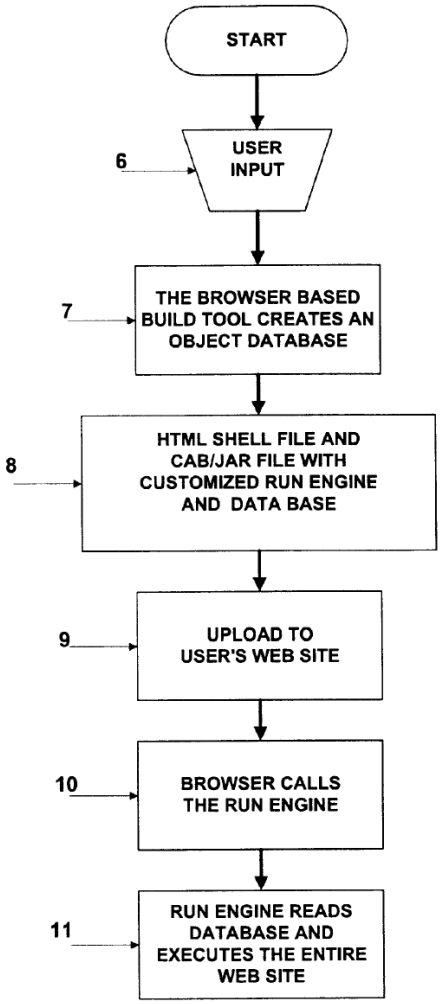


Fig. 2

('168, Fig. 2.)

V. CLAIM CONSTRUCTION

Various terms in the challenged claims have been either previously construed by district courts or constructions have been proposed by Patent Owner or various defendants, as summarized in the below table. Petitioner has listed these constructions below in the interests of complete disclosure, but does not believe express claim construction is necessary at this time. As explained in **Part VI** below,

the prior art discloses and renders obvious these limitations even under the narrowest versions of the constructions set forth below. Because Petitioner has shown that the challenged claims are unpatentable under any of the proposed constructions, or any reasonable construction, express construction at this time is unnecessary.

Term	Construction(s)
runtime engine (cl. 1)	• file that is executed at runtime that utilizes information from the database and generates commands to display a web page or website (EX1021, p.16 (proposed by Patent Owner)) • file that is executed at runtime that facilitates the retrieval of information from the database and generates commands to display a web page or website (EX1017) • file that is executed at runtime that reads information from the database and generates commands to display a web page or website (EX1016, pp.7-8; EX1020, p.1; EX1021, p.17)
database (cl. 1)	• an electronic information storage system offering data storage and retrieval (EX1016, p.9) • an electronic information storage system offering data storage and retrieval and that stores information on a record-by-record basis, each record divided into one or more fields (EX1018, p.19; EX1020, p.2)
multidimensional array (cl. 1)	• a uniquely identifiable indexed set of related elements, wherein each element is addressed by a set of two or more indices, each index corresponding to a dimension of the array (EX1018, p.17; EX1020, p.1) • an indexed set of related elements, wherein each element is addressed by a set of two or more

Term	Construction(s)
	indices (EX1018, p.16 (proposed by Patent Owner))
settings (cl. 4)	• attributes of an object available for selection (EX1020, p.1)
transformations (cl. 1)	• the changing of an object from one state to another based on a timer control, subject to user settings (EX1018, p.7; EX1016, p.13; EX1021, p.10)
time lines (cls. 1, 3)	• a sequence of changes that define the attributes of a text button or image object as the changes occur (EX1016, p.13) • independent process that defines one or more values for an object, including its appearance, animation, speed, and/or resolution over time (EX1016, p.13 (proposed by Patent Owner)) • independent asynchronous process that defines the values for an object, including its appearance, animation, speed, and/or resolution at one or more defined times (EX1025, p.5 (proposed by Patent Owner)
objects (cls. 1-4)	• variable comprising data and routines treated as a discrete entity (EX1027, p.4 (proposed by Petitioner))
style (cls. 1-4)	• attribute that defines the visual appearance of an object (EX1027, p.5 (proposed by Petitioner))
button (cls. 1-3)	• a graphic object that, when activated through user interaction, performs a specified function (EX1016, p.11)
child/child object/child button (cls. 2, 3)	• an element with one or more properties that depends on another element (EX1026, pp.35-36 (proposed by Patent Owner))

Term	Construction(s)
	• element/object/button that is spawned by another element/object/button (i.e., the parent element/object/button) (EX1026, pp.35-36 (proposed by defendant GoDaddy))

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE.

A. Overview of Grounds

The challenged claims are directed to a combination of three well-known features in the prior art: (1) the use of a "database" to store and later retrieve "objects" of a web site; (2) the use of a "runtime engine" to "generate" the web site from the database; and (3) the inclusion of user-created animated "button" or "image" objects on the web pages of the web site.

With respect to features (1) and (2), this Petition relies primarily on **Reynolds** (EX1003) entitled "Interactive Web Book System," which describes a system for users to create websites in the form of interactive Web books or "ibooks." (Reynolds, Abstract; *see also id*., 1:5-8.) The user-created content of Reynolds' ibook web pages is stored in ibook Web pages database **76** (*id.*, 7:35-39), and later retrieved by a navigation tool (i.e., a "runtime engine") at a viewer's browser to build ibook web pages and thus generate the ibook web site. (*Id.*, 11:24-39.)

With respect to feature (3), this petition relies primarily on **Lemay Web** (EX1004) entitled "Teach Yourself Web Publishing with HTML 3.2 in a Week," which describes how a user can add animated image (and button) objects to web

pages in the form of Java applets. (Lemay Web, pp.54-56.) In particular, Lemay Web explains that the makers of Java provided a pre-built Java applet named "Animator.class" that a user could simply add to a web page by specifying the values for various parameters that would define how the visual appearance of the image or button object transforms over time. (*Id.*)

Grounds 1 and 2 rely on other references that are cited primarily for basic web and database technologies that would have fallen within basic knowledge of ordinarily skilled artisans and thus are not described in detail in Reynolds and Lemay Web. *See Spectra-Physics, Inc. v. Coherent, Inc.*, 827 F.2d 1524, 1534 (Fed. Cir. 1987) ("A patent need not teach, and preferably omits, what is well known in the art."). The Petition cites **Perry** (EX1006) for the claimed recitation of a "multidimensional array" in claim 1, which Perry confirms is a known and elementary data structure. (Perry, p.456.). The petition further cites to **Witkowski** (EX1007) in combination with Perry, assigned to Oracle, for rudimentary database techniques such as the implementation of a relational database table organized as a multidimensional array based on rows and columns.

To specifically address narrow constructions that have been proposed in prior litigations and identified above in **Part V**, the petition cites to **Lemay Java** (EX1005) for its explanation of the workings of Java applets (including their ability to operate independently through multithreading), and in Ground 2 adds **Coombs**

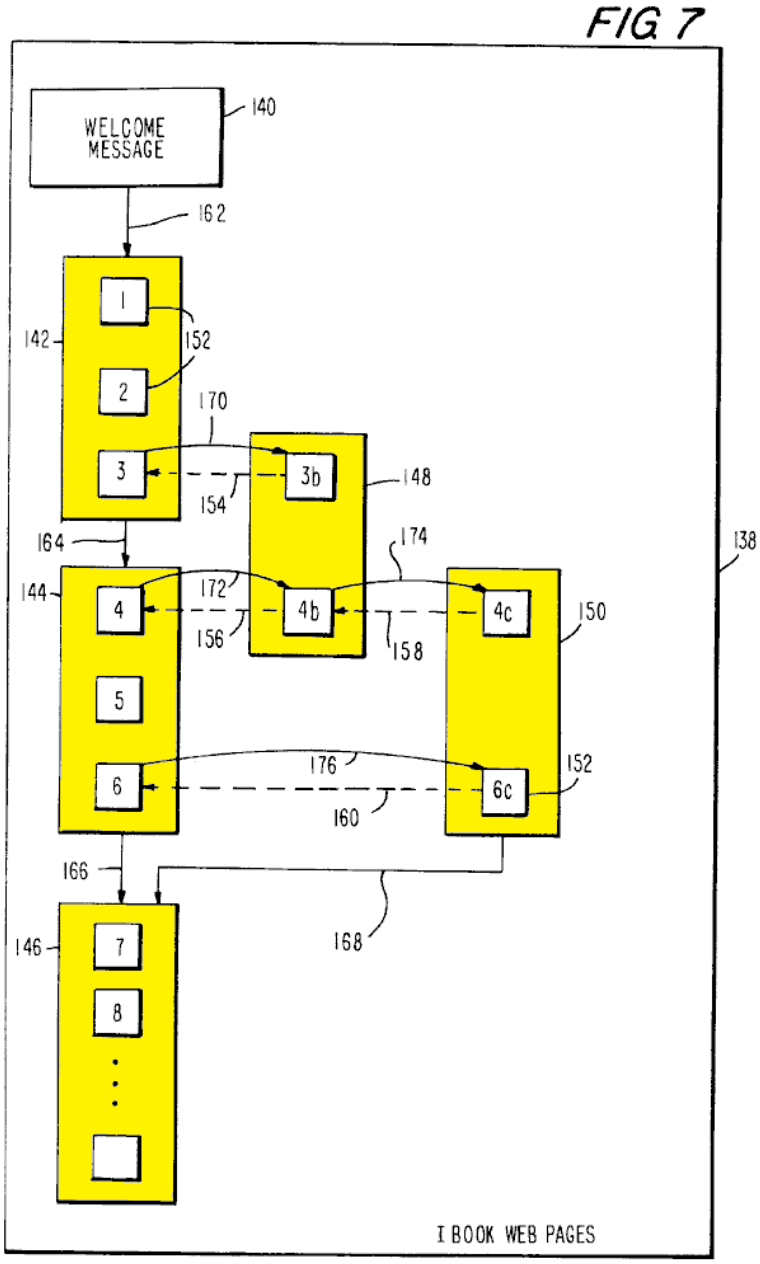
(EX1008) for its description of basic steps involved in setting up a web site, such as the registration of a domain name.

Each reference cited in this Petition qualifies as prior art because each reference was either filed (in the case of patent references) or published long before the claimed December 1999 priority date of the '168 patent. (EX1024, ¶¶40-69.)

B. Ground 1: Obviousness of Claims 1-4 and 6 Over Reynolds, Lemay Web, Lemay Java, Perry, and Witkowski

1. Claim 1

The preamble of claim 1 recites, "[**a**] **system for assembling a website comprising**…." To the extent the preamble is limiting, it is disclosed by Reynolds. The "**website**" in Reynolds corresponds to an "interactive Web book," or "ibook," containing a series of ibook Web pages. (Reynolds, Abstract; *see also id.*, 1:25-27 ("This and other objects of the invention are accomplished in accordance with the principles of the present invention by providing an interactive Web book ('ibook') system."), 1:30-31 ("Material in an ibook is preferably organized in the form of Web pages."); *see also* EX1002, ¶65).) Figure 7 depicts an exemplary ibook with multiple ibook web pages:



(Reynolds, Fig. 7 (highlighting added).) This exemplary ibook includes five web pages **142**, **144**, **146**, **148**, **150** (shown in yellow highlighting). (Reynolds, 7:57-67.) The ibook system in Reynolds thus "**assembl**[es]" a website by assembling the ibook and its web pages. As shown in Figure 7, the web pages of the ibook are interconnected via a series of hyperlinks, *e.g.*, sequence link **164** links web page **142**

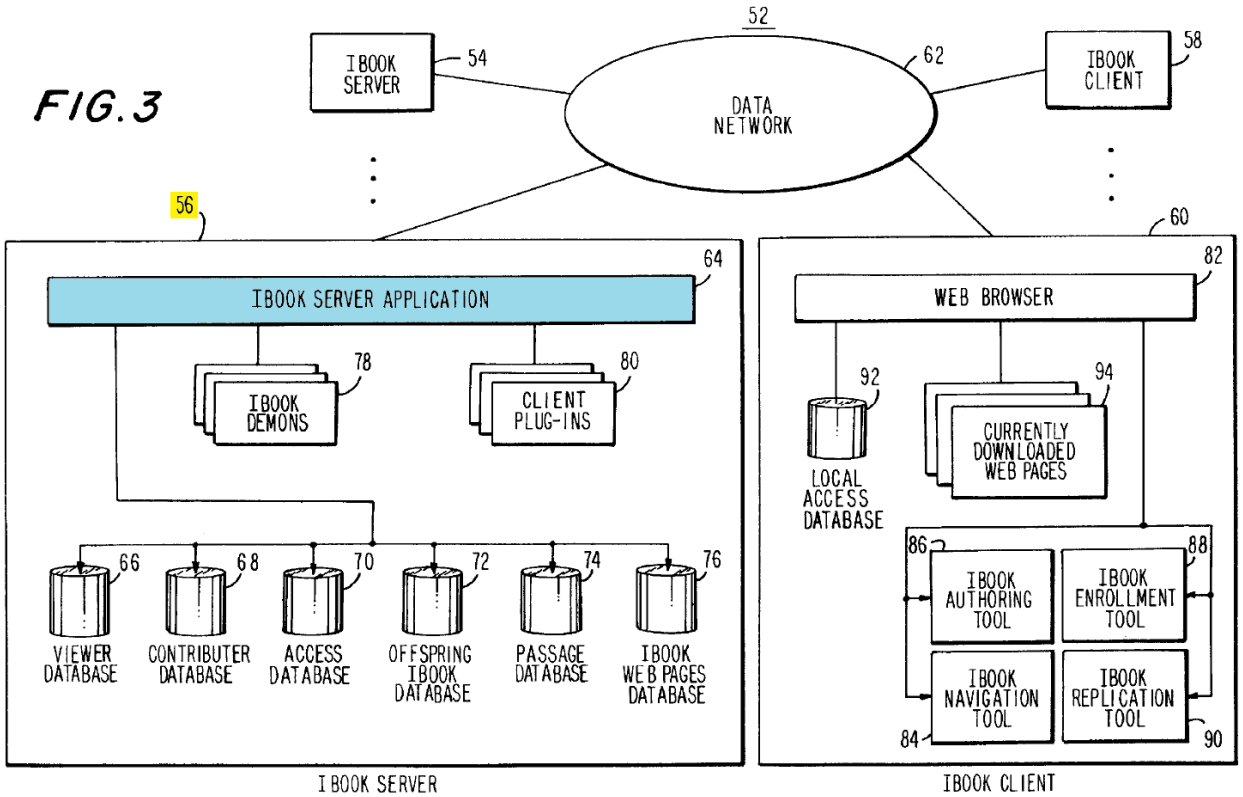
to **144**, sequence link **166** links web page **146** to **148**, and so forth. (Reynolds, 8:53-60; *see also* EX1002, ¶65.)

Each web page of an ibook includes one or more "passages" **152** (depicted as numbered squares within the yellow highlighted web pages), which provide the content of ibook web pages. (Reynolds, 7:66-67, 1:34-38 ("Within each Web page, material may be organized in passages. Passages can be of any suitable size, such as a paragraph for textual material, or a video or audio clip of a certain length for multimedia material.").)[1] As further explained below, "passages" are downloaded from a server and placed onto the web pages by software running on the web browser on the client device. (Reynolds, 11:24-40.) For these reasons, and further explained below, Reynolds discloses a system for "**assembling a website**."

(a) "a server comprising a build engine configured to:" (Claim 1[a])

The "**server**" in Reynolds corresponds to ibook server **56**, which in turn comprises a "**build engine**" in the form of an ibook server application **64**. (EX1002, ¶67.) Both the server and the build engine are depicted in Reynolds Figure 3:

[1] Unless otherwise noted, all underlining, boldface or italics in quotes from prior art references has been added by Petitioner for emphasis.



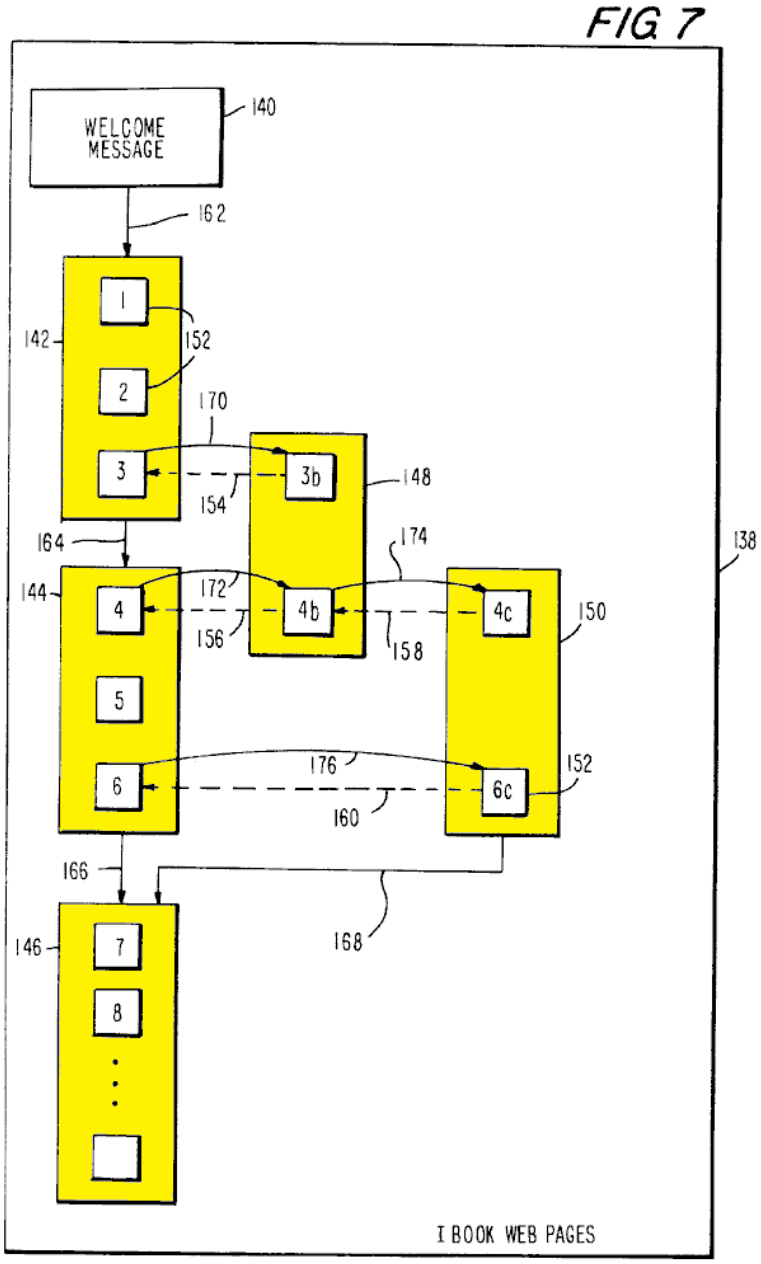
(Reynolds, Fig. 3 (highlighting added).) As depicted, ibook server **56** includes an ibook server application **64** which "directs the overall operation of ibook server **56**." (Reynolds, 5:18-20.) As further explained below, server application **64** ("**build engine**") is configured to perform the functions recited in claim 1.

(b) "accept user input to create a web site, the web site comprising a plurality of web pages, each web page comprising a plurality of objects," (Claim 1[b])

Reynolds' build engine is configured to accept user input to create a web site. For ease of explanation, Petitioner will first explain how Reynolds discloses the "**web site**," "**web page**," and "**objects**" aspects of claim 1**[b]**, before addressing how Reynolds discloses "**accept**[ing] **user input to create a web site**."

As noted, an ibook website corresponds to the claimed "**web site**." An ibook web site "**compris**[es] **a plurality of web pages**," *i.e.* ibook web pages. (Reynolds, *e.g.*, 1:29-30 ("Material in an ibook is preferably organized in the form of Web pages."), 7:36-37.) Each web page comprises "**a plurality of objects**" in the form of passages **152** containing content for ibook web pages. (Reynolds, 1:34-38 ("Within each Web page, material may be organized in passages. Passages can be of any suitable size, such as a paragraph for textual material, or a video or audio clip of a certain length for multimedia material."); EX1002, ¶68.) Passages **152** can include not only text but also "graphic images, animation sequences, portions of a sound track, still video images, or video clips. If an ibook contains executable material, passages **152** can be modules of code." (Reynolds, 8:4-11; *see also id.*, 9:52-54 ("When passages **152** are executable modules, navigation tool **84** presents the passages by downloading and executing the passages.").)

As previously discussed, Figure 7 depicts an exemplary ibook with a "**plurality of web pages**," *i.e.* ibook web pages, "**each web page comprising a plurality of objects**," *i.e.* passages within each web page:



(Reynolds, Fig. 7 (highlighting added).) As previously described, the exemplary ibook in Figure 7 includes five web pages **142**, **144**, **146**, **148**, **150** (shown in yellow highlighting), each including one or more "passages" **152** (shown as numbered white squares in the yellow highlighted web pages), which provide the content for ibook web pages. (Reynolds, 7:57-67, 7:66-67, 1:34-38; EX1002, ¶69.)

The "**plurality of web pages**" for claim 1**[b]** corresponds to at least two ibook web pages containing passages **152**, which can be any two of the web pages shown in yellow highlighting from Figure 7. (EX1002, ¶70.) Because claim 1**[b]** only requires a "**web site *comprising* a plurality of web pages**," the web site can include additional web pages beyond the claimed "plurality of web pages." *See, e.g.*, *Cordis Corp. v. Medtronic Ave, Inc.*, 511 F.3d 1157, 1174-75 (Fed. Cir. 2008) ("comprising" does not exclude additional elements beyond those recited in the claim); *Apple Inc. v. Samsung Elecs., Ltd.*, 695 F.3d 1370, 1378 (Fed. Cir. 2012) ("plurality" means "at least two"). Accordingly, identifying at least two ibook web pages satisfies the required "plurality of web pages," even if the ibook web site may contain additional pages. (EX1002, ¶70.)

As noted above, the claimed "**objects**" correspond to Reynolds' passages. (EX1002, ¶71.) As identified in **Part V**, one proposed construction for "object" was a "*variable comprising data and routines treated as a discrete entity*." Reynolds' passages are "objects" under this proposed construction. Passages comprise a "*variable*" because they are not fixed information and, as explained below, can vary and contain data selected by the user through authoring tool **86**. (*Id.*) As Reynolds also confirms, passages can contain "*data and routines*." (*Id.*) The term "routine" in computer science generally refers to a sequence of software code or instructions that can be invoked to perform a particular function. (*Id.*) Passages in Reynolds

contain "data and routines" because, as noted, they can contain data and executable software or code modules. (Reynolds, *e.g.*, 8:10-11, 9:52-54.) Thus, the executable software modules described in Reynolds would at least include software routines and data used to carry out their functions. (EX1002, ¶71.) Moreover, as explained for claim 1**[c]** below, under the combination of Reynolds with Lemay Web and Lemay Java, ibook passages can include Java applets containing code and image data for displaying an animated image object, which would likewise contain data and routines. Finally, each passage in Reynolds is "*treated as a discrete entity*" because, as explained below, each passage is accessed and retrieved separately from the server when viewing the web page. (Reynolds, 11:24-40; EX1002, ¶71.) Reynolds also discloses the ability to keep track of viewer access on a passage-by-passage basis, further confirming that passages are treated as discrete entities. (Reynolds, 11:7-11, 11:43-55; EX1002, ¶71.)

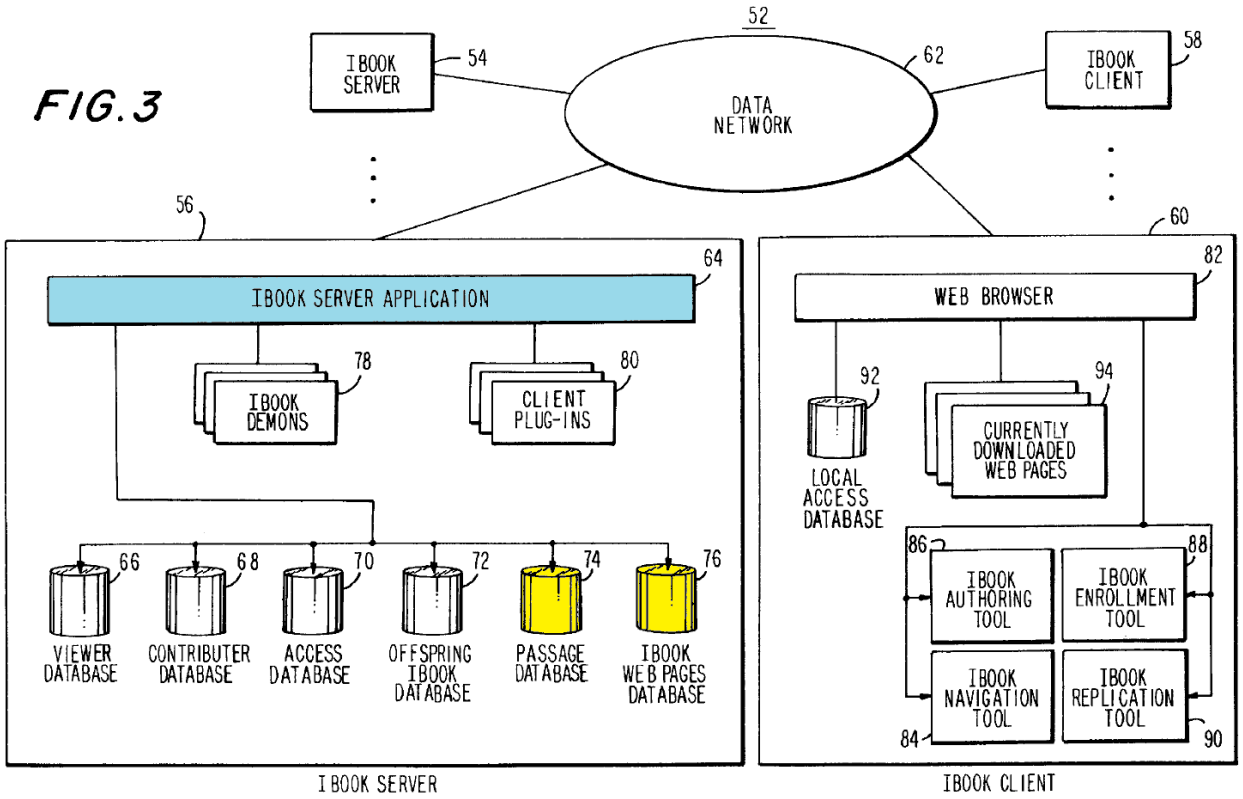
For the reasons above, therefore, Reynolds discloses and renders obvious "**a web site, the web site comprising a plurality of web pages, each web page comprising a plurality of objects**," as claimed.

Now turning back to the other aspects of claim 1**[b]**, Reynolds discloses that web server application **64** (the "**build engine**") is also configured to "**<u>accept user input</u> to create a web site**," as claimed. (EX1002, ¶¶73-75.) Reynolds discloses a set of tools including authoring tool **86**, which is a plug-in software program

downloaded from the server and executed on the client device. (Reynolds, 7:2-5 ("To create or modify material in an ibook, a contributor uses ibook authoring tool **86** (FIG.3). As shown in FIG. 6, ibook authoring tool **86** includes derivation tool **132**, linking tool **134**, and editing tool **136**."), 5:28-37 ("Client plug-ins **80** are downloadable executable program modules that can be run on client **60** in conjunction with Web browser **82** to add additional capabilities to Web browser **82**… When client plug-ins **80** are downloaded onto client **60**, they form tools **84**, **86**, **88**, and **90**.").) Authoring tool **86** accepts user input to create the passages of an ibook, and "[t]he functions provided by authoring tool **86** may differ depending on the content of a particular ibook passage." (Reynolds, *e.g.*, 7:23-24.) Authoring tool **86** includes editing tool **136** which "preferably supports standard word processing functions for entering text and source code and supports graphics, animation, audio, and video editing functions for creating and modifying multimedia material." (Reynolds, 7:25-28.)

Although authoring tool **86** that accepts user input runs on client computer **60**, Reynolds makes clear that ibook server application **64** (the "**build engine**") on the server also "**accept**[s] **user input**" for the passages used to create an ibook. (EX1002, ¶74.) This is because the user input received through authoring tool **86** is communicated to ibook server application **64**. Reynolds explains that "during the creation of passages **152** authoring tool **86** **and** ibook server application **64** cause

information to be stored in passage database **74** that identifies which Web page **138** contains each passage **152**." (Reynolds, 9:27-30.) Figure 3 depicts at a high level the storage of ibook content at the server **56**:



(Reynolds, Fig. 3 (highlighting added).) Reynolds explains that ibook server application **64** shown in Figure 3 ("**build engine**") controls various databases, including passage database **74** and ibook web pages database **76** which store ibook information. (Reynolds, *e.g.*, 5:21-23 ("In particular, ibook server application **64** controls databases **66**, **68**, **70**, **72**, **74**, and **76**, in which various ibook information is stored.").) "The content of the ibook Web pages is preferably stored in ibook Web pages database **76** (FIG. 3)." (Reynolds, 7:37-39.)

It therefore would have been obvious to an ordinarily skilled artisan that ibook server application **64** accepts the user input which was provided via authoring tool **86** to create ibook passages—otherwise ibook server application **64** could not store ibook information in its databases to reflect user selections made using authoring tool **86**. (EX1002, ¶75.) Thus, Reynolds discloses and renders obvious a "**build engine**" that is configured to "**accept user input to create a web site**."

(c) "accept user input to associate a style with objects of the plurality of web pages, wherein each web page comprises at least one button object or at least one image object, and wherein the at least one button object or at least one image object is associated with a style that includes values defining transformations and time lines for the at least one button object or at least one image object;" (Claim 1[c])

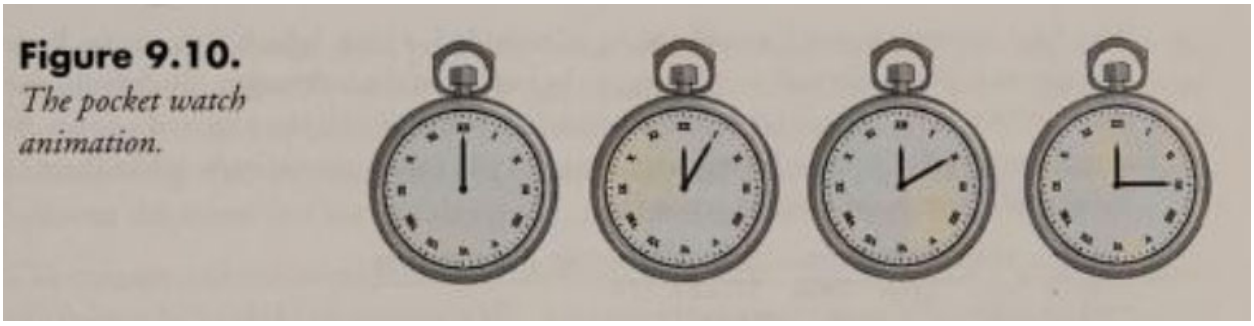
This limitation is rendered obvious by Reynolds in combination with Laura Lemay, *Teach Yourself Web Publishing with HTML 3.2 in a Week* (1996) (**Ex. 1004**) ("**Lemay Web**"). Reynolds and Lemay Web render obvious the ability of the build engine to accept user input to associate a **style** with objects on an ibook web page, such as the style associated with a Java animation object ("**at least one button object or at least one image object**"). (EX1002, ¶¶76-90.)

As previously described, Reynolds discloses that passages **152** of ibook web pages ("**objects of the plurality of web pages**") can include not just text but also graphic images, animation sequences, and executable code modules. (Reynolds, 8:4-11, 9:52-54.) As noted, Reynolds also discloses the ability to **accept user input**

provided by the user through authoring tool **86**. "The functions provided by authoring tool **86** may differ depending on the content of a particular ibook passage"—for example, "editing tool **136** [part of authoring tool **86**] preferably supports standard word processing functions for entering text and source code and supports graphics, animation, audio, and video editing functions for creating and modifying multimedia material." (Reynolds, 7:23-24, 7:25-28.)

Reynolds thus contemplates that authoring tool **86** will have functionality for the user to place content on an ibook web page, but Reynolds does not further detail the specific user inputs provided through authoring tool **86**. (EX1002, ¶78.) Such details are provided by Lemay Web, which describes how to add "**at least one button object or at least one image object**" to a web page and how to specify a **style** for the object as claimed. (EX1002, ¶¶78-84.)

In particular, Lemay Web provides an exemplary, simple Java applet that can be placed on a web page to display an animated image of a pocket watch:



(Lemay Web, Fig. 9.10, p.255.) Figure 9.10 above illustrates the first four frames of an animated pocket watch image, with the minute hand pointing to different

positions on the dial in each frame—in these four examples at 12:00, 12:05, 12:10, and 12:15. Lemay's example includes 12 total image files for the pocket watch (named T1.gif, T2.gif, and so on), each corresponding to 1 of the 12 different positions of the second hand. (*Id.*, p.255.)

Adding this image animation to a web page would have been exceedingly straightforward. (EX1002, ¶80.) As Lemay Web explains, the creators of Java (Sun Microsystems) provided a pre-built Java applet named "Animator.class" enabling a web page creator to specify a series of images to be displayed sequentially on a web page. (Lemay Web, pp.254-55.) Inserting the animated image into a web page would have been as simple as inserting a few lines of HTML:

```
<APPLET CODE="Animator.class" WIDTH=100 HEIGHT=100>
<PARAM NAME="STARTIMAGE" VALUE="1">
<PARAM NAME="ENDIMAGE" VALUE="12">
<PARAM NAME="PAUSE" VALUE="1000">
<PARAM NAME="REPEAT" VALUE="TRUE">
</APPLET>
```

(Lemay Web, p.256.) The above HTML code includes <APPLET> and <PARAM> tags, standard tags for inserting a Java applet onto a web page and would have been well-known to any ordinarily skilled artisan. (Lemay Web, p.255; EX1002, ¶80.) As Lemay Web explains, the first <APPLET> tag specifies the name of the Java applet (*i.e.* "Animator.class"), and in the above example, the WIDTH and HEIGHT for the applet on the page (*i.e.* 100-by-100 pixels). (Lemay Web, p.255.)

The HTML code continues with four <PARAM> tags ("STARTIMAGE," "ENDIMAGE," "PAUSE," and "REPEAT"), each with a NAME and VALUE attribute. (Lemay Web, p.256.) "STARTIMAGE" and "ENDIMAGE" indicate that the animation includes 12 images (from 1-12). "PAUSE" specifies how long (in milliseconds) the applet will pause between successive images in the animation. (Lemay Web, p.256.) As Lemay explains, "[y]ou can experiment with the pause between frames until you get an animation you like (here I picked 1000 milliseconds, or an even second)…." (Lemay Web, p.256.) Finally, the example sets the "REPEAT" parameter to "TRUE," which instructs the applet to repeatedly loop the animation. (Lemay Web, p.256.)

Lemay Web clarifies that the exemplary <APPLET> and <PARAM> tags provide "only a couple of the Animator applet's parameters," and other parameters are available to web site creators. (Lemay Web, p.257.) For example, additional parameters "let you… add a background to the animation or a soundtrack, and control the order that frames are displayed." (Lemay Web, p.257.)

Thus, Lemay Web discloses "**objects of the plurality of web pages**" in the form of image animations that include both graphical images and executable code (*i.e.* the Java animation applet) to display the images in sequence. (EX1002, ¶83.) And as explained above, Lemay Web further describes how to "**accept user input to associate a style**" with such objects, including by entering HTML code with

respect to the many different examples of attributes which define the objects' visual appearance (e.g., width and height on the page, the sequence of the images, a "PAUSE" value indicating how long each image is displayed before advancing to the next image, and a "REPEAT" value indicating whether or not the image animation will occur once or cycle in a loop). (*Id.*)

The proposed combination thus renders obvious "**accept**[ing] **user input to associate a style with objects of the plurality of web pages**," because authoring tool **86** of Reynolds under the proposed combination would be adapted to accept user input to specify attributes that define the visual appearance of an object, such as the WIDTH, HEIGHT, STARTIMAGE, ENDIMAGE, PAUSE, or REPEAT attributes associated with the animated image object in Lemay Web. The website creator can input any one or more of these attributes to associate a style with objects of the plurality of web pages as described above.

Lemay Web further discloses that "**each web page comprises at least one <u>button</u> object or at least one <u>image</u> object**," as claimed. In fact, the animated image object qualifies as both. (EX1002, ¶84.) It qualifies as an "**image object**" because it includes a number of graphical images for display on the web page. It also qualifies as a "**button object**" because "clicking on the animation will start and stop it." (Lemay Web, p.256.) Stated differently, the animated image is a "**button**

object" because it starts and stops the animation when activated by the viewing user's interaction through a mouse click. (EX1002, ¶84.)

Regardless of whether an animated image is viewed as an image object, a button object, or both, it is "**associated with a style that includes values defining transformations and time lines for the at least one button object or at least one image object**," as claimed. First, the animated image includes values defining "**transformations**" because the <PARAM> tags defining the above-specified style define how the image object changes from one state to the next. (EX1002, ¶85.) The "STARTIMAGE" and "ENDIMAGE" tags, as mentioned, specify the images that define where the animation begins and ends—e.g., how the appearance of the exemplary pocket watch transforms as the animation advances. (*Id*.) In addition, the animation is subject to a user-selectable timer control because, as mentioned, the "PAUSE" tag specifies the amount of time between one image and the next. (*Id*.)

Second and for similar reasons, the <PARAM> tags that can be used to define the style likewise define "**time lines**" for the image or button object. (EX1002, ¶86.) As identified in **Part V**, three different constructions for the term "time lines" have been provided in prior litigations, the simpler two being "*a sequence of changes that define the attributes of a text button or image object as the changes occur*," and "*independent process that defines one or more values for an object, including its appearance, animation, speed, and/or resolution over time*." Both definitions are

readily satisfied by the above-mentioned styles because, as explained, they specify attributes that define the appearance of the image and how it changes at defined times (*e.g.*, the resolution of the pocket watch image, how and how often the pocket watch image changes, etc.). (*Id.*) The above-mentioned styles also define an "*independent process*" because the parameters are entirely responsible for specifying the sequence of images and speed of the animation for the object on the web page. (*Id.*) In other words, the parameters define how the pocket watch visually transforms and do not depend on other objects on the web page. (*Id.*)

The third definition that has been previously proposed for "**time lines**" imposes a further requirement of an "*independent asynchronous process that defines the values for an object*," which is disclosed and rendered obvious by multithreading features built-in to Java itself. A skilled artisan would have understood an "asynchronous process" as a process that can occur at the same time as (or concurrently with) other processes. (EX1002, ¶87.) In contrast, a "synchronous" process "blocks" or otherwise prevents other processes from executing until the synchronous process completes. (*Id.*) The proposed requirement of an "*independent asynchronous process*" would have been met by Java's built-in multithreading features that enable applets to be executed concurrently, as disclosed in Laura Lemay, *Teach Yourself Java in 21 Days* (1996) (**Ex. 1005**) ("**Lemay Java**"). (EX1002, ¶¶87-90.)

Lemay Java explains that "[m]ultithreading, as in Java, enables several different execution threads to run at the same time inside the same program, in parallel, without interfering with each other." (Lemay Java, p.178.) Thus, "[u]sing threads, you can have lots of applets running at once on the same page," *i.e.*, they can run independently. (*Id.*) As Lemay further teaches, "[e]ven if you don't have lots of applets, using threads in your applets is good Java programming practice. The general rule of thumb for well-behaved applets: whenever you have any bit of processing that is likely to continue for a long time (<u>such as an animation loop</u>, or a bit of code that takes a long time to execute), put it in a thread." (*Id.*)

As taught by Lemay Java, it would have been obvious to implement the animated Java image object disclosed in Lemay Web as its own process thread. (EX1002, ¶90.) This combination would have resulted in implementation of the animated image object in its own process or thread, thus making it both "independent" and "asynchronous" of other processes and satisfying even the most restrictive construction of "**time lines**" from prior litigations. (*Id.*) The motivation to combine would have been straightforward—Lemay Java teaches that the use of multithreading was a "general rule of thumb" in order to create "well-behaved" "animation loop[s]" and to enable multiple applets to run at the same time on the same page without interfering with each other. (Lemay Java, p.178.)

Manner, Rationale, and Motivation to Combine Reynolds with Lemay Web and Lemay Java: It would have been obvious to an ordinarily skilled artisan to combine Reynolds with Lemay Web and Lemay Java as proposed, predictably resulting in the modification of Reynolds' ibook creation system to allow the user to specify for each web page, during the process of creating an ibook passage, an animated image object based on the teachings of Lemay Web ("**wherein each web page comprises at least one button object or at least one image object**"). An exemplary implementation of the proposed combination would have enabled a user of authoring tool **86** in Reynolds to provide input defining an image object for each page by specifying the "Animator" Java applet taught by Lemay Web, and also providing user-defined HTML parameters to specify the object's visual appearance—including identifying the images and their resolution, sequence timing, and other visual characteristics as previously explained. (EX1002, ¶91.) As previously explained for claim 1**[b]**, such user input would have been accepted by ibook server application **64** ("**build engine**") and stored in its databases.

As an initial matter, Reynolds, Lemay Web, and Lemay Java are analogous references in the same field as the '168 patent, and are reasonably pertinent to a host of problems faced by the inventors in building their web site building system. (EX1002, ¶92.) The proposed combination would also have been straightforward to an ordinarily skilled artisan. Reynolds itself already discloses that passages **152** of

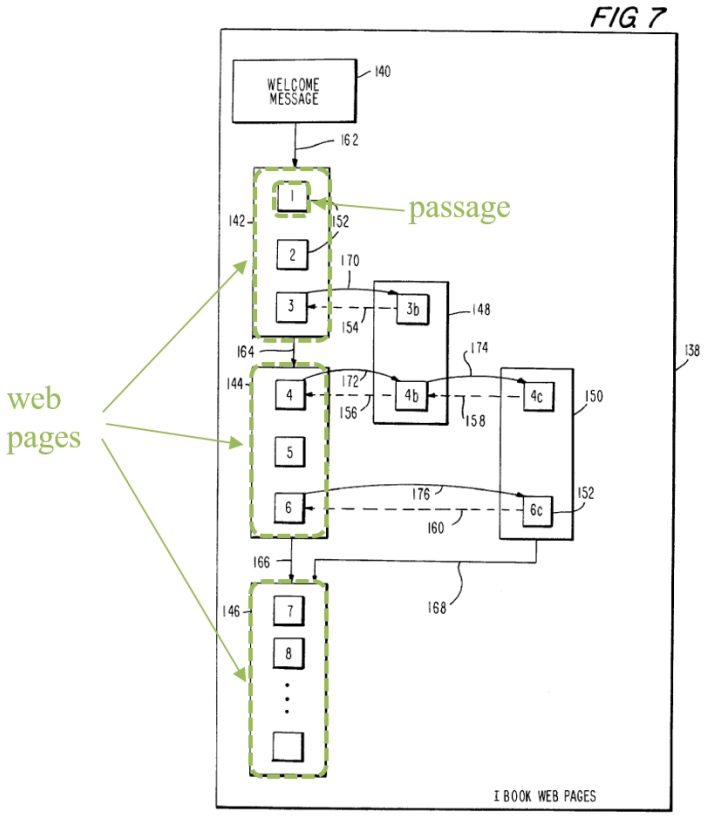
an ibook can include "graphic images," "animation sequences," and "executable material" including "modules of code." (Reynolds, 8:7-11; *see also id.*, 9:52-54 ("When passages **152** are executable modules, navigation tool **84** presents the passages by downloading and executing the passages.").) Moreover, Reynolds also confirms that passages can be implemented as Java programs. (Reynolds, 9:56-58 ("Each style is controlled by an ibook passage (e.g., a Java program) which implements that style.").) Reynolds thus clearly contemplates that a passage could include the Animator Java applet described in Lemay Web.

An ordinarily skilled artisan would have had many motivations to implement Reynolds' ibook system in this manner. (EX1002, ¶93.) Lemay Web provides an express motivation by explaining that the Animator Java applet is provided by the makers of Java as a pre-built applet. (Lemay Web, p.254.) As Lemay Web explains, "you can use pre-built Java applets on your pages to create animation effects without touching a line of Java code." (Lemay Web, p.254; EX1002, ¶93.) "All you have to do is download the Java applet to your system, include a few lines of HTML on your page, and everything works just fine…." (Lemay Web, p.254.) The ease with which the Animator Java applet could have been incorporated into ibook web pages thus provides a motivation to combine. In addition, an ordinarily skilled artisan would have been motivated to adapt the Reynolds ibook creation system to support the animated image objects of Lemay Web to enrich user experience, because

animated images may be more engaging or compelling than textual content. (EX1002, ¶93.) And as explained previously, a skilled artisan would have been motivated to harness the advantages of Java multithreading, resulting in an image object that is both "independent" and "asynchronous" of other processes, allowing applets to run without interference from other processes. (*Id*., ¶¶90, 93.)

(d) "and wherein each web page is defined entirely by each of the plurality of objects comprising that web page and the style associated with the object," (Claim 1[d])

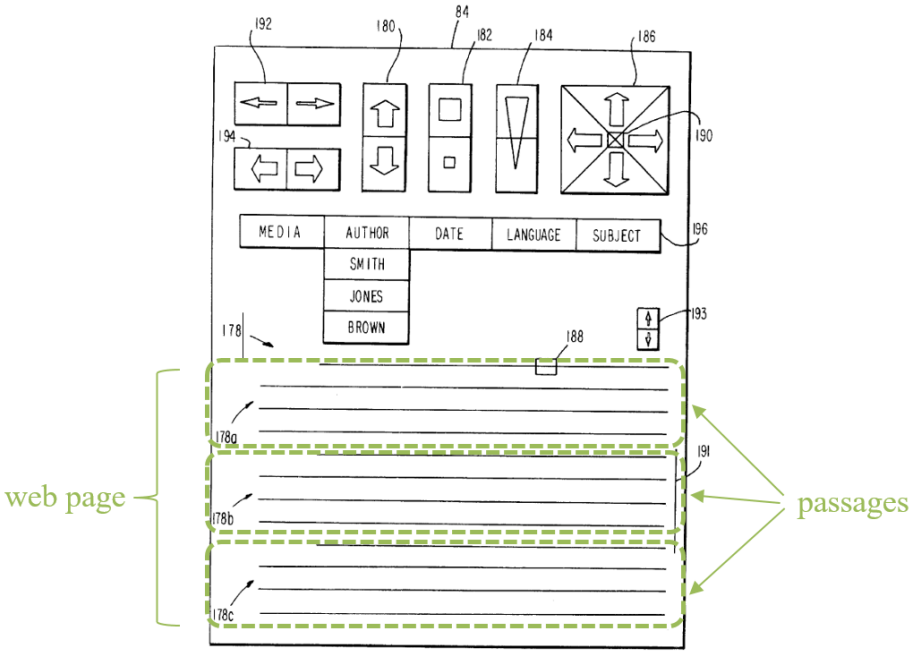
Reynolds discloses and renders obvious that each web page is defined entirely by each of its passages and the associated style. And, as previously explained, it would have been obvious that—under the combination of Reynolds with Lemay Web and Lemay Java—each passage would have included an animated image object. (Reynolds, 8:7-10 ("If an ibook contains multimedia material, passages **152** can be graphic images, <u>animation sequences</u>, portions of a sound track, still video images, or video clips.").) Figure 7 of Reynolds illustrates an ibook website with various web pages (**142**, **144**, **146**, **148**, and **150**), each web page in turn composed of passages **152** (which are retrieved from Web pages database **76**):



(Reynolds, Fig. 7 (annotation added); *see also id.*, 7:66-67 ("Web pages **142**, **144**, **146**, **148**, and **150** are preferably organized in the form of one or more passages **152**."), 7:37-39.)

As Reynolds makes clear, each web page in an ibook is defined **entirely** by passages **152** ("**each of the plurality of objects**") that make up that page, because each ibook web page is nothing more than a collection of passages. (EX1002, ¶95.) Figure 8 of Reynolds illustrates an exemplary ibook web page as displayed in a web browser using navigation tool **84**, with multiple passages **178** (**178**a, **178**b, and **178**c) displayed in accordance with the style associated with those passages:

FIG. 8

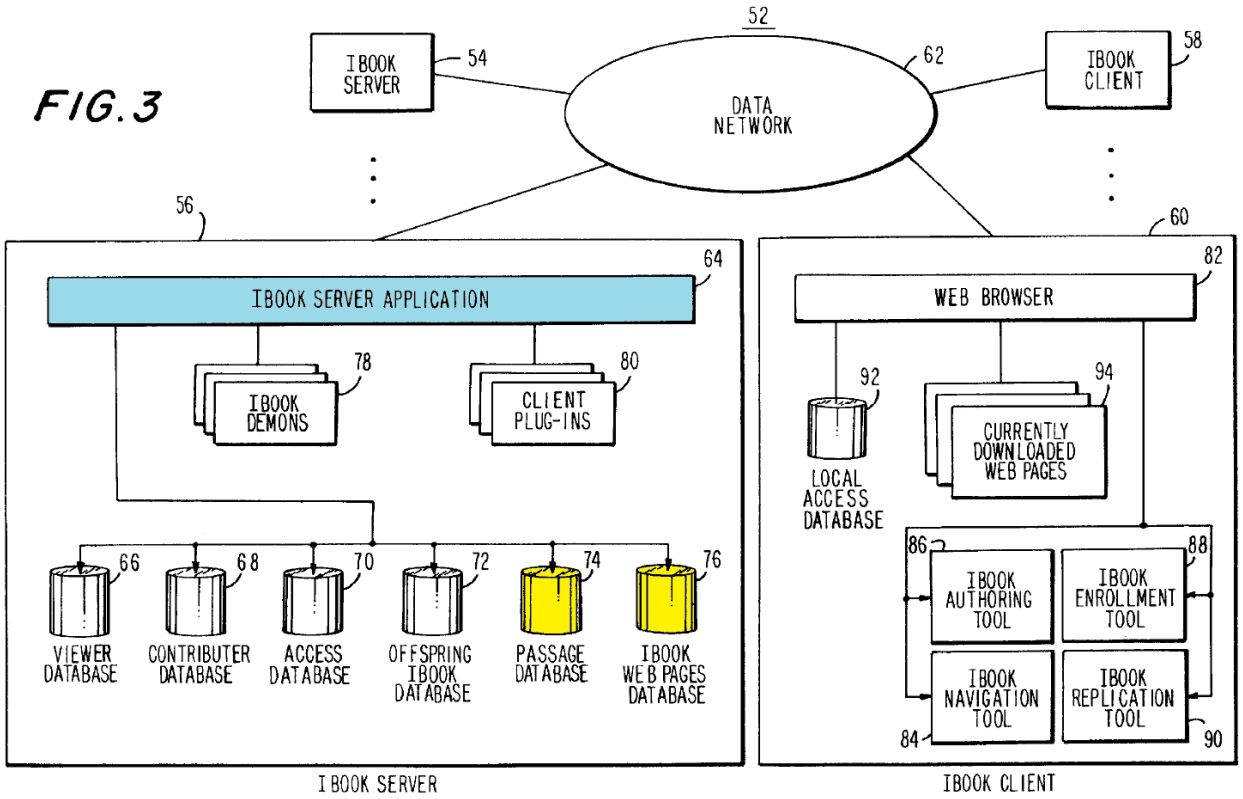


(Reynolds, Fig. 8 (annotations added); *see also id.*, 9:27-30 ("during the creation of passages **152** authoring tool **86** and ibook server application **64** cause information to be stored in passage database **74** that identifies which Web page **138** contains each passage **152**").) Although navigation tool **84** also displays on-screen tools such as buttons (*e.g.*, **180**, **182**, **184**, **186, 188**, **192, 193**) and menus **196**, these are not part of the actual web page but user interface features of navigation tool **84**. (Reynolds, *e.g.*, 9:54-56, 9:62-10:16, 10:27-38.) Stated differently, all content displayed for an ibook web page is based on the passages **152** associated with the page. Under the combination of Reynolds with Lemay Web and Lemay Java, therefore, each web page in an ibook is defined **entirely** by the passages ("**the plurality of objects**") comprising that web page and their associated styles. (EX1002, ¶95.)

Additionally, under the proposed combination, each web page is not only defined entirely by each of the plurality of objects comprising that web page, but also by "**the style associated with the object**," as claimed. As previously explained for element 1**[c]**, each passage (*i.e.*, object) in the proposed combination can also have an associated style specifying how an animated image object is to be displayed. Because each style specifies how its corresponding object will be displayed on the page, the prior art renders obvious that each web page is also defined entirely by each object and "**the style associated with the object**." (EX1002, ¶96.)

(e) "produce a database with a multidimensional array comprising the objects that comprise the web site including data defining, for each object, the object style, an object number, and an indication of the web page that each object is part of, and" (Claim 1[e])

Reynolds discloses and renders obvious that its ibook server application produces a database comprising the objects that make up the web site. As explained for claim 1**[a]-[b]**, Reynolds' ibook server application **64** receives user input defining passages via authoring tool **86** and stores data relating to the passages in its databases. The relevant databases in Reynolds are shown in Figure 3:



(Reynolds, Fig. 3 (highlighting added).) Figure 3 above shows ibook server application **64** ("**build engine**"), as well as passage database **74** and ibook web pages database **76**. The build engine **produces** these databases by controlling the operation of server **56** and the databases that reside within it, which store ibook information. (Reynolds, *e.g.*, 5:19-22 ("Ibook server application **64** directs the overall operation of ibook server **56**. In particular, ibook server application **64** controls databases **66**, **68**, **70**, **72**, **74**, and **76**, in which various ibook information is stored.").)

As will be first explained below, database storage of the particular information items in claim 1**[e]** (*i.e.*, "**the objects that comprise the web site," "the object style," "an object number," "an indication…**") is disclosed and rendered obvious

by Reynolds. It would also have been obvious, in view of Perry and Witkowski, to adapt the databases in Reynolds to store the information recited in claim 1**[e]** in "**a database with a multidimensional array**."

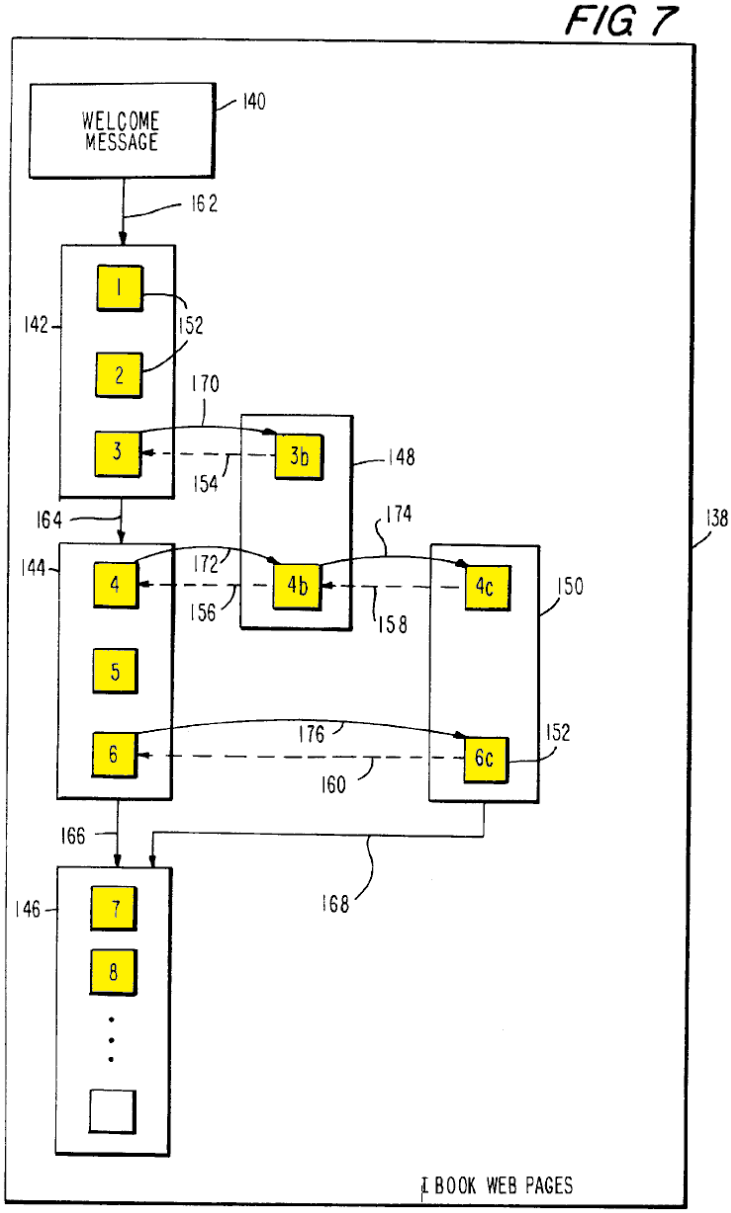
Reynolds discloses that its databases store "**the objects that comprise the web site**," as claimed. As previously explained, the content of ibook web pages is contained in passages stored in Web pages database **76**. (Reynolds, *e.g.*, 1:34-38 ("Within each Web page, material may be organized in passages. Passages can be of any suitable size, such as a paragraph for textual material, or a video or audio clip of a certain length for multimedia material."), 7:37-39 ("The content of the ibook Web pages is preferably stored in ibook Web pages database **76** (FIG. 3).").) Each passage of an ibook web page in Reynolds is stored as a discrete entity (as explained for claim 1**[b]**). Indeed, to view an ibook web page, navigation tool **84** individually requests passages from the server (which in turn retrieves those passages from database **76**) on a passage-by-passage basis. (Reynolds, *e.g.*, 11:28-40.) Reynolds thus discloses and renders obvious producing "**a database…comprising the objects that comprise the web site**," as claimed. (EX1002, ¶98.)

The databases under the proposed combination also "**includ**[e] **data defining, for each object, the object style**," as claimed. In particular, data defining the style for each passage ("**for each object, the object style**") would be provided by the user through Reynolds' authoring tool **86**, and thus it would have been obvious that the

same databases in Reynolds that store passages would likewise store their associated styles. (EX1002, ¶99.) Using the animated image object as described for claim 1**[c]**, the style is specified within the HTML code for the web page on which the animated image will appear. Thus, the styles would be stored in the same Web pages database **76** of Reynolds that stores page content. (*Id.*)

Reynolds' databases also store data defining, for each passage ("**for each object**") in an ibook**, "an indication of the web page that each object is part of**." In particular, "during the creation of passages **152** authoring tool **86** and ibook server application **64** cause <u>information to be stored in passage database **74** that identifies which Web page **138** contains each passage **152**</u>." (Reynolds, *e.g.*, 9:27-30.) Thus, Reynolds discloses storage in a database of an indication of the web page(s) that each passage is a part of. (EX1002, ¶100.)

Reynolds also renders obvious storing "**an object number**," as claimed. Figure 7 of Reynolds provides an exemplary ibook including web pages composed of passages ("**object**[s]") each having a number:



(Reynolds, Fig. 7 (highlighting added).) "Web pages **142**, **144**, **146**, **148** and **150** are preferably organized in the form of one or more passages **152**." (Reynolds, 7:66-67.) Each of these web pages includes passages that are numerically labeled, as highlighted in Figure 7. "For example, ibook Web page **142** contains passages 1, 2, and 3 and ibook Web page **150** contains passages **4c** and **6c**." (Reynolds, 7:67-8:2

(highlighting added).) Reynolds therefore discloses the association of a passage with a numeric value ("**object number**"). (EX1002, ¶101.)

Although Reynolds does not specify whether the passage numbers discussed above are stored in the database (as opposed to identifying passages in the figure), Figure 7 nevertheless confirms that assigning a number to each passage ("**object number**") would have been obvious to an ordinarily skilled artisan. As noted previously, Reynolds explains that during the creation of passages **152**, information is stored in passage database **74** "<u>that identifies which Web page **138** contains each passage **152**</u>." (Reynolds, 9:27-30.) But in order to keep track of the web pages containing a particular passage, the system must assign some kind of value to each passage in the database to allow it to be identified. (EX1002, ¶102.) An ordinarily skilled artisan would have appreciated that a number is the simplest and most easily-implemented way of representing such an identifier, which is confirmed by the use of numbers in Figure 7 to identify each passage. (*Id*.)

Thus, Reynolds discloses and renders obvious storing in a database "**the objects that comprise the web site including data defining, for each object, the object style, an object number, and an indication of the web page that each object is part of**." Further, as noted, Reynolds discloses that ibook server application **64** (the "**build engine**") "**produce**[s] **a database**," as claimed. (Reynolds, 5:20-23 (explaining that ibook server application **64** "directs the overall

operation of ibook server **56**," and, in particular, "controls databases **66**, **68**, **70**, **72**, **74**, and **76**, in which various ibook information is stored").) It therefore would have been obvious for Reynolds' "**build engine**" to have been configured to produce the database as recited in claim 1**[e]**.

Although Reynolds does not itself describe the ability to "**produce a database with a <u>multidimensional array</u>**" comprising the above-identified information, the claimed database with a multidimensional array would have been obvious to an ordinarily skilled artisan in view of **<u>Perry</u>** and **<u>Witkowski</u>**. (EX1002, ¶¶104-109.)

Perry confirms that the '168 patent inventors did not invent the concept of a "**multidimensional array**." In computer science, the term "**array**" refers to a well-known data structure consisting of a number of elements, each element identified by an index that identifies the location of the element in the array. (EX1002, ¶105.) One simple example of an array is a string of characters, such as the five-character string "hello" containing five character elements. (*Id*.) Each element of this array is accessible using an index value that identifies a particular position within the array from zero to four—an index value of zero corresponds to the letter "h" at the front of the string and an index value of four corresponds to the concluding "o" at the end. (*Id*.) This exemplifies a *one-dimensional* array because only one index value (from zero to four) is needed to access each element in the array. (*Id*.)

A multidimensional array, as its name implies, requires at least *two* indices to access each array element. (EX1002, ¶106.) As Perry explains, "[m]ultidimensional arrays, sometimes called *tables* or *matrices*, have at least two dimensions (rows and columns)." (Perry, Ex. 1006, p.455 (italics in original).) "Whereas a single-dimensional array is a list of values, a multidimensional array simulates a table of values, or multiple tables of values. The most commonly used table is a two-dimensional table…." (Perry, p.456.) Table 27.1 from Perry provides a straightforward example of two-dimensional array for "a football team [that] wanted to track its players' total yardage. The team has eight games scheduled, and there are six rushing players on the team." (Perry, p.456.)

Table 27.1. A football team's yardage record.

Player Name	*Game 1*	*Game 2*	*Game 3*	*Game 4*	*Game 5*	*Game 6*	*Game 7*	*Game 8*
Downing	1	0	2	1	0	0	0	0
Edwards	0	3	6	4	6	4	5	3
Franks	2	2	3	2	1	0	2	3
Grady	1	3	2	0	1	5	2	1
Howard	3	1	1	1	2	0	1	0
Williams	2	3	1	0	1	2	1	2

(Perry, Table 27.1, p.456.)

As Perry explains, Table 27.1 "has rows (the first dimension) and columns (the second dimension). Therefore, this is a two-dimensional table with six rows and eight columns." (Perry, p.456.) "Each row has a player's name, and each column has a game number, but these are not part of the actual data. The data consists only of 48 values (six rows by eight columns)." (*Id.*)

Table 27.1 is a "multidimensional array" because it provides a uniquely identifiable indexed set of related elements (e.g., yardage values), where each element is addressed by a set of two or more indices (e.g., the player name (row) and the game number (column)), with each index corresponding to a dimension of the array. (EX1002, ¶108.)

The basic underlying concept behind a multidimensional array—the ability to store data in tables accessible by rows and columns—also underlies some relational database implementations. (EX1002, ¶109.) This is confirmed by **Witkowski** (Ex. 1007), a patent assigned to Oracle that is relied upon merely for its description of prior art database technologies. Witkowski's "Background of the Invention" provides the following description:

> In a database management system (DBMS), data is stored in one or more data containers, each container contains records, and the data within each record is organized into one or more fields. In relational database systems, the data containers are referred to as tables, the records are referred to as rows, and the fields are referred to as columns.

(Witkowski, 1:20-25.) Witkowski sets forth several examples of simple database tables that are organized into rows and columns, including TABLE X illustrated in Figure 4a:

TABLE X

x.row id	X.a	X.b
X1	1	1
X2	2	2
X3	3	3
X4	4	4
X5	9	0

(Witkowski, Fig 4A (partial figure; highlighting added).) "Referring to FIG. 4a," Witkowski explains, "table X includes two columns x.a and x.b and a pseudo-column 'X.rowid'. The pseudo column contains a unique identifier for each row in table X. The pseudo-column may, for example, simply represent the addresses of

the actual storage locations at which the rows are stored." (Witkowski, 15:54-59.)[2] Although Witkowski also describes various techniques for "joining" Table X with Table Y (also shown in Figure 4A), those techniques are not relevant to the limited purpose for Witkowski is relied upon, *i.e.*, to illustrate an exemplary database table that stores information in rows and columns. Witkowski's table is an exemplary "**multidimensional array**," as claimed, because it provides a uniquely identifiable indexed set of related elements (Table X), in which each element is addressed by two indices—the row id (*i.e.* x.rowid) and the column (*i.e.* x.a or x.b). (EX1002, ¶109.) For example, to obtain the value highlighted in yellow above, one could request the database to retrieve the value in column x.a within the row whose id is X1. (*Id.*) Witkowski therefore confirms that data can be stored in a database through a "multidimensional array" by using a database table that stores data in rows and columns, with indices for addressing dimensions (*i.e.* row or column). (*Id.*)

---

[2] Witkowski describes Figure 4A as an example of a "base table," or the values of a table at a particular time. The proposed combination relies on Figure 4A solely for disclosure of a basic table in a relational database system (which would have been general knowledge to a skilled artisan), and not on the specific table joining techniques described later in Witkowski. (EX1002, ¶109 n.2.)

Manner, Rationale and Motivation to Combine: The proposed combination of Reynolds with Perry and Witkowski would have been obvious to an ordinarily skilled artisan, predictably resulting in the system of Reynolds further adapted to include **a database with a multidimensional array** that includes the claimed information and is stored in Reynolds's passage database **74** and/or web page database **76**. (EX1002, ¶110.) Thus, the proposed combination would have resulted in a relational database that stores the information recited in claim 1**[e]**, in a database table with rows for each record, and each row having columns or fields ("a database with a multidimensional array"). This combination could have been implemented in Reynolds in any number of ways, but the most straightforward would have been for the rows to correspond to ibook passages ("**the objects that comprise the web site**"), and for each passage, a number of columns ("**including data defining, for each object**") containing the above-discussed data relating to passages. (*Id.*) This would have qualified as a **multidimensional array** because the table would have provided a uniquely identifiable indexed set of related elements, with each element identified by two indices—an identifier of the row (passage) and an identifier of the column within the row where the desired information is stored. (*Id.*)

An ordinarily skilled artisan would have considered this combination straightforward, and in fact, trivial. (EX1002, ¶111.) Implementation of

multidimensional arrays is considered elementary programming knowledge well within the knowledge of ordinarily skilled artisans, and often taught in introductory undergraduate computer programming courses. (*Id*.) More fundamentally, storing data using two dimensions and accessible using two indices (*e.g.* through rows and columns) is a natural and fundamental way human beings organize information in the "real world," as evidenced by its widespread use with spreadsheets, bookkeeping ledgers, and numerous other real world applications—as exemplified by Perry's football player table. (Perry, Table 27.1, p.456; EX1002, ¶111.) Indeed, any situation that requires storage of information about (a) a number of items and (b) pieces of information about each item lends itself easily to a two-dimensional array with information stored in rows and columns. (EX1002, ¶111.) Thus, it would have been obvious to implement a database in Reynolds, which must store a number of pieces of information about each passage, through a two-dimensional array such as a table with rows and columns. (*Id*.)

And like multidimensional arrays, relational databases were also well-known in the art. Indeed, Reynolds expressly discloses the storage of passage information in a "relational database":

> Although various information concerning the ibook passages, such as derivation linking information, derivation type information, passage characteristics, and attribution information have been described as being stored in passage databases, **a suitable alternative is to store**

> **this information in the Web pages that contain the passages or as a set of relations and tables in a suitable relational database**, or as a set of objects in an object database.

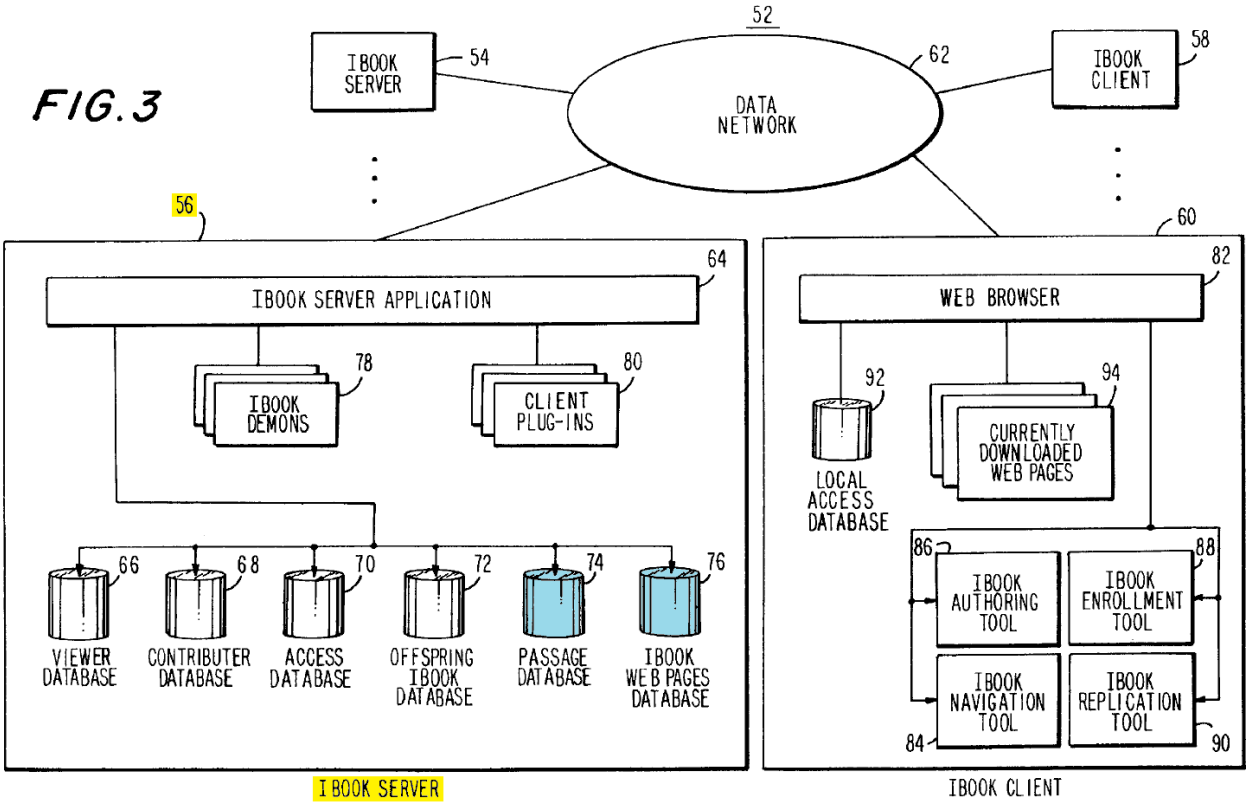
(Reynolds, 15:30-37.) By December 1999, storing data in the form of tables using relational databases was a popular (if not the most popular) form of database storage. (EX1002, ¶112.) Relational databases were commercially available through off-the-shelf tools from third party vendors such as Oracle (the assignee of Witkowski), as well as freely available open source databases (e.g., MySQL). (*Id*.) An ordinarily skilled artisan would have been motivated to use relational database technologies to implement table-based database storage of the passage-related information of Reynolds because relational databases were technologically mature, well-documented, stable, and could have been readily incorporated into existing applications. (*Id*.) Thus, an ordinarily skilled artisan would have had every expectation of success in implementing the database of Reynolds as a relational database which organizes data into a multidimensional array organized into rows and columns. (*Id*.)

Harnessing such preexisting relational database technologies would have enabled a skilled artisan to more quickly implement Reynolds' ibook system, as it would have obviated the need to undertake the potentially time-consuming and costly process of designing and developing a "home-grown" data storage technique.

(EX1002, ¶113.) Moreover, by December 1999, relational databases represented a reliable, decades-old technology that would have enhanced the reliability of Reynolds' database. (*Id.*) Developing a brand new and untested home-grown data storage technique, on the other hand, would have further increased delay and costs associated with application development due to requiring considerably more testing to expose and correct any potential programming bugs. (*Id.*)

(f) "provide the database to a server accessible to [sic; a] web browser;" (Claim 1[f])

Reynolds discloses that web pages database **76** is provided to a server accessible to web browser **82**. (EX1002, ¶114.) As shown in Figure 3, as noted, web pages database **76** is provided to web server **56**:



(Reynolds, Fig. 3 (highlighting added).) Passage database **74** and ibook web pages database **76** are provided to ibook server **56** and ibook server application **64**. In particular, ibook server application **64** (the "**build engine**") **provides** these databases to server **56**, because the build engine controls the databases that reside within server **56** as shown in Figure 3. (Reynolds, 5:19-22 ("Ibook server application **64** directs the overall operation of ibook server **56**. In particular, ibook server application **64** controls databases **66**, **68**, **70**, **72**, **74**, and **76**, in which various ibook information is stored.").) As previously explained for claim **1[e]**, the proposed further combination of Reynolds with Perry and Witkowski would store the information in these databases in "**a database with a multidimensional array**," as claimed. The proposed combination therefore renders obvious that the build engine is configured to "**provide the database to a server**."

Reynolds server, and the claimed database, are "**accessible to** [a] **web browser**," *i.e.* Web browser **82** of Figure 3 above. (EX1002, ¶115.) As Reynolds explains, an ibook client **60** viewing an ibook uses Web browser **82** (with navigation tool **84** installed through a plugin downloaded from Web server **56**) to retrieve passages from web pages database **76** at the server **56**, and display web pages composed of the retrieved passages. (Reynolds, 5:27-31 (describing browser plugins **80** as "downloadable executable program modules that can be run on client **60** in conjunction with Web browser **82** to add additional capabilities to Web browser

82"), 5:35-37 ("When client plug-ins **80** are downloaded onto client **60**, they form tools **84**, **86**, **88**, and **90**."), 11:24-39 (describing how navigation tool **84** communicates with server application **64** to retrieve ibook website content from web pages database **76** to generate ibook web pages).)[3] Thus, the database—adapted to include a multi-dimensional array as described for claim 1**[e]**—and server would be "**accessible to** [a] **web browser**," as claimed.

(g) "wherein the database is produced such that a web browser with access to a runtime engine is configured to generate the web-site from the objects and style data extracted from the provided database." (Claim 1[g])

This limitation was largely addressed by the preceding sections. Reynolds discloses the claimed "**runtime engine**" in the form of navigation tool **84**, which is one of the client plug-ins **80** that "are downloadable executable program modules that can be run on client **60** in conjunction with Web browser **82** to add additional capabilities to Web browser **82**." (Reynolds, 5:28-31; EX1002, ¶116.) "When client

[3] Claim 1**[f]** is unclear as to whether the "**database**" or the "**server**" must be "**accessible to** [a] **web browser**." This distinction is immaterial because Reynolds discloses that both the server and the database are accessible to Web browser **82**, as the browser can access server **56** over network **62** and can also retrieve information from the database, as explained in the text.

plug-ins **80** are downloaded onto client **60**, they form tools **84**, **86**, **88**, and **90**." (Reynolds, 5:35-37.)

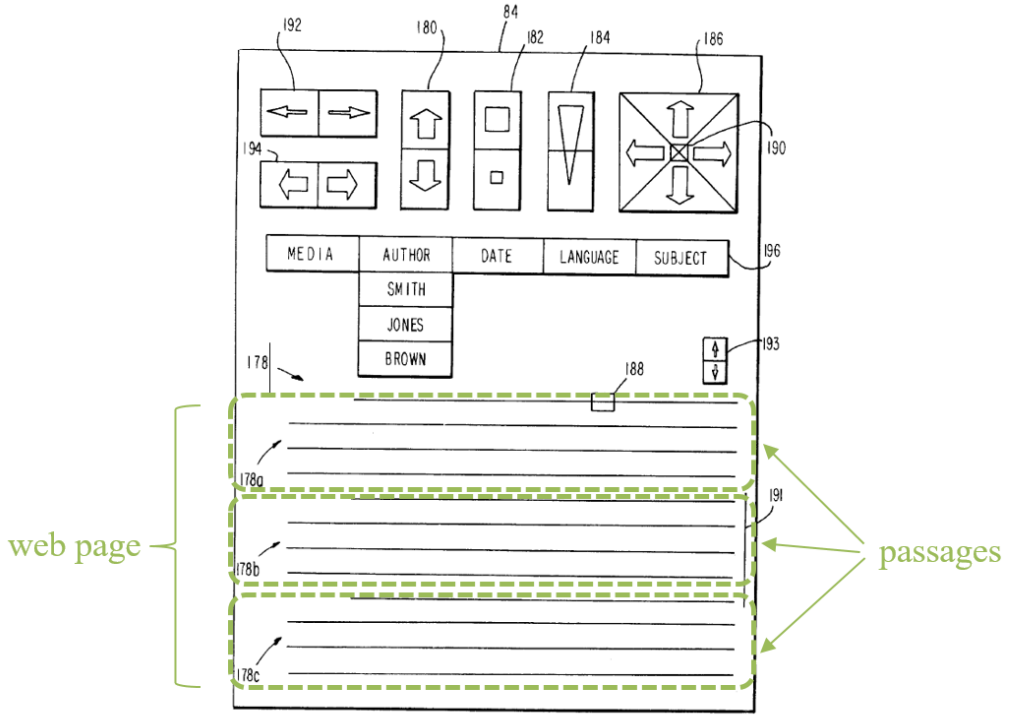
Navigation tool **84** qualifies as a "**runtime engine**," because as Reynolds confirms, the tool is a file executed at runtime (*e.g.* in web browser **82**) to facilitate retrieval of information from the database and generate commands to display the ibook web pages in the browser. Navigation tool **84** builds web pages from passages retrieved from the database, as Reynolds explains:

> At step 198, the user at client **60** determines which passage is to be viewed, for example, by using the various buttons provided by navigation tool **84** (FIG. 8). Navigation tool **84** preferably generates a request to view a desired passage and transmits this request to server **56** at step 200. Server application **64** at server **56** receives the request and locates the requested passage in ibook Web pages database **76** at step 202… Server **56** transmits the requested passage to client **60** at step 206. Client **60** displays the passage and may make a time and date entry in local access database **92** at step **208**.

(Reynolds, 11:28-40.)

Figure 8 of Reynolds shows how navigation tool **84** on the client generates a web page at run-time, built from passages **178**a, **178**b, and **178**c retrieved from the server and database:

FIG. 8



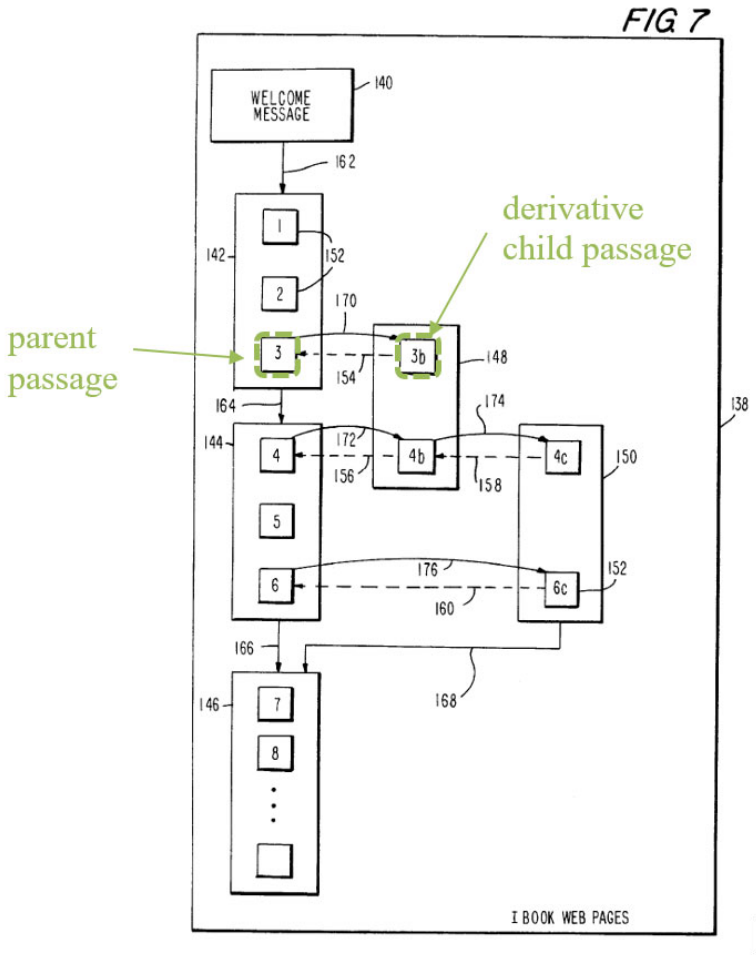
(Reynolds, Fig. 8 (annotation added).) As previously explained, "[w]hen passages **152** are executable modules"—such as executable Java applets under the combination of Reynolds, Lemay Web, and Lemay Java discussed for claim 1**[c]**—"navigation tool **84** presents the passages by downloading and executing the passages." (Reynolds, 9:52-54.) Navigation tool **84** includes various "navigation controls" enabling the viewer to navigate the ibook website, such as cursor keys **186** for navigating within a passage, magnification buttons **184**, and back and forward buttons **194** for navigating between different Web pages. (Reynolds, 9:48-10:16, 10:27-31.) "As the user navigates through an ibook using navigation tool **84**," "navigation tool **84** transmits certain access information to server **56**, making

requests to view material." (Reynolds, 11:24-26.) In other words, as the viewing user navigates through the web pages of the ibook website, navigation tool **84** generates those web pages by requesting and displaying the passages of those web pages from the web server, which are retrieved (i.e., extracted) from the database at the web server. (Reynolds, 11:24-38; EX1002, ¶118.)

For the above reasons, Reynolds discloses that the database "**is produced such that a web browser with access to a runtime engine**"—*i.e.* web browser **82** executing navigation tool **84**—"**is configured to generate the web-site from the objects and style data extracted from the provided database**." Under the combination of Reynolds, Lemay Web, and Lemay Java previously described for claim 1**[b]-[c]**, ibook passages are associated with style data (*e.g.* data defining the appearance of a Java image or button object), which as explained for claim 1**[d]** is stored in the database. (EX1002, ¶119.) Thus, under the proposed combination, navigation tool **84** will generate the website for display in the browser from passages ("**objects**") and "**style data"** extracted from the database. (*Id.*)

2. Claim 2: "The system of claim 1, wherein one of said plurality of objects is a child, and wherein the build engine is configured to accept user input to associate a style with child button and child image objects."

Reynolds discloses that passages ("objects") can have a **child**, in the form of a derivative passage. Figure 7 shows an exemplary derivative passage 3b:



(Reynolds, Fig. 7 (annotation added).) Reynolds and Lemay Web render obvious that server application **64** (the "**build engine**") "**is configured to accept user input to associate a style with child button and child image objects**" in two ways: (1) by accepting user input that directly associates a style with a derivative passage ("object"); or (2) by associating the user-inputted style of the parent passage to the derivative passage. (EX1002, ¶¶120-125.)

First, as previously explained for claim 1**[c]**, the proposed adaptation of Reynolds' server application **64** based on Lemay Web would enable the user to

specify for each web page, during the process of creating an ibook passage, an animated image object (e.g., by specifying an instance of Lemay Web's "Animator" Java applet)—including attributes defining the object's visual appearance (i.e., the object's "**style**"). The proposed adaptation of Reynolds' server application **64** applies equally to all of Reynolds' passages, thus including derivative passages that are "based on" the parent passage but otherwise created by the author of the derivative passage. (Reynolds, 1:47-48 ("Works that are created based on previously contributed material are called derivations."); EX1002, ¶121.) Under the proposed combination, Reynolds' ibook creation system would accept user input to associate a style with an animated image object of a derivative passage that is otherwise based on a corresponding animated image object of a parent passage. (EX1002, ¶121.)

Alternatively, the content of a parent passage may be copied to, and thus associated with, a derivative passage. (Reynolds, 1:47-61 ("Works that are created based on previously contributed material are called derivations" that can be created "us[ing] certain authoring functions in the ibook system, such as by performing a cut and paste operation from existing material"), 8:28-35 ("Attribution information is preferably automatically generated when a contributor copies existing ibook material. For example, contributor Jones may have used a cut and paste feature in derivation tool **132** to cut and paste text or multimedia material from passage 3 when

creating passage 3b. During the process of cutting and pasting from passage 3 to passage 3b, authoring tool **86** automatically generates attribution information including attribution link **154**.”).) Copying an animated image object from a parent passage to a derivative passage would naturally include copying the style defining the object’s visual appearance. (EX1002, ¶122.)

Two constructions have been proposed in previous litigations for “child”/“child object”/“child button”: “*an element with one or more properties that depends on another element*” and “*element/object/button that is spawned by another element/object/button (i.e., the parent element/object/button)*.” (*See supra*, Part V.) Reynolds’ derivative passages meet either construction because the content of a derivative passage “*depends on*” and is likewise “*spawned by*” its parent passage. (EX1002, ¶123.)

A derivative passage not only includes content derived from a parent passage, but also has attribution properties linking the derivative passage to the parent passage—further indicating that at least part of the derivative passage *depends on* and is *spawned* from a corresponding parent passage:

> Attribution information is preferably automatically generated when a contributor copies existing ibook material. For example, contributor Jones may have used a cut and paste feature in derivation tool **132** to cut and paste text or multimedia material from passage 3 when creating passage 3b. During the process of cutting and pasting from passage 3

> to passage 3b, authoring tool **86** automatically generates attribution information including attribution link **154**.

(Reynolds, 8:28-35; *see also id.*, 8:19-21 ("Attribution link **154** indicates that passage 3b, which was contributed by Jones, was based on passage 3, contributed by Smith."), 8:61-9:7 ("Derivation links **170**, **172**, **174**, and **176** are another type of link that is formed in connection with ibook web pages **142**, **144**, **146**, **148**, and **150**. Derivation links **170**, **172**, **174**, and **176** preferably point in the opposite direction of attribution links **154**, **156**, **158**, and **160**."); EX1002, ¶124.) The child passage's attribution properties—in the form of derivation link **170** and attribution link **154**—are stored in passage database 74 to be "used by navigation tool 84 when a user navigates through the material in an ibook." (Reynolds, 9:8-19.) When navigating through the ibook, a user is "alerted" when a "derivation exists" and is "given an opportunity to view the derivation" passage (*i.e.*, child). (Reynolds, 9:12-15.) Conversely, the user is alerted when a passage is a derivation and has a "corresponding original" passage (*i.e.*, parent). (Reynolds, 9:16-17.) Thus, both the content and the attribution properties of the child passage *depend on* and are *spawned from* the parent passage.

For the above reasons, it would have been obvious for an ordinarily skilled artisan to adapt Reynolds' ibook server application to accept user input to associate a style with animated image objects (which are also button objects) of derivative

child passages, either through user input of new animation styles for the derivative child passages or through copying the user-specified styles from the parent passages.

3. Claim 3: "The system of claim 2, wherein at least one of said styles includes values defining time lines for child button and child image objects."

Under the proposed combination of Reynolds and Lemay Web as described for claim **2**, Reynolds' ibook server application would accept user input to associate an animation style for animated image objects (which are also button objects) of derivation (*i.e.*, "**child**") passages. As also explained for claim 1**[c]**, it would have been obvious in view of Lemay Web and Lemay Java for that animation style to include "**values defining time lines for the… objects"** in the form of <PARAM> tags such as "STARTIMAGE," "ENDIMAGE," "PAUSE," and "REPEAT." (*See, e.g.*, Lemay Web, p.256.) The combination of Reynolds, Lemay Web, and Lemay Java therefore renders obvious that the animation **style** for animated image objects of a derivation passage (**child button and child image objects**) includes **values defining time lines for the child button and child image objects**, as claimed. (EX1002, ¶126.)

4. Claim 4: "The system of claim 1, wherein at least one of said styles includes settings for multiple object states."

As previously explained for claim **1[c]**, Reynolds and Lemay Web disclose and render obvious that the user-defined animation style for Java's Animator applet would include the STARTIMAGE and ENDIMAGE parameters defining which

images (and how many) are used for the animation style. (*See, e.g.*, Lemay Web, p.256 (describing how a user sets the parameters for the Animator Java applet).) Reynolds and Lemay Web therefore disclose styles including "**settings for multiple object states**," because each image represents a different object state, and the animation transitions between successive states each displaying a different image (as set by STARTIMAGE and ENDIMAGE). (EX1002, ¶127.)

5. Claim 6: "The system of claim 1, where said data is stored as one or more of a Boolean an integer, a string, a floating point variables, or a URL."

Claim 6 depends from claim 1, which recites "produce a database with a multidimensional array comprising the objects that comprise the web site including **data defining**, for each object, the object style, an object number, and an indication of the web page that each object is part of[.]" "**Said data**" in claim 6 therefore refers to the specific data (*i.e.* object style, object number, and indication of the web page of which each object is part of) recited in claim 1. But the storage of such data as "**a Boolean**[,] **an integer, a string, a floating point variables, or a URL**," adds nothing significant and is disclosed by or obvious in view of the prior art cited for claim 1. (EX1002, ¶¶128-133.)

As previously explained for claim **1**, under the combination of Reynolds, Lemay Web, and Lemay Java, the "**object style**" is expressed through a series of

HTML statements (*e.g.* <APPLET> and <PARAM> tags) which identify the Java "Animator" applet and specify characteristics of its visual appearance on the screen.

```
<APPLET CODE="Animator.class" WIDTH=100 HEIGHT=100>
<PARAM NAME="STARTIMAGE" VALUE="1">
<PARAM NAME="ENDIMAGE" VALUE="12">
<PARAM NAME="PAUSE" VALUE="1000">
<PARAM NAME="REPEAT" VALUE="TRUE">
</APPLET>
```

(Lemay Web, p.256.) These HTML statements are a collection of textual characters that would have been stored as a "**string**," as claimed. (*See* EX1009, p.451 (defining "string" as "[a] data structure composed of a sequence of characters usually representing human-readable text"); EX1002, ¶129.) In addition, the <APPLET> tag stores a textual **string** containing the Animator applet's name ("Animator.class"), and the <PARAM> tags store "**integer**" variables (*i.e.* values for the STARTIMAGE, ENDIMAGE, PAUSE parameters) and "**Boolean**" variables (*i.e.* the TRUE or FALSE value for REPEAT). (EX1002, ¶129.)

As for "**object number**," it would have been obvious (as previously explained for claim 1) to assign a numeric value to each passage for the purposes of facilitating identification of those passages. An ordinarily skilled artisan would have recognized that an "object number" would be represented at least as an "**integer**." (*See* EX1002, ¶130; EX1009, p.254 (defining "integer" as "[a] data type representing whole numbers," which is "widely used in programming for counting and numbering

purposes").) It would have been obvious to store the object number using such a data type that was "widely used in programming" for the very purpose of "counting and numbering." (EX1002, ¶130; EX1009, p.254.)

As for the claimed "**indication of the web page that each object is part of**," it would have been obvious to store this data as the claimed "**URL**." As previously explained for claim 1, Reynolds explains that "during the creation of passages **152** authoring tool **86** and ibook server application **64** cause information to be stored in passage database **74** that identifies which Web page **138** contains each passage **152**." (Reynolds, 9:27-30.) Although Reynolds does not state how this information is formatted, it would have been obvious to store it in the form of a Uniform Resource Locator (URL), which was the standard (and widely known) format to represent and uniquely identify documents or resources on the World Wide Web. (EX1002, ¶131.) Reynolds itself confirms this by describing various examples of storing URLs to identify ibook web pages and passages in ibook Web pages database **76**. (Reynolds, *e.g.*, 15:24-26 ("For example, server **278** may contain the URLs of passages **276** and **290** in ibook database such as passage database **74** and ibook Web pages database **76**."), 11:53-55 ("The passage identifier is preferably a universal resource locator (URL) for the passage requested from the ibook Web pages database **76**."), 15:48-51 ("If desired, ibook Web pages on a given server or client may be represented by a reference (e.g., a URL) to material that is actually stored elsewhere in the data

network.”).) Because Reynolds expressly discloses how URLs can be used to identify passages and web pages, and the use of URLs to identify web pages was widely known to ordinarily skilled artisans and even casual users, it would have been obvious to employ URLs to track which objects are associated with which web pages, thus providing the claimed “indication” of the web page of which each object is a part. (EX1002, ¶131.)

Petitioner notes that, in the end, claim 6 adds nothing of patentable significance because regardless of what information is stored in the database, it would have been obvious to store it as “**Boolean,**” “**integer,**” “**string,**” or “**floating point variables**,” because these represent basic and fundamental data types for representing information in a computer system. (EX1002, ¶132.) As the *Microsoft Press Computer Dictionary* explains, “[d]efining the data type allows a computer to manipulate the data appropriately,” and “[d]ata types are most often supported in high-level languages and <u>often include types such as</u> real, <u>integer, floating point, character, Boolean</u>, and pointer.” (EX1009, p.133.) Often taught in introductory undergraduate computer programming courses, these data types would have been familiar to any ordinarily skilled artisan. (EX1002, ¶132.) Indeed, understanding how to use different data types (including at least integer and string variables) would have been needed just to build a functioning computer program. (*Id.*) At bottom, it would have been plainly obvious to use such fundamental data types to represent

data in the database (including the data recited in claim 1), and a skilled artisan would have been motivated to employ widely-used data types already "supported in high-level languages." (EX1009, p.133; EX1002, ¶132.)

For example, claim 6 encompasses the storage of the claimed data as a "**string**," which is simply "a sequence of characters" that can be used to store any type of computer-usable information. (*See* Ex. 1009, p.451; EX1002, ¶133.) Indeed, even if "string" were to be more narrowly interpreted as a sequence of human-readable textual (ASCII) characters, the use of strings to represent data was well-known. (EX1002, ¶133.) For example, well-known encoding techniques could encode any arbitrary data as a sequence of readable textual characters—a technique widely used to transmit data over Internet using email or newsgroups. (*Id.*; *see also* Ex. 1009, p.489 (defining uuencode), p.51 (defining BinHex), p.309 (defining MIME), p.475 (defining translated file).) It would therefore have been obvious for <u>all</u> data in Reynolds' database,[4] including data used to generate the "plurality of web

[4] For example, as explained earlier in the discussion of claim 6, it would have been obvious to store the claimed "**indication of the web page that each object is part of**," in the form of a URL. (EX1002, ¶131.) But the claimed indication could also be represented as a **string** as a URL is simply a particular type of string. As

pages," to be stored in the form of strings. A skilled artisan would have been motivated to do so because strings are a fundamental and well-known data type implemented in any high level programming language, and also ideal for server-client transmissions using known communications protocols for transferring data (e.g., HTTP). (EX1002, ¶133 (citing, *e.g.*, Ex. 1009, p.451; Coombs, p.351 (defining the "Hypertext Transfer Protocol (HTTP)" as "[a] system for transferring files across the World Wide Web")).) For example, under the proposed combination, strings could have been used to store the data in the database in a form suitable for ready transmission to Reynolds' navigation tool **84** at the client using conventional Internet protocols. (EX1002, ¶133.) In turn, navigation tool **84** would use the transmitted information to generate the ibook web pages, as explained for claim 1**[g]**.

C. Ground 2: Obviousness of Claims 1-4 and 6 Over Reynolds, Lemay Web, Lemay Java, Perry, Witkowski, and Coombs

1. Claim 1

As explained for Ground 1, Reynolds discloses a method for users to assemble **websites** in the form of interactive Web books ("ibooks") accessible via the World Wide Web. (Reynolds, Abstract; *see also id.*, 1:5-8 ("This invention relates to the Internet, and more particularly, to techniques for creating and viewing material on

---

explained in the text, it would have been obvious that any of the pieces of information stored in the database of claim 1 could be stored as strings.

the World Wide Web in the form of an interactive Web book.").) Each "Web book," or "ibook," is a collection of web pages. (Reynolds, 3:46-47 ("Contributed material is preferably stored in the form of ibook web pages **42**"), 7:36-37 ("The material contributed to an ibook is contained in one or more Web pages").). As shown for Ground 1, the collection of related web pages for an ibook forms a "**website**."

But Reynolds does not expressly use the word "**website**" to describe the collection of ibook web pages. In the event Patent Owner were to argue that "**website**" requires that the ibook web pages be accessed through a separate and dedicated domain name (such as "www.*domainname.com*"), and speculate that Reynolds does not disclose a website under that definition, such an interpretation would nevertheless have been obvious. (EX1002, ¶137.) In particular, it would have been obvious to a skilled artisan to register a domain name for an ibook such that viewing users would access the ibook web pages through a URL defined by a separate and dedicated domain name. (EX1002, ¶¶137-138.) For all other limitations of claim 1, the analysis from Ground 1 applies here in the same way.

Registering a domain name was, and continues to be, one of the most easy and basic initial steps to creating a website. (EX1002, ¶138.) As **Coombs** explains, obtaining a domain name for a website merely required confirming the domain name was not already taken, submitting an application for the domain name with the appropriate registrar, and paying the appropriate fee. (Coombs, pp.29-39.) The

"first steps that everyone needs to take in setting up an Internet site," such as a web site, are "getting a domain name and establishing a presence on the Internet." (Coombs, p.3.)

Rationale and Motivation to Combine: As taught by Coombs, it would have been obvious to a skilled artisan to implement the ibook system according to the teachings of Reynolds, Lemay Web, Lemay Java, Perry and Witkowski—as described in Ground 1—under a separate domain name. (EX1002, ¶¶ 139-143.) This would have predictably resulted in the ibook web pages being accessible under a dedicated domain name, for example, a URL in a form such as <http://www.reynoldsibook.com>. (EX1002, ¶139.) This URL is composed of various sections which identify the website: "http://" identifies the HTTP protocol to be used by the browser to communicate with Reynolds' web server, "www" identifies that the ibook is located on the "World Wide Web," and reynoldsibook.com identifies the domain which is translated to the IP address of Reynolds' web server. (EX1002, ¶139.) Thus, even under a narrow interpretation of website as requiring a separate domain name for an ibook, all limitations reciting a "**website**" in claim 1 would have been satisfied by the ibook of Reynolds in combination with Coombs, because the ibook web pages would have been available under a separate domain name dedicated to that ibook. For all other limitations of claim 1, the analysis from Ground 1 applies equally here.

A skilled artisan would have had many motivations to apply the teachings of Coombs in the manner proposed above. (EX1002, ¶140.) As explained above, Coombs confirms that domain name registration was a common first step to establish a separate website. Thus, the teachings of Coombs are analogous and reasonably pertinent to the problems faced by the inventors of the '168 patent because Coombs explains how to use the common technique of enabling visitor access to a website by establishing a separate domain name for the website. (*Id.*)

Both Reynolds and Coombs provide express motivations to combine. First, as any user of web pages who has typed in a web address into a browser would appreciate, a domain name provides a simple and easy way for visitors to reach a website. (EX1002, ¶141.) As Coombs explains, registering a domain name enables Internet users to access a website without "worry[ing] about translating computer names into IP addresses." (Coombs, p.25.) Thus, to make it easier to access the website, it would have been straightforward for a "sponsor who sets up an ibook on an ibook server" (Reynolds, 3:17-21), to register a desired domain name such as reynoldsibook.com. (EX1002, ¶141.)

Second, Coombs explains that registering a website domain name would have enabled a website developer to register the website with popular search engines such as Yahoo!. Reynolds itself recognizes the desirability of advertising the ibook's presence on the World Wide Web by adding the ibook to a "Web index" maintained

by a search engine such as "Yahoo, Alta Vista, and Lycos." (*See* Reynolds, 3:17-21 ("Ibooks may initially be supported by a sponsor who sets up an ibook on an ibook server and who may, for example, advertise that ibook's presence and chosen theme by arranging to have suitable information regarding the ibook included in Web index **40**."), 14:15-21 ("ibooks preferably support various functions for advertising their presence to popular search engine databases. Suitable search engines to which the ibook can advertise its presence include Yahoo, Alta Vista, and Lycos.").) As Coombs explains, "[t]aking advantage of search engines in promoting your site is the single most important thing you can do in marketing your site," and therefore "[h]aving Yahoo! include your Internet site in its directory is very important." (Coombs, p.246.)

Coombs confirms that the process of advertising via a search engine would have been made simple for websites having a separate domain and URL. As Coombs illustrates, a skilled artisan would have registered an ibook website to the Yahoo! directory by submitting a simple form to Yahoo! at http://add.yahoo.com/fast/add:

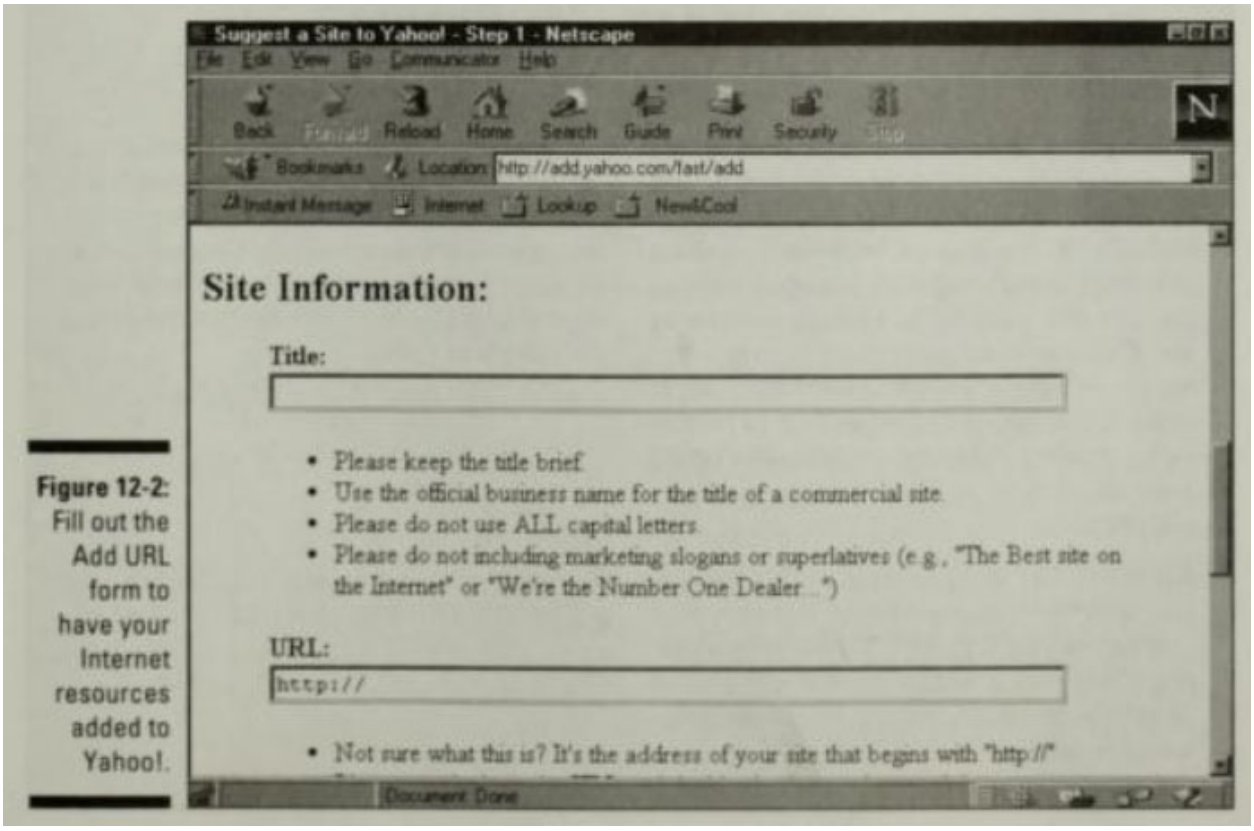


Figure 12-2: Fill out the Add URL form to have your Internet resources added to Yahoo!.

(Coombs, Fig. 12-2, p.249.) Thus, Reynolds' ibook sponsor could easily advertise an ibook web site on the Yahoo! web index by submitting the ibook's title and URL under the registered domain name, providing yet another motivation to combine Reynolds with Coombs. (EX1002, ¶¶142-143.)

2. Claim 2: "The system of claim 1, wherein one of said plurality of objects is a child, and wherein the build engine is configured to accept user input to associate a style with child button and child image objects."

As explained in Section VI.B.2, Reynolds in combination with Lemay Web discloses this limitation.

3. Claim 3: "The system of claim 2, wherein at least one of said styles includes values defining time lines for child button and child image objects."

As explained in Section VI.B.3, Reynolds in combination with Lemay Web and Lemay Java discloses this limitation.

4. Claim 4: "The system of claim 1, wherein at least one of said styles includes settings for multiple object states."

As explained in Section VI.B.4, Reynolds in combination with Lemay Web discloses this limitation.

5. Claim 6: "The system of claim 1, where said data is stored as one or more of a Boolean an integer, a string, a floating point variables, or a URL."

As explained in Section VI.B.5, this limitation would have been obvious in view of Reynolds.

VII. CONCLUSION

Petitioner respectfully requests IPR institution.

Dated: July 6, 2021

Respectfully submitted,

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW
Suite 700
Washington, DC 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

By: */ Heidi L. Keefe /*
Heidi L. Keefe
Reg. No. 40,673
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this petition complies with the type-volume limits of 37 C.F.R. § 42.24(a)(1)(i) because it contains 13,926 words, according to the word-processing system used to prepare this petition, excluding the parts of this petition that are exempted by 37 C.F.R. § 42.24(a) (including the table of contents, a table of authorities, mandatory notices, a certificate of service or this certificate word count, appendix of exhibits, and claim listings).

DATED: July 6, 2021

COOLEY LLP
ATTN: Patent Docketing
1299 Pennsylvania Avenue NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

/ Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673

CERTIFICATE OF SERVICE

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 7,594,168 B2,** including all exhibits (**Nos. 1001-1027**) and related documents, are being served via Federal Express on the 6th day of July 2021, the same day as the filing of the above-identified document in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon Patent Owner by serving the correspondence address of record with the USPTO as follows:

Lowenstein & Weatherwax LLP
1880 Century Park East, Suite 815
Los Angeles, CA 90067

And, via Federal Express upon counsel of record for Patent Owner in the litigation pending before the U.S. District Court for the Western District of Texas entitled *Express Mobile, Inc. v. Facebook, Inc.*, Case No. 6:20-cv-00803-ADA (W.D. Tex.) as follows:

David Alberti
Marc Belloli
Feinberg Day Kramer Alberti Lim Tonkovich & Belloli
577 Airport Blvd., Suite 250
Burlingame, CA 94010

Charles L. Ainsworth
Parker, Bunt & Ainsworth, P.C.
100 East Ferguson, Suite 418
Tyler, TX 75702

DATED: July 6, 2021

/ Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673