



US007831926C1

(12) **EX PARTE REEXAMINATION CERTIFICATE** (11956th)
United States Patent
Rohrbaugh et al.

(10) **Number:** **US 7,831,926 C1**(45) **Certificate Issued:** ***Dec. 6, 2021**(54) **SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES**(75) Inventors: **Gary B. Rohrbaugh**, Bellingham, WA (US); **Scott A. Sherman**, Seattle, WA (US)(73) Assignee: **Softview L.L.C.****Reexamination Request:**

No. 90/009,995, May 29, 2012

Reexamination Certificate for:

Patent No.: **7,831,926**
 Issued: **Nov. 9, 2010**
 Appl. No.: **11/738,486**
 Filed: **Apr. 21, 2007**

(*) Notice: This patent is subject to a terminal disclaimer.

Related U.S. Application Data

(63) Continuation of application No. 09/878,097, filed on Jun. 8, 2001, now Pat. No. 7,210,099, which is a (Continued)

(51) **Int. Cl.**

G06F 17/00 (2019.01)
G06F 40/143 (2020.01)
G06F 9/451 (2018.01)
G06F 16/958 (2019.01)
G06F 16/957 (2019.01)
G06F 40/134 (2020.01)
G06F 40/103 (2020.01)
G06F 3/0488 (2013.01)
H04L 29/08 (2006.01)
G06F 16/95 (2019.01)

(Continued)

(52) **U.S. Cl.**

CPC **G06F 40/143** (2020.01); **G06F 3/0488** (2013.01); **G06F 9/451** (2018.02); **G06F 16/9577** (2019.01); **G06F 16/986** (2019.01); **G06F 40/103** (2020.01); **G06F 40/134**

(2020.01); **H04L 67/04** (2013.01); **H04L 67/2823** (2013.01); **G06F 3/0481** (2013.01); **G06F 16/95** (2019.01); **G06F 16/957** (2019.01); **G06F 16/958** (2019.01); **G06F 16/9537** (2019.01); **G06F 16/9574** (2019.01); **G06F 2203/04806** (2013.01); **G06F 2212/1048** (2013.01); **G06Q 30/02** (2013.01); **G09G 5/24** (2013.01); **G09G 2340/0407** (2013.01); **G09G 2340/0492** (2013.01); **H04L 67/02** (2013.01)

(58) **Field of Classification Search**

None

See application file for complete search history.

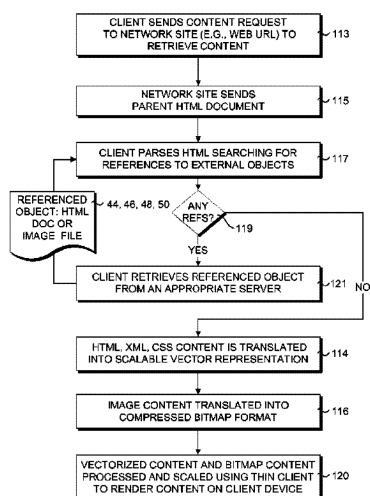
(56) **References Cited**

To view the complete listing of prior art documents cited during the proceeding for Reexamination Control Number 90/009,995, please refer to the USPTO's public Patent Application Information Retrieval (PAIR) system under the Display References tab.

Primary Examiner — Adam L Basehoar(57) **ABSTRACT**

Mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes. The mobile devices employ software-based processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Display lists may also be employed to provide further enhancements in rendering speed.

At the time of issuance and publication of this certificate, the patent remains subject to pending reexamination control numbers 95/000,635 and 95/002,126 filed May 23, 2011 and Aug. 31, 2012 respectively. The claim content of the patent may be subsequently revised if a reexamination certificate issues from the reexamination proceedings.



Related U.S. Application Data

continuation-in-part of application No. 09/828,511,
filed on Apr. 7, 2001, now abandoned.

- (60) Provisional application No. 60/217,345, filed on Jul.
11, 2000, provisional application No. 60/211,019,
filed on Jun. 12, 2000.

- (51) **Int. Cl.**

<i>G06F 16/9537</i>	(2019.01)
<i>G06F 3/0481</i>	(2013.01)
<i>G06Q 30/02</i>	(2012.01)
<i>G09G 5/24</i>	(2006.01)

EX PARTE
REEXAMINATION CERTIFICATE

THE PATENT IS HEREBY AMENDED AS 5
INDICATED BELOW.

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

The patentability of claim **29** is confirmed. 10

Claims **30, 31, 40, 41, 43, 52, 55, 59, 72** and **75** were
previously cancelled.

Claims **1-28, 32-39, 42, 44-51** and **69** are cancelled.

Claims **53-54, 56-58, 60-68, 70-71, 73-74** and **76-88** were 15
not reexamined.

* * * * *



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/009,995	05/29/2012	7,831,926	161539-0005(926PO-2)	3089
59860	7590	11/15/2021		
LAW OFFICE OF R. ALAN BURNETT			EXAMINER	
4108 131ST AVE. SE			BASEHOAR, ADAM L	
BELLEVUE, WA 98006				
			ART UNIT	PAPER NUMBER
			3992	
			MAIL DATE	DELIVERY MODE
			11/15/2021	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

POLSINELLI PC
(NDQ REEXAMINATION GROUP)
1000 LOUISIANA STREET
FIFTY-THIRD FLOOR
HOUSTON, TX 77002

***EX PARTE* REEXAMINATION COMMUNICATION TRANSMITTAL FORM**

REEXAMINATION CONTROL NO. 90/009,995 .

PATENT UNDER REEXAMINATION 7831926 .

ART UNIT 3992 .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Notice of Intent to Issue Ex Parte Reexamination Certificate	Control No. 90/009,995	Patent Under Reexamination 7831926	
	Examiner Adam L Basehoar	Art Unit 3992	AIA Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

1. ☒ Prosecution on the merits is (or remains) closed in this *ex parte* reexamination proceeding. This proceeding is subject to reopening at the initiative of the Office or upon petition. *Cf.* 37 CFR 1.313(a). A Certificate will be issued in view of
 - (a) ☐ Patent owner's communication(s) filed: _____.
 - (b) ☐ Patent owner's failure to file an appropriate timely response to the Office action mailed: _____.
 - (c) ☐ Patent owner's failure to timely file an Appeal Brief (37 CFR 41.31).
 - (d) ☒ The decision on appeal by the ☒ Board of Patent Appeals and Interferences ☐ Court dated 01 May 2020
 - (e) ☒ Other: Decision by the United States Court of Appeals for the Federal Circuit (CAFC) on 05/14/2021.
2. The Reexamination Certificate will indicate the following:
 - (a) Change in the Specification: ☐ Yes ☒ No
 - (b) Change in the Drawing(s): ☐ Yes ☒ No
 - (c) Status of the Claim(s):
 - (1) Patent claim(s) confirmed: 29.
 - (2) Patent claim(s) amended (including dependent on amended claim(s)): _____
 - (3) Patent claim(s) canceled: 1-28,32-39,42,44-51 and 69.
 - (4) Newly presented claim(s) patentable: _____.
 - (5) Newly presented canceled claims: _____.
 - (6) Patent claim(s) ☐ previously ☐ currently disclaimed: _____
 - (7) Patent claim(s) not subject to reexamination: 53-54,56-58,60-68,70-71,73-74 and 76-88.
3. ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.
4. ☒ Note the attached statement of reasons for patentability and/or confirmation. Any comments considered necessary by patent owner regarding reasons for patentability and/or confirmation must be submitted promptly to avoid processing delays. Such submission(s) should be labeled: "Comments On Statement of Reasons for Patentability and/or Confirmation."
5. ☐ Note attached NOTICE OF REFERENCES CITED (PTO-892).
6. ☒ Note attached LIST OF REFERENCES CITED (PTO/SB/08 or PTO/SB/08 substitute).
7. ☐ The drawing correction request filed on _____ is: ☐ approved ☐ disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. § 119(a)-(d) or (f).
 - a) ☐ All b) ☐ Some* c) ☐ None of the certified copies have
 - ☐ been received.
 - ☐ not been received.
 - ☐ been filed in Application No. _____.
 - ☐ been filed in reexamination Control No. _____.
 - ☐ been received by the International Bureau in PCT Application No. _____.

* Certified copies not received: _____.
9. ☒ Note attached Examiner's Amendment.
10. ☐ Note attached Interview Summary (PTO-474).
11. ☒ Other: See Continuation Sheet.

All correspondence relating to this reexamination proceeding should be directed to the **Central Reexamination Unit** at the mail, FAX, or hand-carry addresses given at the end of this Office action.

/ADAM L BASEHOAR/ Primary Examiner, Art Unit 3992	
--	--

cc: Requester (if third party requester)

Continuation of 11. Other: Claims 30, 31, 40, 41, 43, 52, 55, 59, 72 and 75 were held invalid in a final decision by the Court of Appeals for the Federal Circuit. *Softview LLC v. Kyocera Corp. and Motorola Mobility LLC.*, No. 2014-1599 (Fed. Cir. Feb. 9, 2015). Thereafter, claims 30, 31, 40, 41, 43, 52, 55, 59, 72, and 75 were cancelled in the Inter Partes Review Certificate issued on 01/15/2016. Thus, claims 30, 31, 40, 41, 43, 52, 55, 59, 72 and 75 are withdrawn from consideration in the reexamination.

DETAILED ACTION

1. This Office Action addresses claims 1-29, 32-39, 42, 44-51, and 69 of United States Patent Number 7,831,926 B2 (hereafter “Rohrbaugh ‘926”), for which it has been determined in the Order Granting *Ex Parte* Reexamination (hereafter the “Order”) mailed 07/17/2012 that a substantial new question of patentability (SNQ) was raised in the Request for *Ex Parte* reexamination filed on 05/29/2012 (hereafter the “Request”).

The Examiner notes that claims 53, 54, 56-58, 60-68, 70, 71, 73, 74 and 76-88 of the Rohrbaugh ‘926 patent are not subject to this reexamination. Additionally, claims 30, 31, 40, 41, 43, 52, 55, 59, 72 and 75 of the Rohrbaugh ‘926 patent were previously canceled (via the IPR2013-00004 & IPR2013-00257 proceedings) and are thus withdrawn from consideration in the instant reexamination.

This Office Action is a Notice of Intent to Issue *Ex Parte* Reexamination Certificate (hereafter “NIRC”).

2. On 05/01/2020 a decision (hereafter the “PTAB Decision”) on appeal by the Patent Trial and Appeal Board (“PTAB”) was rendered (PTAB Decision: p. 2: “We affirm-in-part”) and mailed affirming the Examiner’s decision rejecting claims 1-28, 32-39, 42, and 44-51 (PTAB Decision: p. 28: “rejecting claims 1-28, 32-39, 42, and 44-51...is affirmed”) and reversing the Examiner’s decision rejecting claim 29 (PTAB Decision: p. 28: “rejecting claim 29...is reversed”). The Examiner notes that full faith and credit has been given to the review/decision rendered by the PTAB (see: MPEP 1214.04). On 5/14/2021 a decision (hereafter the “CAFC Decision”) by the United States Court of Appeals for the Federal Circuit (“CAFC”) was rendered

(CAFC Decision: p. 2: “AFFIRMED. See Fed. Cir. R. 36”) and mailed affirming the PTAB Decision. The time period for seeking further review of said decisions has expired.

3. On 05/18/2021 the Patent Owner (“PO”) requested immediate issuance of an *Ex Parte* reexamination certificate and filed a corresponding Amendment to the Claims (hereafter the “Amendment”) to rewrite dependent claim 29 in independent form. PO noted that since “claim 1 will be cancelled, claim 29 needs to be re-written in independent form.” The Examiner notes that PO’s Amendment will not be entered because 1) rewritten claim 29 would not maintain identical claim text with corresponding claim 29 in the currently pending 95/000,635 and 95/002,126 *Inter Partes* reexamination proceedings, and 2) the appropriate approach to take in this situation is to simply cancel unamended independent claim 1 and confirm unamended dependent claim 29 (see: MPEP 2278(II)(D) in view of 1216.01(I)(B)). An Examiner’s Amendment below reflects the proper procedure.

Claim Status Summary

4. The current status of the claims of the Rohrabough ‘926 patent in this *Ex Parte* Reexamination proceeding are as follows:

- Claim 29 is confirmed
- Claims 1-28, 32-39, 42, and 44-51 are canceled below via the Examiner’s Amendment
- Claim 69 was canceled during the prosecution of the instant 90/009,995 reexamination proceeding

- Claims 53, 54, 56-58, 60-68, 70, 71, 73, 74 and 76-88 are not subject to the instant reexamination
- Claims 30, 31, 40, 41, 43, 52, 55, 59, 72 and 75 were previously canceled and are thus withdrawn from consideration in the instant reexamination

Information Disclosure Statement

5. Regarding Information Disclosure Statement (IDS) submissions, MPEP 2256 recites the following: “Where patents, publications, and other such items of information are submitted by a party (patent owner or requester) in compliance with the requirements of the rules, the requisite degree of consideration to be given to such information will be normally limited by the degree to which the party filing the information citation has explained the content and relevance of the information. The initials of the examiner placed adjacent to the citations on the form PTO/SB/08A and 08B or its equivalent, without an indication to the contrary in the record, do not signify that the information has been considered by the examiner any further than to the extent noted above.”

Accordingly, the IDS submission filed by Patent Owner on 07/18/2018 has been considered by the Examiner only with the scope required by MPEP 2256, unless otherwise noted.

Examiner’s Amendment

6. An Examiner’s Amendment to the record appears below. As noted above, rejections covering claims 1-28, 32-39, 42, and 44-51 have been affirmed in the PTAB Decision of 05/01/2020 and the CAFC Decision of 05/14/2021. Accordingly, claims 1-28, 32-39, 42, and

44-51 are now being canceled. The changes made by this Examiner's Amendment will be reflected in the reexamination certificate to issue in due course.

Please cancel claims 1-28, 32-39, 42, and 44-51 as follows:

Claims 1-28 (Canceled)

Claims 32-39 (Canceled)

Claim 42 (Canceled)

Claims 44-51 (Canceled)

Statement of Reasons for Patentability and/or Confirmation

7. Claim 29 is confirmed.

The following is an Examiner's statement of reasons for patentability and/or confirmation of claim 29 found confirmed in this reexamination proceeding. Claims 29 is confirmed over the prior art that was explained in the Request and determined to raise a substantial new question of patentability in the Order granting reexamination and over the prior art that was applied and discussed by the Examiner in the present reexamination proceeding because of the following:

Please note the detailed analysis on pages 23-27 of the PTAB Decision. Said analysis is hereby incorporated by reference.

Any comments considered necessary by PATENT OWNER regarding the above statement must be submitted promptly to avoid processing delays. Such submission by the Patent Owner should be labeled: "Comments on Statement of Reasons for Patentability and/or Confirmation" and will be placed in the reexamination file.

Conclusion

8. All correspondence relating to this *Ex Parte* reexamination proceeding should be directed as follows:

By U.S. Postal Service Mail to:

Mail Stop Ex Parte Reexam
ATTN: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to:

(571) 273-9900
Central Reexamination Unit

By hand to:

Customer Service Window
Randolph Building
401 Dulany St.
Alexandria, VA 22314

By EFS-Web:

Registered users of EFS-Web may alternatively submit such correspondence via the electronic filing system EFS-Web, at

<https://efs.uspto.gov/efile/myportal/efs-registered>

EFS-Web offers the benefit of quick submission to the particular area of the Office that needs to act on the correspondence. Also, EFS-Web submissions are “soft scanned” (i.e., electronically uploaded) directly into the official file for the reexamination proceeding, which offers parties the opportunity to review the content of their submissions after the “soft scanning” process is complete.

Any inquiry concerning this communication or earlier communications from the Reexamination Legal Advisor or Examiner, or as to the status of this proceeding, should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

/Adam L Basehoar/
Primary Examiner, Art Unit 3992

Conferees:

/JOSHUA D CAMPBELL/
Primary Examiner, Art Unit 3992

/ALEXANDER J KOSOWSKI/
Supervisory Patent Examiner, Art Unit 3992

NOTE: This disposition is nonprecedential.

**United States Court of Appeals
for the Federal Circuit**

IN RE: SOFTVIEW LLC,
Appellant

2020-1971

Appeal from the United States Patent and Trademark
Office, Patent Trial and Appeal Board in No. 90/009,995.

JUDGMENT

ALAN BURNETT, Law Office of R. Alan Burnett, Bellevue, WA, argued for appellant.

MICHAEL S. FORMAN, Office of the Solicitor, United States Patent and Trademark Office, Alexandria, VA, argued for appellee Andrew Hirshfeld. Also represented by THOMAS W. KRAUSE, FARHEENA YASMEEN RASHEED.

THIS CAUSE having been heard and considered, it is

ORDERED and ADJUDGED:

PER CURIAM (NEWMAN, REYNA, and TARANTO, *Circuit Judges*).

AFFIRMED. See Fed. Cir. R. 36.

ENTERED BY ORDER OF THE COURT

May 14, 2021
Date

/s/ Peter R. Marksteiner
Peter R. Marksteiner
Clerk of Court

**United States Court of Appeals
for the Federal Circuit**

In re: SOFTVIEW LLC,
Appellant

2020-1971

Appeal from the United States Patent and Trademark
Office, Patent Trial and Appeal Board in No. 90/009,995.

MANDATE

In accordance with the judgment of this Court, entered
May 14, 2021, and pursuant to Rule 41 of the Federal Rules
of Appellate Procedure, the formal mandate is hereby
issued.

FOR THE COURT

July 6, 2021

/s/ Peter R. Marksteiner

Peter R. Marksteiner
Clerk of Court



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/009,995	05/29/2012	7,831,926	161539-0005(926PO-2)	3089
29000	7590	05/01/2020		
Irell & Manella LLP 840 Newport Center Dr., Ste. 400 Newport Beach, CA 92677			EXAMINER BASEHOAR, ADAM L	
			ART UNIT	PAPER NUMBER
			3992	
			MAIL DATE	DELIVERY MODE
			05/01/2020	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte SOFTVIEW L.L.C.
Appellant and Patent Owner

Appeal 2019-001127
Reexamination Control 90/009,995
Patent 7,831,926 B2
Technology Center 3900

Before JOHN A. JEFFERY, ERIC B. CHEN, and BRIAN J. McNAMARA,
Administrative Patent Judges.

CHEN, *Administrative Patent Judge.*

DECISION ON APPEAL

This is an appeal under 35 U.S.C. §§ 134(b) and 306 from the final rejection of claims 1–29, 32–39, 42, and 44–51. Claim 69 was cancelled during the reexamination proceeding. Claims 53, 54, 56–58, 60–68, 70, 71, 73, 74, and 76–88 were not subject to reexamination. Claims 30, 31, 40, 41, 43, 52, 55, 59, 72 and 75 were cancelled in a previous *inter partes* review proceeding. We have jurisdiction under §§ 134(b) and 306.

We affirm-in-part.

STATEMENT OF THE CASE

Reexamination Proceedings

A request for *ex parte* reexamination of U.S. Patent No. 7,831,926 B2 (“the ’926 patent”) was filed on April 13, 2012, and assigned Reexamination Control No. 90/009,995. The ’926 patent, entitled “Scalable Display of Internet Content on Mobile Devices,” issued November 9, 2010 to Gary B. Rohrabough and Scott A. Sherman, based on Application No. 11/738,486, filed April 21, 2007.

The ’926 patent is said to be a continuation of Application No. 09/878,097, filed June 8, 2001, now U.S. Patent No. 7,210,099, which is a continuation-in-part of Application No. 09/828,511, filed April 7, 2001, now abandoned. The ’926 patent is also said to claim priority to U.S. Provisional Application No. 60/211,019, filed June 12, 2000, and U.S. Provisional Application No. 60/217,345, filed July 11, 2000.

The ’926 patent is said to be assigned to SoftView, L.L.C., the real party in interest. (Appeal Br. 1.)

Patent Owner's Invention

Patent Owner's invention relates to mobile devices that are enabled to support resolution-independent scalable display of Internet (Web) content, allowing Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes. (Abstract.)

Related Litigation

The '926 patent has been asserted in *SoftView LLC v. Apple Inc.*, No. 1:10-cv-00389 (D. Del. May 10, 2010) and *SoftView LLC v. HTC Corp.*, No. 1:12-cv-00984 (D. Del. July 26, 2012). (Appeal Br. 1.)

The '926 patent was also subject to *inter partes* review in *Softview LLC v. Kyocera Corp. and Motorola Mobility LLC*, Nos. IPR2013-00004 and IPR2013-00257 (PTAB Mar. 27, 2014), *aff'd*, No. 2014-1599 (Fed. Cir. Feb. 9, 2015).

The Claimed Invention

Claims 1, 2, and 29 are exemplary, with disputed limitations in italics:

1. A mobile device, comprising:
 - a processor,
 - a wireless communications device operatively coupled to the processor, to facilitate communication with a network via which Web content may be accessed;
 - a touch-sensitive display;
 - a memory, operatively coupled to the processor; and
 - storage means, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile phone to perform operations including,

enabling a user to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving HTML-based Web content associated with the Web page;

translating the HTML-based Web content to produce scalable vector-based page layout information;

employing the scalable vector-based page layout information and/or data derived therefrom to,

render a view of at least a portion of the Web page on the touch-sensitive display using a first scale factor; and

re-render the Web page in response to associated user inputs to enable a user to iteratively zoom in and out views of the Web page on the display while preserving the original page layout, functionality, and design of the content on the Web page defined by the HTML based Web content,

wherein preservation of the functionality defined by the HTML-based content includes preservation of hyperlink functionality.

2. The mobile device of claim 1, wherein the device comprises a mobile phone.

29. The mobile device of claim 1, wherein the HTML-based Web content *includes cascading style sheet content defining layout and presentation attributes for the Web page.*

The Rejections

A. Claims 1 and 3–28 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the Pad++¹ and SVF.²

B. Claims 2, 32–39, 42, and 44–51 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pad++, SVF, Nokia (Nokia Press Release, Nokia Unveils World’s First All-In-One Communicator for the Americas, (Sept. 19, 1996), and Palm (Palm Press Release, 3Com Announces the Palm VII Connected Organizer, the First Handheld Solution for Out-of-the-box Wireless Internet Access (Dec. 2, 1998)).

C. Claims 2, 32–39, 42, and 44–51 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pad++, SVF, and Newton (Apple Computer, Inc., News Release, Apple Outlines Plethora of Newton Wireless

¹ “Pad++” refers collectively to the following references: (i) Benjamin B. Bederson & James D. Hollan, Pad++: *A Zoomable Graphical Interface System*, CHI ‘95 MOSAIC OF CREATIVITY (1995); (ii) Benjamin B. Bederson & George W. Furnas, *Space-Scale Diagrams: Understanding Multiscale Interfaces*, CHI ‘95 PROCEEDINGS (1995); (iii) Benjamin B. Bederson et al., *A Zooming Web Browser*, 2667 SPIE 260–71 (1996); (iv) Ben Bederson & Jon Meyer, *Implementing a Zooming User Interface: Experience Building Pad++*, 28 SOFTWARE–PRACTICE & EXPERIENCE 1101–35 (1998); (v) Benjamin B. Bederson et al., *Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics*, 7 J. VISUAL LANGUAGES & COMPUTING 3–31 (1996); (vi) Pad++ Reference Manual Version 0.2.7 (1996); (vii) Pad++ Programmer’s Guide Version 0.2.7 (1996); and (viii) Jonathan Meyer, A Brief Tour Through Pad++, *available at* <http://www.cs.umd.edu/hcil/pad++/tour/> (“Pad++ Tour”).

² “SVF” (Simple Vector Format) refers collectively to: (i) Specification for the Simple Vector Format v. 1.1 (1995); (ii) “New CAD System Works With AutoCAD Drawings Without Translation” (1996); and (iii) “Bring New CAD Viewing Power to the Internet” (1996).

Communications Solutions at MessagePad 120 Launch (Jan. 30, 1995)).

D. Claim 29 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Pad++, SVF, and Bray (Tim Bray, *XML support in IE5*, XLM.com (Mar. 18, 1999)).

E. Claims 1 and 3–28 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pad++ and VML (Brian Matthews et al., *Vector Markup Language (VML)*, World Wide Web Consortium Note, Note-VML-19980513 (May 13, 1998)).

F. Claims 2, 32–39, 42, and 44–51 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pad++, VLM, Nokia, and Palm.

G. Claims 2, 32–39, 42, and 44–51 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pad++, VLM, and Newton.

H. Claim 29 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Pad++, VML, and Bray.

I. Claims 12, 14, 21, 42, and 51 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Pad++, SVF, and various combinations of Nokia, Palm, Newton, and VML.

Patent Owner relied upon the following in rebuttal to the Examiner's rejections:

Declaration under 37 C.F.R. § 1.132 of Michael L. Howard, November 16, 2016 (Ex. A).

Declaration under 37 C.F.R. § 1.132 of Wolf (formerly Scott Collins), dated July 15, 2017 (Ex. B).

Declaration under 37 C.F.R. § 1.132 of Wolf (formerly Scott Collins), dated July 24, 2017 (Ex. D).

Declaration under 37 C.F.R. § 1.132 of Glenn Reinman, Ph.D., dated July 17, 2013 (Ex. F).

Deposition of Scott Forstall, dated May 3, 2013, *SoftView LLC v. Apple Inc.*, No. 1:10-cv-00389 (D. Del. May 10, 2010) (Ex. H).

Deposition of Benjamin B. Bederson, dated February 13, 2013, *SoftView LLC v. Apple Inc.*, No. 1:10-cv-00389 (D. Del. May 10, 2010) (Ex. G).

ANALYSIS

§ 103 Rejection—Pad++ and SVF

Claims 1 and 3–28

We are unpersuaded by Patent Owner’s arguments (Appeal Br. 17–20; *see also* Reply Br. 10–12) that the combination of Pad++ and SVF would not have rendered obvious independent claim 1, which includes the limitation “re-render the Web page . . . while preserving the original page layout, functionality, and design of the content on the Web page defined by the HTML based Web content.”

The Examiner found that the zoomable HTML web browser of Pad++ Tour,³ which includes displaying HTML documents, corresponds to the limitation “re-render the Web page . . . while preserving the original page layout, functionality, and design of the content on the Web page defined by the HTML based Web content, wherein preservation of the functionality defined by the HTML-based content includes preservation of hyperlink functionality.” (Ans. 13.) We agree with the Examiner’s findings.

Pad++ Tour relates to a computer screen having a zoomable surface, “[f]or example, several million pages of text could be fit on the surface by reducing it sufficiently in scale, making any number of on-line information

³ The Examiner cited to Jonathan Meyer, *A Brief Tour Through Pad++*.

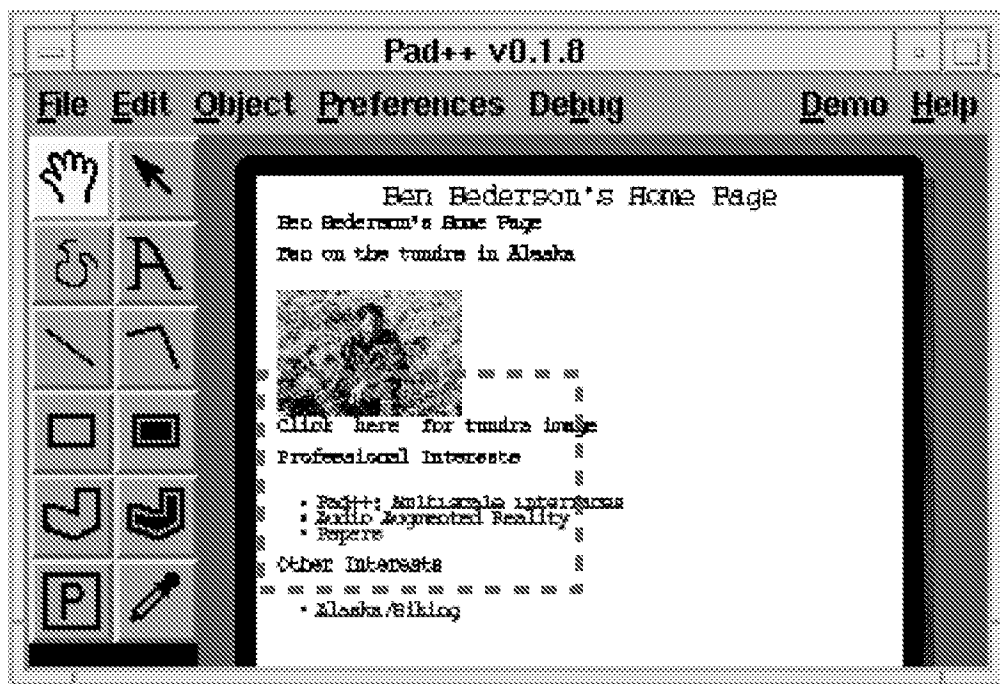
services, encyclopedias, etc., directly available.” (P. 2.) In one example of zooming a user interface, entitled “An HTML browser,” Pad++ Tour explains the following:

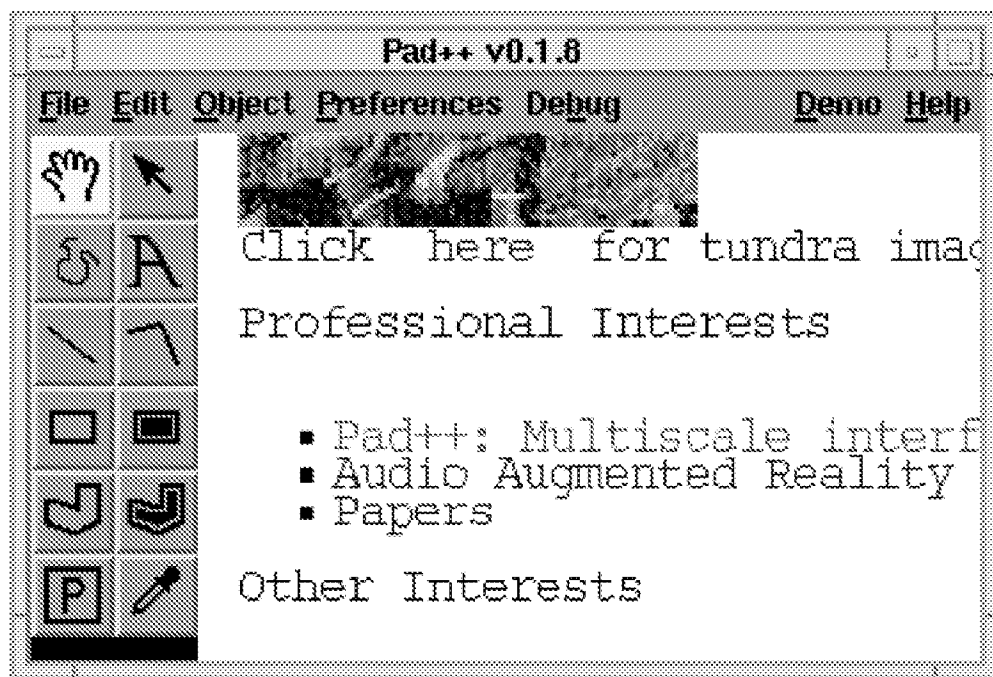
A zooming version of Mosaic and Netscape? This is not such a strange idea. The third example Pad++ application we will look at is a simple Pad++ web browser.

Although the web browser we show here is still fairly primitive, you could imagine a much more advanced version. Does the idea of using a zoomable Web browser excite you?

(P. 86.)

In two unreferenced figures, reproduced below, Pad++ Tour depicts screen snapshots of: (i) a regularly sized HTML document entitled “Ben Bederson’s Home Page” (annotated with a dashed rectangle indicating the zoomed-in region); and (ii) the zoomed-in view of the same HTML document:



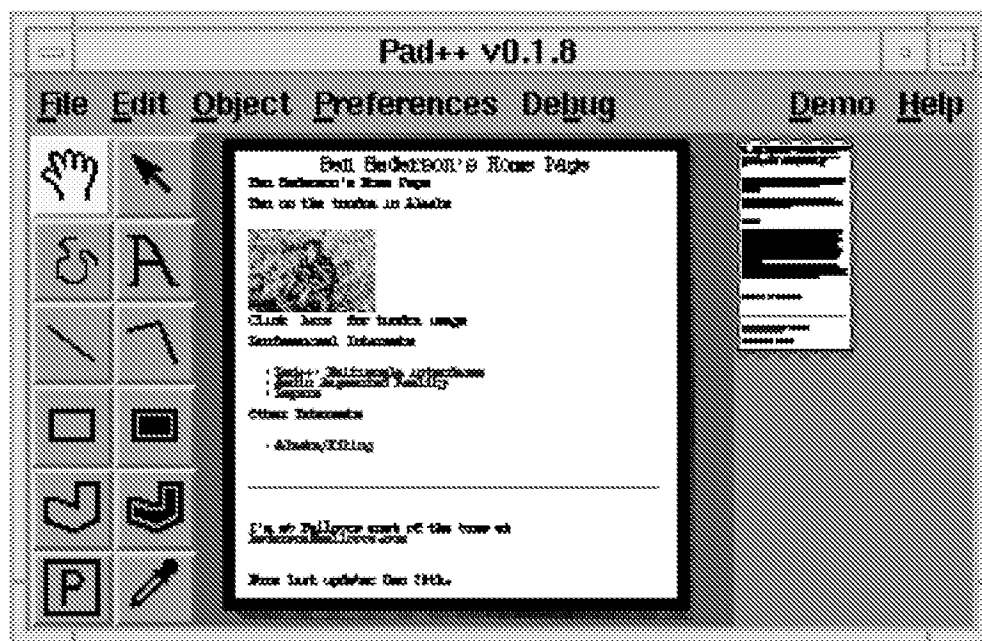


Moreover, Pad++ Tour explains the following:

When you follow a link, the relevant document is loaded into Pad++ and placed on the surface to the right of the original document, at a smaller scale. Here you can see the 'Pad++: Multiscale interfaces' document loaded beside the home page.

The Pad++ HTML browser will lay out sub-documents in two columns next to the parent document. Because Pad++ is zoomable, there is always enough space between those two columns for placing further documents reached from those sub-documents!

(P. 91.) In a third unreferenced figure, reproduced below, Pad++ Tour depicts screen snapshots of the "Pad++: Multiscale interfaces" HTML document (annotated with a dashed rectangle) loaded next to the regularly sized HTML document entitled "Ben Bederson's Home Page":



Because the first and third unreferenced figures in Pad++ Tour depict a regularly sized HTML document entitled “Ben Bederson’s Home Page,” Pad++ Tour teaches the limitation “the original page layout, functionality, and design of the content on the Web page defined by the HTML based Web content.” Moreover, because the second unreferenced figure in Pad++ Tour depicts a zoomed-in view of the regularly sized HTML document entitled “Ben Bederson’s Home Page,” and the third unreferenced figure in Pad++ Tour depicts the HTML document associated with the “Pad++: Multiscale interfaces” link is displayed, Pad++ Tour teaches the limitation “re-render the Web page . . . while preserving the original page layout, functionality, and design of the content on the Web page defined by the HTML based Web content, wherein preservation of the functionality defined by the HTML-based content includes preservation of hyperlink functionality.”

Patent Owner argues that “[t]here is no evidence that the views shown in Figures 1 and 2 of Bederson preserves the HTML-based Web content defining the original page layout, functionality, and design of content on the Web page, as the HTML-based Web content for these pages is not available”⁴ and “what is known is Pad++ was not capable of rendering the vast majority of HTML elements, such as basic layout elements including tables, forms, divs and spans.” (Appeal Br. 13; *see also* Reply Br 10.) To support this argument, Patent Owner references paragraphs 25–34 of the Reinman Declaration (Ex. F). (Appeal Br. 13.) However, the Examiner also cited to Pad++ Tour for teaching the limitation “while preserving the original page layout, functionality, and design of the content on the Web page defined by the HTML based Web content.”

Moreover, one relevant portion of the Reinman Declaration (Ex. F) states the following:

Section 3 of the Brief Tour only shows how the web page looks like after it has been rendered by Pad++. The Brief Tour does not show the original web page before being rendered by Pad++ to determine whether the original page layout, functionality, and design was preserved. . . . One of ordinary skill in the art cannot determine the original page layout, functionality, and design of the original web page from the Brief Tour

(¶ 23.) However, as discussed previously, the first and third unreferenced figures in Pad++ Tour depict a regularly sized HTML document entitled “Ben Bederson’s Home Page,” and accordingly, Pad++ Tour teaches the

⁴ Patent Owner references Figure 1 and 2 of Benjamin B. Bederson et al., *A Zooming Web Browser*, 2667 SPIE 260–71 (1996).

limitation “the original page layout, functionality, and design of the content on the Web page defined by the HTML based Web content.”

Patent Owner further argues the following:

Pad++ Tour suggests that a zooming version of Mosaic or Netscape is not a strange idea. This statement is a mere suggestion of possibly creating a zoomable browser using Mosaic or Netscape at the time the Tour Web pages were created. Based on the version v.0.1.8 shown in the Tour images, this predates the version of Pad++ used in Bederson-4 (v0.2.3), meaning the Tour Web pages (including the foregoing statement) were originally posted sometime in 1995 or earlier. Moreover, this statement is not a teaching that the Pad++ browser could be (or could have been) combined with an existing browser such as Netscape or Mosaic (in 1995), nor provides any detail on any level to how Pad++ could have been combined with Mosaic or Netscape.

(Appeal Br. 17 (emphases omitted); *see also* Reply Br. 10–11.) In particular, Patent Owner argues the following:

Patent Owner has introduced substantial evidence a PHOSITA would not have been able to implement a zoomable version of Netscape at the time of the invention without undue experimentation, there would have been no motivation to do so, and no reasonable expectation of success. The developers of Mozilla themselves, who had skill levels well beyond a PHOSITA, were not able to implement a zooming version of a desktop browser (Mozilla Firefox, which superseded Netscape Navigator) until 2007, even though this was initially identified as a feature in 1999 and Mozilla was motivated to implement the feature. With further respect to Mosaic at the time of the invention it had not been updated for approximately 4 years, and didn’t even support HTML 3 or 3.2, much less HTML 4 or CSS and had been superseded by Netscape and Internet Explorer; accordingly, there would have been no motivation at the time of the invention to implement a zoomable version of Mosaic.

(Appeal Br. 17 (citations and footnotes omitted); *see also* Reply Br. 10–11.) However, as discussed previously, Pad++ Tour explains that for “[t]he third example Pad++ application we will look at is a simple Pad++ web browser” and “[a]lthough the web browser we show here is still fairly primitive, you could imagine a much more advanced version.” (P. 86.) Even if Patent Owner is correct that a zoomable version of Netscape or Mosaic could not have been technically implemented, such evidence does not necessarily mean that one of ordinary skill in the art would have excluded *all* web browsers from the implementation of a zoomable browser.

Therefore, we agree with the Examiner that the combination of Pad++ and SVF would have rendered obvious independent claim 1, which includes the limitation “while preserving the original page layout, functionality, and design of the content on the Web page defined by the HTML based Web content.”

We are also unpersuaded by Patent Owner’s arguments (Appeal Br. 20; *see also* Reply Br. 10–11) that the combination of Pad++ and SVF would not have rendered obvious independent claim 1, which includes the limitation “translating the HTML-based Web content to product scalable vector-based page layout information.”

The Examiner found that the multiple objects of Pad++ Tour, residing at specific x, y, zoom coordinates within a space, correspond to the limitation “translating the HTML-based Web content to product scalable vector-based page layout information.” (Final Act. 7–8 (citing Request for Reexamination 463–64, filed April 13, 2012).) Alternatively, the Examiner found the Simple Vector Format plug-ins feature for scalable vector graphics

of SVL corresponds to the limitation “translating the HTML-based Web content to product scalable vector-based page layout information.”

(Ans. 17–18.) We agree with the Examiner’s findings with respect to Pad++ Tour.

Pad++ Tour explains the following:

The Pad++ surface as an infinite zoomable space, with objects residing at specific x, y, zoom coordinates within that space.

Sometimes, however, you want to be able to compare two distant locations, or to look at a zoomed out view of a document together with a close up view.

(P. 97.) In other words, Pad++ explains that: (i) an (x,y) coordinate is associated with an object for a “zoomed out view”; and (ii) an (x, y) zoom coordinate is associated with the object in “a close up view.” Because Pad++ explains that there are two sets of (x,y) coordinates (i.e., a vector), one set for the “zoomed out view” and one set for “a close up view,” Pad++ teaches the limitation “translating the HTML-based Web content to produce scalable vector-based page layout information.”

Patent Owner argues that following:

Pad++ uses vector-based drawings facilities in X Windows. The SVF plugin was designed to enable SVF content that was embedded in a Web page (such as a CAD drawings or vector graphic) to be rendered by a browser. . . . This combination makes no sense, and there is nothing in any of the Pad++ references or the SVF references that suggests it.

(Appeal Br. 20.) However, even if Patent Owner is correct that the combination of Pad++ and SVF is improper, the Examiner cited SVF as a cumulative reference for teaching the limitation “translating the HTML-

based Web content to product scalable vector-based page layout information.” (Ans. 17–18.)

Therefore, we agree with the Examiner that the combination of Pad++ and SVF would have rendered obvious independent claim 1, which includes the limitation “translating the HTML-based Web content to product scalable vector-based page layout information.”

Accordingly, we sustain the rejection of independent claim 1 under 35 U.S.C. § 103(a). Claims 3–28 depend from independent claim 1, and Patent Owner had not presented any additional substantive arguments with respect to these claims. Therefore, we sustain the rejection of claims 3–28 under 35 U.S.C. § 103(a) for the same reasons discussed with respect to independent claim 1.

§ 103 Rejection—Pad++, SVF, Nokia, and Palm

We are unpersuaded by Patent Owner’s arguments (Appeal Br. 25–41; *see also* Reply Br. 16–30) that the combination of Pad++, SVF, Nokia, and Palm would not have rendered obvious dependent claim 2, which includes the limitation “wherein the device comprises a mobile phone.”

The Examiner found that the zoomable PDA (Personal Digital Assistants) of Pad++ Tour corresponds to the limitation “wherein the device comprises one of a handheld device or a palm-held device.” (Ans. 21–22.) We agree with the Examiner’s findings.

Pad++ Tour explains the following:

Pad++ could also be viewed as a way to implement an alternative windowing system based on zooming. Your whole desktop could be zoomable. This seems especially attractive

for systems which have small screens, such as handheld computers (i.e. PDA's).

(P. 141.) One relevant definition of a “PDA” is “[a] lightweight palmtop computer designed to provide specific functions like personal organization (calendar, note taking, database, calculator, and so on) as well as communications.”⁵ Accordingly, because a PDA is defined as a palmtop computer having communication functions, a mobile phone is an obvious variant. Moreover, because Pad++ Tour explains that the zoomable windowing system is attractive for systems having small screens, such as PDAs or mobile phones, Pad++ Tour teaches the limitation “wherein the device comprises a mobile phone.”

First, Patent Owner argues that “the Examiner has not presented any evidence that the ‘the Pad++ references specifically teach wherein the Pad++ browser was designed to operate on PDAs’” and “[t]o the contrary, the Pad++ references provide a mountain of evidence that the Pad++ browser was specifically designed to operate on a desktop computer or workstation.” (Appeal Br. 25.) Contrary to Patent Owner’s arguments, Pad++ Tour expressly states that “[y]our whole desktop could be zoomable” and “[t]his seems especially attractive for systems which have small screens, such as handheld computers (i.e. PDA’s).” (P. 141.)

Second, Patent Owner argues the following:

Pad++ was never implemented on a mobile device, PDA, or mobile phone. . . . Ben Bederson acknowledged in his deposition, “I didn’t actually do it” (run Pad++ on a PDA), and, “Certainly the code itself didn’t run on PDAs.” The Examiner has presented no evidence that Pad++ was implemented on any

⁵ MICROSOFT® COMPUTER DICTIONARY 396 (5th ed. 2002).

PDA, nor any evidence of anyone ever attempting to implement Pad++ on a PDA, or that the references provide an implementation of the Pad++ browser on a mobile device.

(Appeal Br. 25; *see also* Reply Br. 16–17.) Similarly, Patent Owner argues that:

Ben Bederson testified that (1) he never implemented Pad++ on a PDA; (2) the Pad++ code did not run on a PDA; (3) he was not familiar with the technical capabilities and amount of memory available by PDAs at the time (1995 or 1999); and (4) the target platform was any computer that could run Linux and X graphics system

(Appeal Br. 26.) However, the Bederson Deposition⁶ also states the following:

Q. Could Pad++ have run on PDAs?

A. At that time, in 1995? I think technically, *it would have been a challenge to work well*, that is with the smooth zooming one of the goals of Pad++, because that was a much slower processor that would be used in those PDAs.

(Ex. G, 180:19–25 (emphasis added).)

. . .

So my answer would be conceptually, yes, it should run. It's a general purpose computer system. Practically speaking, it could well be difficult to make it actually run because of limitations, limitations of those devices at the time, but I don't have any more information to know how easy or hard it would have been to get it to run.

(Ex. G, 181: 9–16 (emphasis added).) Accordingly, while Professor Bederson testifies that while “it would have been a challenge to work well,”

⁶ Professor Bederson, a computer science faculty member, is the lead author for five of the “Pad++” prior art references, which are technical publications. *See supra* note 1.

he further testifies that “conceptually, yes, it [Pad++] should run” because “[i]t’s [the PDA] a general purpose computer system.” As recognized by the Supreme Court, “[a] person of ordinary skill in the art is also a person of ordinary creativity, not an automaton.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 421 (2007). Accordingly, one of ordinary skill in demonstrating “ordinary creativity” would experiment with the technical features, such that Pad++ could run on a PDA or mobile phone.

Third, Patent Owner argues that “[t]he lack of motivation to combine Pad++ with a PDA or mobile phone is further supported by the testimony of Scott Forstall, who is the co-developer of the iPhone’s user interface and who worked closely with Steve Jobs during development of the iPhone user interface” and “Mr. Forstall testified in the litigation that there were challenges in designing a web browser that could bring the entire Internet to a mobile phone.” (Appeal Br. 27; *see also* Reply Br. 17.) The Forstall Deposition state that following:

Q. So what challenges did you face in designing a web browser that could bring the entire real Internet to a mobile phone?

A. The mobile phone is relatively small compared to a computer And so *one challenge had to do with content that was created for a window size significantly different from a mobile phone.*

A second challenge we faced was designing a touch interface. We were designing the iPhone to work based on your finger touching, and that’s not what the Internet in general had been designed around.

. . .
There were challenges related to the performance of the processor on a mobile device.

The speed of the data connection on a mobile device was slower than the general speed on a computer, especially when the computer was plugged into the Internet.

...

The amount of memory we had had in the iPhone was less than the amount of memory in a computer.

(Ex. H, 42:24 to 44:6 (emphases added).) Again, as recognized by the Supreme Court, “[a] person of ordinary skill in the art is also a person of ordinary creativity, not an automaton.” *KSR*, 550 U.S. 398 at 421. Accordingly, one of ordinary skill in demonstrating “ordinary creativity” would experiment with the technical features (e.g., window size, touch interface, performance of processor, speed to data connection, and memory), such that Pad++ could run on a mobile device (e.g., iPhone). Moreover, Pad++ Tour expressly states that “[y]our whole desktop could be zoomable” and “[t]his seems especially attractive for systems which have small screens, such as handheld computers (i.e. PDA’s).” (P. 141.) As such, even if Patent Owner is correct there is a “lack of motivation” to combine Pad++ with a mobile phone (e.g. Apple iPhone), Apple, Nokia, and Palm are cumulative references.

Fourth, Patent Owner argues that “[a] PHOSITA could not have implemented Pad++ on any PDA or mobile phone available by June, 2001, including the Nokia Communicator, Palm VII, or Apple Newton” and points to the Wolf Declaration for support. (Appeal Br. 29; *see also* Reply Br. 18.) The Wolf Declaration (Ex. B) states the following:

The zooming interface requirements of Pad++ were to support real-time animations of at least 20,000 objects at a frame rate of 10 frames per second. Pad++ also used floating point coordinates to speedy a zoom range of 14 orders of

magnitude. Bederson further notes that scalable fonts such as Microsoft Window's TrueType fonts were not suitable.

(Ex. B, ¶ 40 (citations omitted).)

Testing with the processor enabled also demonstrates how limited the HTML capabilities of the Pad++ browser were. The Pad++ browser would have been completely unusable on a desktop PC in 2000-2001. Rendering of commercial Web pages was abysmal. Without even considering that the animation frame rate would have been on the order of 100 times too slow on the fastest PDA, the browser itself would have been even more unusable on a PDA.

(Ex. B, ¶ 46.) Similarly, Patent Owner also has performed testing to determine “performance for a text-only demo Web page” and “[t]esting of window zooms with Pad++,” using the “Requirements” and “Zooming interface requirements” as disclose in a Pad++ reference,⁷ and points to paragraphs 102–107 of the Howard Declaration for such test results, which are characterized by “changes in zoom levels [being] large and unpredictable and continued after the mouse button was released” and “large lags would make the window zooming option totally unusable.” (Appeal Br. 32.) Thus, Patent Owner argues, “[a]s supported by substantial fact-based evidence, the combined teachings of references would have suggested to a PHOSITA that the resulting combination would have been completely unusable.” (*Id.* at 34.) Again, as recognized by the Supreme Court, “[a] person of ordinary skill in the art is also a person of ordinary creativity, not an automaton.” *KSR*, 550 U.S. at 421. Accordingly, one of ordinary skill in demonstrating

⁷ Ben Bederson & Jon Meyer, *Implementing a Zooming User Interface: Experience Building Pad ++*, 28 SOFTWARE–PRACTICE & EXPERIENCE 1101–35 (1998).

“ordinary creativity” would experiment with the parameters listed in the “Requirements” or “Zooming interface requirements” in order to achieved optimal results, rather than strictly adhering to such requirement to achieve an “unusable” result. Moreover, Pad++ Tour expressly states that “[y]our whole desktop could be zoomable” and “[t]his seems especially attractive for systems which have small screens, such as handheld computers (i.e. PDA’s).” (P. 141.) As such, even if Patent Owner is correct that the Nokia Communicator 9000, Apple MessagePad, and Palm VII, are “unusable,” Apple, Nokia, and Palm are cumulative references.

Fourth, Patent Owner agues the following:

The rendered text using the text rendering techniques used by Pad++ would also be unreadable for many, if not most page views. Examples of the rendering of several popular Web pages using Pad++ and simulating a screen resolution of 320x240, which corresponds to the resolution of the Apple Newton . . . The text is clearly unreadable at a zoom level of 100%, and zooming in does not improve it much, as demonstrated on the following zoom in on a portion on aol.com

. . .

(Appeal Br. 35.) Similarly, Patent Owner agues that “[e]ven if Pad++ could have been implemented on the Nokia Communicator 9000, Apple MessagePad, and Palm VII, the text rendering speed would have been orders of magnitude too slow to be useable, and most of the text would have been unreadable, even when zoomed in at 300%.” (*Id.* at 36; *see also* Reply Br. 22–26.) Again, Pad++ Tour expressly states that “[y]our whole desktop could be zoomable” and “[t]his seems especially attractive for systems which have small screens, such as handheld computers (i.e. PDA’s).” (P. 141.) As such, even if Patent Owner is correct that the Nokia

Communicator 9000, Apple MessagePad, and Palm VII, are “unusable,”
Apple, Nokia, and Palm are cumulative references.

Fifth, Patent Owner argues

In contrast to Pad++, which unequivocally was not designed to operate on a PDA, the PocketPiccolo.NET framework was designed specifically for PDAs and mobile phones running Microsoft’s Pocket PC operating system. These PDAs had processors that were approximately 27 times or more times faster than the Apple Newton processor. Even so, the performance of applications using the PocketPiccolo.NET were limited by the processor limitations of PDAs, as explicitly acknowledged by Bederson.

(Appeal Br. 39; *see also* Reply Br. 26–30.) Accordingly, Patent Owner further argues the following:

A PHOSITA with any experience whatsoever in developing software to operate on a PDA prior to June, 2001 would immediately recognize the impossibility of implementing a usable version of Pad++ on a PDA such as Nokia Communicator 9000, the Apple Newton, and the Palm VII following the principles and teachings of the Pad++ references, would have had no motivation to even attempt to implement Pad++ on any of these devices, nor have any expectation of success.

(*Id.* at 41.) However, as discussed previously, Professor Bederson testified that “[s]o my answer would be conceptually, yes, it [Pad++] should run” because “[i]t’s [the PDA] a general purpose computer system.” (Ex. G, 181: 9–16.) Moreover, Pad++ Tour expressly states that “[y]our whole desktop could be zoomable” and “[t]his seems especially attractive for systems which have small screens, such as handheld computers (i.e. PDA’s).” (P. 141.)

Therefore, we agree with the Examiner that the combination of Pad++ and SVF would have rendered obvious dependent claim 2, which includes the limitation “wherein the device comprises a mobile phone.”

Accordingly, we sustain the rejection of dependent claim 2 under 35 U.S.C. § 103(a). Independent claims 32 and 42 recite limitations similar to those discussed with respect to dependent claim 2. Claims 33–39 and 44–51 depend from claim 32, and Patent Owner had not presented any additional substantive arguments with respect to these claims. Therefore, we sustain the rejection of claims 32–39, 42, and 44–51 under 35 U.S.C. § 103(a) for the same reasons discussed with respect to dependent claim 2.

§ 103 Rejection—Pad++, SVF, and Bray

We are persuaded by Patent Owner’s arguments (Appeal Br. 48–49; *see also* Reply Br. 31) that the combination of Pad++, SVF, and Bray would not have rendered obvious dependent claim 29, which includes the limitation “wherein the HTML-based Web content includes cascading style sheet content defining layout and presentation attributes for the Web page.”

The Examiner found that that the XML (extensible markup language) delivery over the Web, using the cascading style sheet (CSS) of Bray, corresponds to the limitation “wherein the HTML-based Web content includes cascading style sheet content defining layout and presentation attributes for the Web page.” (Ans. 46–47.) The Examiner concluded that “one of ordinary skill in the art would have been motivated to modify the Pad++ zoomable Web browser to accommodate the newer extensible markup standard, XML, along with the formatting standard CSS.” (*Id.* at

47.) While the Examiner has established a prima facie case of obviousness, the Examiner has not properly considered Patent Owner's rebuttal evidence.

Bray relates to "Microsoft Internet Explorer 5's promised and actual implementation of XML." (P. 1.) Bray explains the following:

There are a bunch of different ways to deliver XML over the Web. . . . The right thing to do is to send the XML to a browser with a stylesheet, today in in CSS and before too long, in XSL.

This has a bunch of advantages

(*Id.* (emphases omitted).) Bray explains that such advantages include "cutting[] down on the amount of data you have to transmit," "offloading[] a lot of the formatting work to the browser that's coasting along on your under-worked Pentium desktop," and "allowing[] you to do cool things with the DOM [document object model]." (P. 2.)

Initially, we agree with the Examiner that a person of ordinary skill in the art would have recognized that incorporating the known XML of Bray having a cascading style sheet (CSS) into the known Pad++ zoomable web browser, would improve Pad++ by providing the advantages of cutting down on the amount of data transmitted or offloading formatting to the browser. *See KSR*, 550 U.S. at 417 ("[I]f a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way.").

However, to rebut the Examiner's prima facie case, Patent Owner submitted a Declaration under 37 C.F.R. § 1.132 of Wolf (formerly Scott

Collins), dated July 24, 2017 (Ex. D).⁸ The relevant portions of the Wolf declaration are reproduced below:

Adding support for CSS2 required a complete redesign of the rendering engine architecture. . . . This aspect alone required a tremendous amount of experimentation, with the results being anything but predictable, and well beyond what could have been implemented by a PHOSITA. Gecko had hundreds of classes, and thousands of methods. Modifying a method for one class often would “break” the code for another class or method. We had dozens of people working on the code at Mozilla, including those working including those working on Gecko, and many more working on other aspects of what would become the Netscape 6 browser. And these people were very skilled.

(Ex. D, ¶ 29.)

Neither of the HTML or CSS specs define what functions to implement. Rather, they define a complex set of interrelated rules that must be followed to comply with the standards. *It took highly skilled teams for each of Microsoft, Mozilla, and Opera several years to implement HTML 4/4.01 and CSS2. To assert a PHOSITA could do the same without undue experimentation is just plain wrong.*

(Ex. D, ¶ 30.)

Implementing HTML 4/4.01 was exceedingly difficult and frustrating, and required an extraordinary level of experimentation. It was made that much more difficult by the inclusion of CSS 2 and the CSS box model, which is inherently tied to a proper implementation of HTML 4/4.01. CSS was not

⁸ Wolf has “extensive experience working with HTML, Cascading Style Sheets (CSS), Web browser rendering engines and Web browsers gained from . . . work experience at *Netscape, Mozilla.org, and SourceForge, Inc.*” and “was a member of the core team that developed *the Mozilla “Gecko”* Rendering engine. (Ex. D, ¶ 1 (emphases in original).)

something you could merely add to an existing browser. In fact, when Netscape tried to implement CSS in Netscape 4 it did so by internally translating CSS rules into snippets of JavaScript. Needless to say, this was not a viable approach. Microsoft, even with its hordes of developers working on IE, also had difficulty with CSS, as evidenced by the Web Standards Project (WaSP), which published IE's Top 10 CSS Problems in November 1998.

(Ex. D, ¶ 66.)

In view of Wolf's statements that: (i) "[a]dding support for CSS2 required a complete redesign of the rendering engine architecture" and "[t]his aspect alone required a tremendous amount of experimentation"; (ii) "[i]t took highly skilled teams for each of Microsoft, Mozilla, and Opera several years to implement HTML 4/4.01 and CSS2"; (iii) "[i]t took highly skilled teams for each of Microsoft, Mozilla, and Opera several years to implement HTML 4/4.01 and CSS2"; and (iv) "[i]mplementing HTML 4/4.01 was exceedingly difficult and frustrating, and required an extraordinary level of experimentation" and "[i]t was made that much more difficult by the inclusion of CSS 2 and the CSS box model, which is inherently tied to a proper implementation of HTML 4/4.01," one of ordinary skill in the art could not have combined the cascading style sheet (CSS) content of Bray with the Pad++ zoomable browser, due to technological difficulties. The Examiner has not provided any commentary with respect to adequacy of Patent Owner's declaration evidence, which is legal error. *See In re Alton*, 76 F.3d 1168, 1174 (Fed. Cir. 1996) ("the summary dismissal of the declaration, without an adequate explanation" is error).

Accordingly, we are persuaded by Patent Owner's argument as follows:

It would have been well-outside the capabilities of a PHOSITA to add CSS support to Pad++, there would be no motivation to attempt to do so, and no expectation of success. Additionally, CSS could not merely be added to an existing browser.

(Appeal Br. 48–49; *see also* Reply Br. 31.)

Therefore, we do not agree with the Examiner that the combination of Pad++, SVF, and Bray would have rendered obvious dependent claim 29, which includes the limitation “wherein the HTML-based Web content includes cascading style sheet content defining layout and presentation attributes for the Web page.”

Accordingly, we do not sustain the rejection of dependent claim 29 under 35 U.S.C. § 103(a).

Other Pending Rejections

We do not reach the rejections of claims 1–28, 32–39, 42, and 44–51 under 35 U.S.C. § 103(a) as unpatentable over various combinations of Pad++, SVF, VML, Nokia, Palm, and Newton. Affirmance of the rejections discussed previously renders it unnecessary to reach the remaining rejections, as all of pending claims have been addressed and found unpatentable. *Cf. In re Gleave*, 560 F.3d 1331, 1338 (Fed. Cir. 2009) (not reaching additional obviousness rejections).

Moreover, for the rejection of claims 1–29 under 35 U.S.C. § 103(a), we do not reach the issue of whether claims 1–29 are patentably distinct over cancelled claims 30, 31, 40, 41, 43, 52, 55, 59, 72 and 75.

We do not sustain the rejections for claim 29 under 35 U.S.C. § 103(a) as being unpatentable over various combinations of Pad++, SVF, VML, Nokia, Palm, Newton, and Bray for the same reasons discussed previously with respect to the rejection of dependent claim 29 under 35 U.S.C. § 103(a) as being unpatentable over Pad++, VLM, Nokia, and Palm.

DECISION

The Examiner's decision rejecting claims 1–28, 32–39, 42, and 44–51 under 35 U.S.C. § 103(a) is affirmed.

The Examiner's decision rejecting claim 29 under 35 U.S.C. § 103(a) is reversed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED-IN-PART

dm

Appeal 2019-001127
Reexamination Control 90/009,995
Patent 7,831,926 B2

PATENT OWNER:

Irell & Manella LLP
840 Newport Center Dr., Ste. 400
Newport Beach, CA 92660

THIRD PARTY REQUESTER:

NOVAK DRUCE & QUIGG LLP
1000 Louisiana Street, 53rd Floor
Houston, TX 77002



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/009,995	05/29/2012	7,831,926	161539-0005(926PO-2)	3089
29000	7590	05/18/2018		
Irell & Manella LLP 840 Newport Center Dr., Ste. 400 Newport Beach, CA 92660			EXAMINER BASEHOAR, ADAM L	
			ART UNIT	PAPER NUMBER
			3992	
			MAIL DATE	DELIVERY MODE
			05/18/2018	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

POLSINELLI PC

(NDQ REEXAMINATION GROUP)

1000 LOUISIANA STREET

FIFTY-THIRD FLOOR

HOUSTON, TX 77002

***EX PARTE* REEXAMINATION COMMUNICATION TRANSMITTAL FORM**

REEXAMINATION CONTROL NO. 90/009,995.

PATENT NO. 7,831,926.

ART UNIT 3992.

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Art Unit: 3992



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Application Number: 90/009,995

Filing Date: May 29, 2012

Appellant(s): 7,831,926

Babak Redjaian (Reg. No. 42,096)
For Appellant

EXAMINER'S ANSWER

This is in response to Patent Owner's Revised Appeal Brief filed on 02/08/2018
appealing from the Final Office Action mailed on 06/20/2017.

(1) Grounds of Rejection to be Reviewed on Appeal

Every ground of rejection set forth in the Final Office Action dated 06/20/2017 (hereafter the “Final Action”) from which the appeal is taken is being maintained by the Examiner except for the grounds of rejection (if any) listed under the subheading “WITHDRAWN REJECTIONS.” New grounds of rejection (if any) are provided under the subheading “NEW GROUNDS OF REJECTION.”

(2) Response to Argument

Appellant’s arguments filed 02/08/2018 have been fully considered but they are not persuasive.

VI. Preliminary Matters

A. Preclusion Under 37 C.F.R. 42.73(d)(3)(i)

Appellant notes (Appeal Brief: p. 6) that each of the claims under reexamination is patentably distinct from the claims cancelled in the ‘926 IPR, thus the preclusion under 37 C.F.R. 42.73(d)(3)(i) does not apply.

The Examiner respectfully disagrees with Appellant. As noted below, not all of the claims under the instant reexamination (e.g., original independent claim 1) are patentably distinct from the claims cancelled (i.e., claims 30, 31, 40, 41, 43, 52, 55, 59, 72, and 75) in the ‘926 IPR. As stated in 37 C.F.R. 42.73(d)(3)(i), “A patent applicant or owner is precluded from taking

Art Unit: 3992

action inconsistent with the adverse judgment, including obtaining in any patent: (i) A claim that is not patentably distinct from a finally refused or canceled claim.”

B. PTAB Reasoning in IPR Not Confirmed by Rule 36 Order

Appellant argues (Appeal Brief: pp. 6-7) that a Rule 36 Order for an IPR appeal under which obviousness was found under two separate combinations of references would likewise not endorse or reject any specific part of the IPR board’s reasoning. Appellant notes that in the ‘926 Final Written Decision (“FWD”), the PTAB found independent claims 30 and 52 and dependent claims 31, 40, 41, 43, 55, 59, 72, and 75 to be obvious over two separate combinations of art: (1) Zaurus, Pad++, and SVF; and (2) Zaurus, Hara, Tsutsumitake, and SVF. Thus, there cannot be preclusion with respect to any determination concerning any individual reference or combination of references, as the Rule 36 Order did not identify which issues were ‘necessary’ or ‘essential’ to the cancellation of any claim.

The Examiner respectfully disagrees with Appellant. As noted above, in the FWD the Board found that claims 30, 31, 40, 41, 43, 52, 55, 59, 72, and 75 of the Rohrabach ‘926 patent, as demonstrated by a preponderance of the evidence, were obvious to one of ordinary skill in the art and thus said claims were held invalid. The FWD indicated that Patent Owner argued that the meaning of the limitation “preserve[s] an original page layout, functionality, and design” was “a major distinction over the art cited in Petitioner’s challenges.” Further, the Federal Circuit then affirmed the Board’s decision under a Rule 36 judgment. The case cited by Appellant (*Phil-Insul Corp. v. Airlite Plastics Co.*, (Fed. Cir. Apr. 17, 2017)) appears to indicate that “A Rule 36

Art Unit: 3992

summary affirmance is a valid and final judgement of this court” and can be used “in support of ‘claim preclusion, issue preclusion, judicial estoppel, law of the case, and the like.’” In this case, the Rule 36 judgment of the IPR appeal does not appear to specifically track the “independent” and “necessary” fact pattern related to the “TecSec” decision described in *Phil-Insul Corp. v. Airlite Plastics Co.*, (Fed. Cir. Apr. 17, 2017). Therefore, as discussed in more detail below, Appellant arguments toward limitations or determinations (i.e., the same or substantially the same issues) that were already ruled to be taught or obvious in light of the prior art in the adverse judgement will not be considered persuasive in the instant proceeding.

C. Procedural History

Appellant notes (Appeal Brief: pp. 7-8) that the PTAB introduced new claim constructions (for the first time) regarding the “preservation” limitation that were different than any claim construction asserted by either Petitioner or Patent Owner during the IPR proceeding. Appellant argues that many of the claims rejections in the Final Action are premised on errant assertions and conclusory determinations made by the PTAB for which the Patent Owner was denied an opportunity for response. Appellant argues that Patent Owner should be able to argue that said errant assertions and conclusory determinations are clearly in error, as supported by evidence that was not considered by the PTAB.

The Examiner respectfully disagrees with Appellant. It appears that during the IPR proceeding the PTAB adopted its own claim construction for the “preserving limitation” after the PTAB authorized an additional claim construction briefing. While it is true that the PTAB did

Art Unit: 3992

not adopt either Petitioner's or Patent Owner's proposed constructions for the "preserving limitation," it does not appear that the PTAB acted unreasonably or "change[d] theories in midstream" with regard to the proper construction of the "preserving limitation." The PTAB merely adopted a construction in the FWD, which as noted in *SAS Institute, Inc. v. Complementsoft, LLC.*, No. 15-1346 (Fed. Cir. 2016), the "the Board is free to do."

D. PTAB Did Not Hear Similar Arguments

Appellant notes (Appeal Brief: pp. 8-10) that substantial factual evidence that was never considered by the PTAB in the '926 FWD has been introduced that clearly refutes the Examiner's assertions in support of the claims being obvious.

The Examiner agrees with Appellant that additional factual evidence has been presented that was never considered by the PTAB in the '926 FWD. However, as discussed below, most of Appellant's additional factual evidence appears to be substantially analogous to the prior submitted evidence and related arguments considered by the PTAB. For example, the current evidence and arguments, related to the difficulty in porting Pad++ onto supposedly hardware/software deficient mobile devices (Nokia Communicator, Apple Newton, and Palm) as well as analyzing and testing the Pad++ source code in light of said devices, appears to substantially overlap Appellant's prior submitted evidence and related arguments regarding porting Pad++ onto the supposedly hardware/software deficient mobile Zaurus device (not cited herein) in view of analyzing and testing the Pad++ source code in light of said device (see: Exhibit F – Reinman Declaration, Paragraphs 25-37: "I have downloaded and operated the

Art Unit: 3992

Pad++ software as well as reviewed the source code for Pad++” and Paragraphs 54-72: “Moving Pad++ to a device with a small screen was not contemplated, tested, or examined. Nor did the Bederson References disclose or teach how to specifically implement Pad++ on a PDA, or any special considerations for doing so...Bederson himself characterizes Pad++ as being too slow and difficult to port to different systems...Zaurus device’s CPU ran at a much slower frequency...memory capacity...was significantly lower...no motivation to put their Pad++ on a weaker device...conducted an experiment...system’s performance was significantly degraded...Mr. Forstall testified that one could not just take a web browser solution for a desktop and just port it to a mobile phone”). As discussed below, the Examiner does not agree that the cumulative evidence (old + new) overcomes the current obviousness rejections.

VII. Issue 1: Claims 1-29 are Patentably Distinct from the Cancelled Claims

A. Issued Claims 1-29 of the ‘926 Patent Require Translating and Preserving the Entire HTML-Based Web Content that is Retrieved

Appellant argues (Appeal Brief: pp. 10-12) that claim 1 of the ‘926 patent (and its dependents) is patentably distinct from cancelled claims 30, 31, and 52 of the ‘926 patent because the claim requires translating and preserving the entire HTML-based Web content that is retrieved, not just a portion of the HTML-based content that is retrieved. Appellant argues that, even when broadly construed in this way the limitations of claim 1 are patentably distinct over the limitations in cancelled claims 30 and 52. Additionally, Appellant argues that the Examiner has not provided any evidence that a PHOSITA skilled in the Web browser art in 2001, having reviewed the ‘926 patent specification and drawings, would interpret claim 1 in the manner alleged by the Examiner.

The Examiner respectfully disagrees with Appellant. Previously, the Appellant had specifically argued that claim 1 of the '926 patent (and its dependents) was patentably distinct from cancelled claims 30, 31, and 52 of the '926 patent because the claim requires translating and preserving the entire HTML-based Web content for at least the HTML parent document, not just a portion (see: Response After Final Action, 08/21/2017, pp. 17-18). Notably the Appellant is now more specifically arguing that independent claim 1 is patentably distinct because it requires translating all (or whatever portion) of the HTML-based Web content that is retrieved. As noted by Appellant, the Examiner acknowledges that only the retrieved portion of "HTML-based Web content associated with the Web page" is necessarily translated to produce scalable vector-based page layout information for unamended independent claim 1. However, independent claim 1 is not patentably distinct because the retrieved portion of "HTML-based Web content associated with the Web page" that is necessarily translated is not necessarily at least a portion of "HTML-based Web content associated with the Web page" that "preserves an original page layout, functionality, and design of the at least a portion of the HTML-based content" as required by at least cancelled independent claim 30 (now represented via amended independent claims 32 and 42). As noted in the Final Written Decision, "the preservation of the original layout, function, and design turns on what elements of the HTML are translated to be interpreted by the browser" (emphasis added). Thus, despite Appellant's arguments to the contrary, the claim is not patentably distinct and does not clearly require translating and preserving "the entire" HTML-based content defining an original page layout, functionality, and design on the Web page.

It is noted that an amendment to independent claim 1 was suggested on page 14 of the Final Action to overcome the above determination. Said amendment would (1) bring independent claim 1 in line with amended independent claims 32 and 42; (2) require translating and preserving the entire HTML-based Web content defining an original page layout, functionality, and design, not just a portion; and (3) formally make said independent claim (including its dependents) patentably distinct from the finally refused or cancelled claims.

The Examiner notes that the interpretation of claim 1 is at least considered valid in view of (1) claim differentiation and/or (2) the prosecution history of the Rohrabough '926 patent.

Regarding (1) the Examiner notes that unlike amended independent claims 32 and 42, which incorporate the language of cancelled independent claim 30 and at least newly require “retrieving the HTML-based content associated with the Web page” (emphasis added to highlight the new part), unamended independent claim 1 merely requires “retrieving HTML-based content associated with the Web page.” Thus, unlike amended independent claims 32 and 42, unamended independent claim 1 does not require retrieving and thus translating the entire HTML-based Web content. Similarly, cancelled independent claim 52 of the '926 patent specifically recited “retrieving and processing the HTML-based Web content” (emphasis added) and the other independent claim 79 (not subject to this reexamination) also specifically recited “retrieving and processing the HTML-based Web content” (emphasis added). Therefore, there is clearly a claim scope distinction regarding how much of the HTML-based Web content associated with the Web page is retrieved.

Regarding (2) the Examiner notes that during the prosecution history of the Rohrabough '926 patent (see: Response filed 09/09/2010 in 11/738,486), the discussion of the meaning of the

Art Unit: 3992

terminology “preserving the original page layout, functionality, and design” stated at least in part, “It is common terminology to refer to a browser ‘retrieving’ or ‘downloading’ a Web page...**It is well understood that this doesn’t imply that all of the content associated with the Web page must be retrieved or downloaded**...content may be referenced that is not supported by the requesting browser...When a browser encounters content that is not supported ‘natively’ by the browser, the browser will typically check to see if an appropriate plug-in is available...**In other instances, the content is simply ignored...the Web page may reference content that is never retrieved when the Web page is retrieved by the browser**...when a browser rendering engine encounters content or a CSS element it doesn’t recognize, **it simply ignores is...the preservation is relative to the interpretation of the page by the browser implementation itself, as opposed to preservation...based on some rigid consideration of a ‘perfect’ interpretation of the page.**” (emphasis added). Therefore, the prosecution history of the Rohrbaugh ‘926 patent itself establishes that only a portion of an HTML document or referenced content that is part of the HTML-based Web content for a Web page might be retrieved.

VIII. Issues 2 and 6: Claims 1 and 3-28 are not Obvious over the Combination of the Pad++ References on Record, New Pad++ References, and SVF References or VML, Alone or in Combination with other References

A. Bederson did not Preserve the HTML-Based Web-Content

Appellant argues (Appeal Brief: pp. 12-16) that the Examiner has misinterpreted the ‘926 FWD and that nowhere in the FWD does the Board state or allege that the “Pad++ references [do] disclose preserving the layout, functionality, and design of the Web page.”

Additionally, Appellant argues that Pad++ was not capable of rendering the vast majority of HTML elements (e.g., tables, forms, divs, spans, etc.) or handling many fundamental HTML tags used for page formatting and layout (e.g., <col>, <frame>, <button>, etc.). Appellant notes that this evidence was not rebutted in the IPR proceeding and confirmed by Patent Owner's expert Michael L. Howard. Thus, Appellant argues that this means Pad++ could not meet the preservation limitation to render any Web page that included any of the HTML tags that were not implemented or missing and pertain to one or more of page layout, functionality, and design, as the rendered page would not be rendered as defined by the HTML-based content for the Web page. Appellant argues that all of the HTML-based content must be interpreted in accordance with the HTML standards to meet the "as defined by the HTML-based content limitation." Appellant further argues that the Pad++ prototype browser could have correctly rendered approximately .005% of the billions of Web pages in existence by June 2001 and the combination of Pad++ implemented on a mobile device (e.g., PDA, mobile phone, etc.) would've been unsatisfactory for its intended purpose 99.995% of the time.

The Examiner respectfully disagrees with Appellant. The Final Written Decision appears to explicitly state that Bederson/Pad++ teaches preserving the functionality of a Web page (see: Final Written Decision, p. 20: "Pad++ discloses reading HTML and following links across the Internet...Thus, Bederson discloses preserving HTML functionality, i.e., hyperlinking, associated with a Web page", p. 21: "Bederson's disclosure...demonstrates preserving the functionality of HTML in a zoomed view") and preserving the layout, functionality, and design of a Web page (see: Final Written Decision, p. 22: "as discussed above, both Zaurus and Bederson disclose

Art Unit: 3992

preserving the layout, functionality, and design of the part of the web page that is translated (claim 30) or interpreted by the rendering engine (claim 52)"). The Examiner notes that the Board's discussion of the Zaurus reference (not utilized herein) in combination with Bederson/Pad++ does not detract from what the Board appears to have explicitly determined was taught and/or obvious in view of Bederson/Pad++.

The Examiner agrees with the Appellant that the simple Pad++ browser does not implement each and every feature of HTML. However, as currently claimed, the HTML-based Web content of the independent claims do not require any of the fundamental HTML elements/tags that Appellant argues cannot be handled by the Pad++ browser. Thus, while the simple Pad++ browser does not implement each and every feature of HTML, the Pad++ references do disclose preserving the layout, functionality, and design of the Web page. The Examiner notes the PTAB reached a similar determination in the Final Written Decision (see: Final Written Decision, p. 22: "Although...nor Bederson implements each and every feature of HTML...Bederson disclose preserving the layout, functionality, and design of the part of the web page...underlying issue is not...whether...Bederson disclose implementing all the features of HTML...issue is whether...would have been obvious", p. 23: "The fact that Bederson did not implement all, or even many, of the known capabilities of HTML does not alter the fact that Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers", p. 26: "even if Bederson did not implement all available HTML tags, the selection of which tags to implement is a matter that would be well within the knowledge and abilities of a person of ordinary skill in the art, and would be obvious").

For example, the Pad++ Tour reference clearly shows the Pad++ HTML browser preserving the original page layout, functionality, and design of the HTML code of the portion of the Web page that has been translated and rendered (see: Pad++ Tour, p. 86: “A zooming version of Mosaic and Netscape?...is a simple Pad++ web browser....fairly primitive...imagine a much more advanced version.”; p. 87: “showing Pad++ displaying an HTML document”; p. 89: “Here is a zoomed in view of the document”; p. 91: “When you follow a link, the relevant document is loaded into Pad++...Pad++ HTML browser will lay out sub-documents”; p. 93: “Simply zoom in on the sub-document to read it”). Thus, Pad++ could preserve the entire Web page of at least any HTML document that consists only of text and images (see: Bederson-3, p. 264: “visualization and navigation of WWW pages written using standard HTML...HTML provides very few basic types...includes text, images, bullet items, and horizontal rules”; Bederson-5: p. 6: “how to provide effective access to a large structure of information on a much smaller display...standard objects that pad++ supports are colored text, graphics, images, portals and hypertext markup language (HTML)”). As noted by Appellant, the Pad++ browser could correctly render some web pages. The claims simply do not require preserving the original page layout, functionality, and design for all possible HTML elements and for all possible Web pages in existence by June 2001.

B. There is no Teaching that Pad++ could have been Combined with Netscape or Mozilla, nor how such a Combination could be Implemented

Appellant argues (Appeal Brief: pp. 16-18) that while Pad++ suggests a zooming version of Mosaic or Netscape, the statement in Pad++ is not a teaching that the Pad++ browser could be combined with an existing browser such as Netscape or Mosaic, nor provides any detail on any

Art Unit: 3992

level to how Pad++ could have been combined with Mosaic or Netscape. Appellant argues that Patent Owner has provided substantial evidence that a PHOSITA would have been unable to implement a zoomable version of Netscape at the time of the invention. Regarding Mosaic, Appellant points out that at the time of the invention it had not been updated for approximately 4 years and didn't even support HTML 3 or 3.2, much less HTML 4 or CSS. Appellant further argues that PadPrints, which piggybacks on Netscape, was not a zooming version of Netscape in any way, shape, or form, nor does such a combination come remotely close to practicing the invention of any of the claims under reexamination.

The Examiner respectfully disagrees with Appellant. First, the Examiner notes that no specific version of the HTML standard is required by the current claim language and that combining the simple zoomable Pad++ browser with Mosaic and Netscape is not necessary to meet the current claim language. In regards to extending the zoomable Pad++ browser to implement all the features of HTML, which is not explicitly required, the Board noted in the FWD (see: Final Written Decision, pp. 22-23) that Bederson "suggests that a zooming version of the Netscape and Mosaic browser could be implemented using the techniques in Pad++" and that "Pad++ demonstrates the concept, [and] suggests it could be applied to the Netscape and Mosaic browsers." The PTAB's determination that extending the zoomable Pad++ browser to implement any available HTML tag was well within the knowledge and abilities of a PHOSITA is buttressed by the disclosure of the '926 patent which teaches that it was conventional to process the well-known HTML standard via the well-known functionality of preexisting web browsers (see: Rohrabough '926 Patent, column 17, lines 31-41: "commonly performed by

Art Unit: 3992

conventional browsers...Mozilla rendering engine...open source software that is readily available for use by developers”). Similarly, Pad++ Tour teaches that instead of competing with major Web browser companies to take advantage of the new features of HTML, it would be simpler to utilize existing commercial ones (see: Pad++ Tour, pp. 139-140: “a cutting edge browser - with HTML 1.0...taking advantage of the new features of HTML...piggybacks on existing commercial ones”).

C. PTAB did not Determine or Indicate that Selecting and Implementing any Available HTML Tag or All Available HTML Tags would be well within the Knowledge and Abilities of a PHOSITA

Appellant argues (Appeal Brief: p. 19) that the FWD does not say anything about the notion that implementing any available HTML tag would be well within the knowledge and abilities of a PHOSITA. Appellant argues that the FWD merely states that the “selection of which tags to implement is a matter that would be well within the knowledge and abilities of a person of ordinary skill in the art, and would be obvious.” Appellant argues that this legal conclusion is unsupported by any factual evidence of record and notes that at the time of the invention, all available HTML tags would correspond to all available HTML tags defined by HTML 4/4.01.

The Examiner respectfully disagrees with Appellant. First, the Examiner again notes that the claims do not require implementing all available tags defined by HTML 4/4.01. Second, it appears clear that the PTAB reached a determination in the Final Written Decision that implementing any available HTML tag within the Pad++ HTML browser would have been well

Art Unit: 3992

within the knowledge and abilities of a PHOSITA (see: Final Written Decision, p. 22:

“Although...nor Bederson implements each and every feature of HTML...Bederson disclose preserving the layout, functionality, and design of the part of the web page...underlying issue is not...whether...Bederson disclose implementing all the features of HTML...issue is whether...would have been obvious”, p. 23: “The fact that Bederson did not implement all, or even many, of the known capabilities of HTML does not alter the fact that Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers”, p. 26: “even if Bederson did not implement all available HTML tags, the selection of which tags to implement is a matter that would be well within the knowledge and abilities of a person of ordinary skill in the art, and would be obvious”). The PTAB’s determination clearly indicates that selecting and implementing any available HTML tag would be well within the knowledge and abilities of a PHOSITA. Here, a PHOSITA would not be implementing any available HTML tag “from scratch” but would be relying on the well-known functionality of preexisting web browsers (e.g., Pad++, Mosaic, Mozilla, etc.). For example, as noted above, this determination is buttressed by the disclosure of the ‘926 patent which teaches that it was conventional to process the well-known HTML standard via the well-known functionality of preexisting web browsers (see: Rohrabough ‘926 Patent, column 17, lines 31-41: “commonly performed by conventional browsers...Mozilla rendering engine...open source software that is readily available for use by developers”).

D. A PHOSITA would have no Motivation to Combine the Pad++ References with the SVF References

Appellant generally argues (Appeal Brief: pp. 20-21) that the combination of the Pad++ references with the SVF references does not support a *prima facie* case of obviousness, as the combination does not teach or suggest all the claim limitations, there would be no motivation to combine, nor would a PHOSITA have a reasonable expectation of success in doing so.

Appellant argues that Dr. Grimes testified in his deposition that the teaching of using vectors in SVF is redundant to the teaching of vectors in Pad++. Appellant notes that Pad++ generates Pad++ objects and that adding SVF to Pad++ would only add another layer of complexity and provides no benefit, whatsoever. Appellant opines that the combination makes no sense, and there is nothing in any of the Pad++ references or the SVF references that suggests it. Appellant argues the SVF approach to rendering vector content is not compatible with the approach used by Pad++.

The Examiner respectfully disagrees with Appellant. Both the SVF (i.e., “Simple Vector Format”) and Pad++ (see: Final Written Decision, p. 15: “there appears to be little dispute that Bederson discloses vectorized content”) references teach utilizing vectors (a redundant concept), and the rationale for the combination of the Pad++ references and the SVF references is clear in the applied rejections. In the instant case, the SVF references are evidence that further buttress that the teachings of vectors in the Pad++ references are actually vectors and by showing how the utilization of vectors and their benefits were notoriously well-known in the art. Additionally, the rejections at least noted that vectors taught in the SVF references would have increased the efficiency of presenting, zooming and panning the information rendered by the Pad++ browser (e.g., see: Grimes-1, Paragraphs 98-109: “SVF was a publicly available and well-known way of

Art Unit: 3992

presenting images and objects, such as those found on webpages, in a vector format that simplified and increased the efficiency...further enhance, increase the efficiency of, and broaden the applicability to existing operating systems and platforms of the zoomable web browser described in the Pad++ references”; SVF, p. 1: “designed to be a simple format for describing vectors”; SVF Press 1: “SVF plug-ins feature scalable vector graphics...supports additional capabilities...work with popular Web browsers”; SVF Press 2: “World-Wide Web users can now access the full power...using the Simple Vector Format...adding a host of elegant, new options for any application or context”; ‘926 patent, column 4, line 40-column 5, line 49: “attains...efficiency by taking advantage of the power of SVF...to describe almost any current web content...for adoption as a standard for vector content on the World Wide Web...SVF is also very compressible”). Therefore, the combination of the Pad++ references and the SVF references is considered proper and the mere fact that the SVF references redundantly teach vectors does not preclude said determination.

Since the functions of the software of Pad++ and the SVF standard are well-known in the art, a PHOISTA would have been enabled to combine said references as described without undue experimentation (see: Grimes-1: Paragraphs 105-109: “SVF was a Common Technique that Would Have Been Combined with the Pad++ References...to Yield Predictable Results”). The Examiner notes the PTAB reached a similar determination in the Final Written Decision (see: Final Written Decision, p. 23: “discloses vector scaling, in a manner similar to SVF...we are persuaded that claims 30 and 52 would have been obvious over the combination of Zaurus, Bederson, and SVF”). The Examiner further notes that the cited benefit of encoding the coordinate values of the Pad++ references using the SVF references to “broaden the applicability

Art Unit: 3992

to existing operating systems and platforms of the zoomable web browser described in the Pad++ references” was explicitly a benefit (i.e., increased portability) sought by the developers of Pad++ (see: Bederson-4, p. 1128: “continuing goal has been to make Pad++ as portable as possible. Because people download, compile, and run Pad++ on all kinds of systems around the world”).

E. A PHOSITA would have no Motivation to Combine the Pad++ References with VML

Appellant generally argues (Appeal Brief: pp. 21-23) that the combination of the Pad++ references with VML does not support a *prima facie* case of obviousness, as the combination does not teach or suggest all the claim limitations, there would be no motivation to combine, nor would a PHOSITA have a reasonable expectation of success in doing so. Appellant argues there is a failure to identify how Pad++ could be combined with VML or to identify where in the Pad++ references or VML reference there is any suggestion of combining the references. Appellant notes that Pad++ uses a spatial indexing scheme that is implemented, in part based on Pad++ objects, which use Pad++ coordinates but VML does not use a spatial indexing scheme. Thus, Appellant argues that Pad++ and VML are not compatible and a PHOSITA would not be able to figure out a way to combine them.

The Examiner respectfully disagrees with Appellant. Both the VML (i.e., “Vector Markup Language”) and Pad++ references (see: Final Written Decision, p. 15: “there appears to be little dispute that Bederson discloses vectorized content”) teach utilizing vectors, and the

Art Unit: 3992

rationale for the combination of the Pad++ references and the VML reference is clear in the applied rejections. In the instant case, the VML reference is evidence that further buttresses that the teachings of vectors in the Pad++ references are actually vectors and by showing how the utilization of vectors and their benefits were notoriously well-known in the art. Additionally, the rejections at least noted that vectors taught in the VML reference would have increased the efficiency of presenting, zooming and panning the information rendered by the Pad++ browser (e.g., see: Grimes-2, Paragraphs 18-22: “VML was a publicly available and well-known way of presenting images and objects, such as those found on webpages...format that simplified and increased the efficiency of presenting, zooming, and panning such information...broaden the applicability to existing operating systems and platforms of the zoomable web browser described in the Pad++ references...would have yielded the predictable results”; VML, p. 1: “an overview of the way VML is organized and how it interacts with...HTML”, p. 7: “Use the existing mechanisms of HTML and CSS...Provide efficient representations of vector graphics”).

Therefore, the combination of the Pad++ references and the VML reference is considered proper.

Since the functions of the software of Pad++ and the VML standard are well-known in the art, a PHOISTA would have been enabled to combine said references as described without undue experimentation (see: Grimes-2, Paragraphs 19-22: “Combining VML with Either of the Pad++ References...Yields Predictable Results”). The Examiner notes that the cited benefit of encoding the coordinate values of the Pad++ references using the VML standard to “broaden the applicability to existing operating systems and platforms of the zoomable web browser described in the Pad++ references” was explicitly a benefit (i.e., increased portability) sought by the developers of Pad++ (see: Bederson-4, p. 1128: “continuing goal has been to make Pad++ as

Art Unit: 3992

portable as possible. Because people download, compile, and run Pad++ on all kinds of systems around the world”).

IX. Issues 3-4 and Issues 7-8: Claims 2, 32-39, 42, and 44-51 are not Obvious over the Combination of the Pad++ References on Record, New Pad++ References, SVF, and any of Nokia, Palm, or Newton or over the Combination of the Pad++ References on Record, New Pad++ References, VML, and any of Nokia, Palm, or Newton

A. Pad++ Browser was not Designed to Operate on a PDA

Appellant argues (Appeal Brief: pp. 23-25) that it is a demonstrably false statement that the Pad++ browser was designed to operate on PDAs. Appellant argues that all of the Pad++ references relating to the Pad++ browser illustrate the browser being implemented on a desktop computer or workstation. Appellant further argues that the usefulness of the Pad++ “tree” browser requires the use of a large high-resolution display, which is in stark contrast to PDAs at the time.

The Examiner respectfully disagrees with Appellant. The Pad++ references specifically teach wherein the Pad++ browser was designed to operate on PDAs (see: Bederson-4, p. 1128: “continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems around the world”; Bederson-5, p. 5: “Pad++...designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)”) and was noted to be especially attractive for systems which have small screens, such as handheld computers (i.e., PDAs)(see: Benderson-5, p. 6: “provide effective access to a large structure of information on a much smaller display”; Pad++ Tour, pp. 140-141: “Pad++...windowing system based on

Art Unit: 3992

zooming. Your whole desktop could be zoomable. This seems especially attractive for systems which have small screens, such as handheld computers (i.e., PDA's)"). Therefore, the Pad++ references indisputably suggest that the Pad++ browser can and should be applied to PDAs. The motivation is clear. The Examiner notes that the "tree" browser is merely one feature of the Pad++ browser, and regardless, the Pad++ references specifically encourage the use of said Pad++ browser on PDAs with small screens so that a user can have "effective access to a large structure of information" because "Your whole desktop could be zoomable."

The Examiner notes the PTAB heard similar arguments (see: Exhibit F - Reinman Declaration, Paragraphs 54-71: "One of ordinary skill in the art would not think to display...Pad++, on a PDA...would not believe that there would be reasonable expectation of success to combine the Bederson References with the Zaurus References") and reached a similar determination in the Final Written Decision (see: Final Written Decision, pp. 18-19: "We are not persuaded...that one would not be motivated to port Bederson's Pad++ browser...because of technical difficulties resulting from limited computing capacity and system incompatibilities", p. 23: "Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers, and states that it was being designed for use on devices with small screens, such as PDAs...would be motivated to combine"). Therefore, the PTAB clearly determined that the Pad++ browser was motivated to be and could be ported to PDAs at the time of the invention.

B. Pad++ Was Never Implemented on a Mobile Device, PDA, or Mobile Phone

Appellant argues (Appeal Brief: p. 25), citing testimony of Ben Bederson, that it is undisputed that Pad++ was never implemented on a mobile device, PDA, or mobile phone.

The Examiner generally agrees with Appellant. As noted above, while the Pad++ references specifically teach wherein the Pad++ browser was designed to operate on PDAs (see: Bederson-4, p. 1128: “Pad++ is implemented in C++...continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems all around the world”; Bederson-5, p. 5: “Pad++...designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)”) and was noted to be especially attractive for systems which have small screens, such as handheld computers (i.e., PDAs) (see: Benderson-5, p. 6: “provide effective access to a large structure of information on a much smaller display”; Pad++ Tour, pp. 140-141: “Pad++...windowing system based on zooming. Your whole desktop could be zoomable. This seems especially attractive for systems which have small screens, such as handheld computers (i.e., PDA’s)”), the evidence of record does not actively show Pad++ being implemented on a mobile device, PDA, or mobile phone. The Examiner agrees Ben Bederson admitted that the Pad++ “code itself did not run on PDAs” (see: Exhibit G - Bederson Testimony, p. 290) and that he “did not” implement Pad++ on a PDA device (see: Exhibit G - Bederson Testimony, p. 291).

C. Pad++ was never Designed to Operate on a PDA

Appellant argues (Appeal Brief: pp. 25-28) that the final design of Pad++ clearly discloses a system that was specifically designed for a desktop computer or workstation running UNIX or Linux and X Windows, and not designed to be operated on a PDA. Appellant notes that Ben Bederson testified that (1) he never implemented Pad++ on a PDA; (2) the Pad++ code

Art Unit: 3992

did not run on a PDA; (3) he was not familiar with the technical capabilities and amount of memory available by PDAs at the time; and (4) the target platform was any computer that could run Linux and X graphics systems. Therefore, Appellant argues that Ben Bederson's statement on being designed to operate on a PDA cannot be true, since there were not considerations of the hardware capabilities of PDAs, the target platform, and the like. Additionally, Appellant argues that subsequent development of the ZUI "Piccolo" in 2002 by Ben Bederson provides further evidence that Pad++ was not designed to operate on a PDA. Appellant notes that at the time of the invention, PDAs were not considered general purpose computing systems but rather had very limited processor and memory resources as well as operating systems entirely different than computers running Linux and X graphics. Appellant also argues that the lack of motivation to combine Pad++ with a PDA or mobile phone is further supported by the testimony of Scott Forstall who testified that there were challenges in designing a web browser that could bring the entire Internet to a mobile phone.

The Examiner respectfully disagrees with Appellant. Again, as noted above, the Pad++ references specifically teach wherein the Pad++ browser was designed to operate on PDAs (see: Bederson-4, p. 1128: "Pad++ is implemented in C++...continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems all around the world"; Bederson-5, p. 5: "Pad++...designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)") and was noted to be especially attractive for systems which have small screens, such as handheld computers (i.e., PDAs) (see: Benderson-5, p. 6: "provide effective access to a large structure of information on a much smaller display"; Pad++

Art Unit: 3992

Tour, pp. 140-141: “Pad++...windowing system based on zooming. Your whole desktop could be zoomable. This seems especially attractive for systems which have small screens, such as handheld computers (i.e., PDA’s)”). While Ben Bederson did admit that the Pad++ “code itself did not run on PDAs” (see: Exhibit G - Bederson Testimony, p. 290) and that he “did not” implement Pad++ on a PDA device (see: Exhibit G - Bederson Testimony, p. 291), Mr. Bederson also admitted that even though it could well be difficult, Pad++ should be able to run on PDAs (see: Exhibit G - Bederson Testimony, pp. 180-181: “my answer would be conceptually, yes, it should run”). The fact that Ben Bederson himself did not choose to implement and run Pad++ on a PDA is not relevant, especially in light of the fact that he was not familiar with the technical capabilities of PDAs at the time (see: Exhibit G - Bederson Testimony, pp. 181 and 293). What is important is what the combined teachings of the references would have suggested to those of ordinary skill in the art. In the instant case, as noted above, the rationale for combining the references is clear. Additionally, as noted above, the PTAB has determined that the Pad++ browser was motivated to be and could be ported to PDAs at the time of the invention. The Examiner notes that the Pad++ developers “continuing goal...to make Pad++ as portable as possible...and run Pad++ on all kinds of systems all around the world” provides additional motivation (i.e., increased flexibility) for developing Pad++ so it could be run on as many disparate systems/platforms as possible.

The Examiner further notes that the testimony of Scott Forstall (see: Exhibit H – Forstall Testimony & Exhibit F – Reinman Declaration, Paragraph 70: “Mr. Forstall testified that one could not just take a web browser solution for a desktop and just port it to a mobile phone”) was previously before the PTAB, and as noted above, the PTAB determined that the Pad++ browser

Art Unit: 3992

was motivated to be and could be ported to PDAs at the time of the invention (see: Final Written Decision, pp. 18-19: “We are not persuaded...that one would not be motivated to port Bederson’s Pad++ browser...because of technical difficulties resulting from limited computing capacity and system incompatibilities”, p. 23: “Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers, and states that it was being designed for use on devices with small screens, such as PDAs...would be motivated to combine”). Additionally, the Examiner notes that a web browser that could bring the entire Internet to a mobile phone is not a limitation of the claimed invention. The scope of the claim functionality of the Rohrabough ‘926 patent is not nearly that vast. The instant claims only require a user requesting access to a Web page comprising HTML-based Web content.

D. A PHOSITA could not have Implemented Pad++ in a Usable Manner on any PDA or Mobile Phone Available by June, 2001

Appellant argues (Appeal Brief: pp. 29-31) that a PHOSITA could not have implemented Pad++ on any PDA or mobile phone available by June, 2001 including the Nokia Communicator, Palm VII, or Apple Newton. Appellant notes that the system of Bederson required floating point coordinates rather than integer coordinates and there were no PDAs with numeric processors or FPUs available by June, 2001. Appellant further argues, in light of demonstrated testing, that the extremely slow frame rate, unpredictable step changes, and lack of real-time visual feedback would render Pad++ completely unusable on the fastest processor available for a PDA or mobile phone. Additionally, Appellant argues that processors of the

Art Unit: 3992

Nokia Communicator, Apple Newton (MessagePad 120), and Palm VII are extremely slower than the 200 MHz Intel Pentium Pro.

The Examiner respectfully disagrees with Appellant. First, as noted above, the PTAB clearly determined that the Pad++ browser was motivated to be and could be ported to PDAs at the time of the invention (see: Final Written Decision, pp. 18-19: “We are not persuaded...that one would not be motivated to port Bederson’s Pad++ browser...because of technical difficulties resulting from limited computing capacity and system incompatibilities”, p. 23: “Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers, and states that it was being designed for use on devices with small screens, such as PDAs...would be motivated to combine”). The PTAB heard similar arguments regarding a supposedly hardware/software deficient mobile device as well as analyzing and testing the Pad++ source code in light of said device (see: Exhibit F - Reinman Declaration, Paragraphs 25-37: “I have downloaded and operated the Pad++ software as well as reviewed the source code for Pad++” and Paragraphs 54-72: “Moving Pad++ to a device with a small screen was not contemplated, tested, or examined. Nor did the Bederson References disclose or teach how to specifically implement Pad++ on a PDA, or any special considerations for doing so...Bederson himself characterizes Pad++ as being too slow and difficult to port to different systems...Zaurus device’s CPU ran at a much slower frequency...memory capacity...was significantly lower...no motivation to put their Pad++ on a weaker device which would not have the processing capability or memory capacity to handle Pad++...conducted an experiment...system’s performance was significantly degraded”) and explicitly determined that the Pad++ browser

Art Unit: 3992

could be ported to the mobile Zaurus device. While the mobile Zaurus device is not utilized herein, the Examiner notes that said device is not significantly more capable/powerful than the Nokia Communicator, Apple Newton (MessagePad 120), and Palm VII devices argued above.

Second, the specific hardware/software capabilities of the portable devices taught in Nokia, Palm, and/or Newton do not preclude the appropriate combination of said references with the Pad++ references. The test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference (or vice versa); nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981). Again, as noted above, the Pad++ references specifically teach wherein the Pad++ browser was designed to operate on PDAs (see: Bederson-4, p. 1128: “Pad++ is implemented in C++...continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems all around the world”; Bederson-5, p. 5: “Pad++...designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)”) and was noted to be especially attractive for systems which have small screens, such as handheld computers (i.e., PDAs) (see: Benderson-5, p. 6: “provide effective access to a large structure of information on a much smaller display”; Pad++ Tour, pp. 140-141: “Pad++...windowing system based on zooming. Your whole desktop could be zoomable. This seems especially attractive for systems which have small screens, such as handheld computers (i.e., PDA’s”).

Further, it is noted that certain portions of the specific hardware/software capabilities of the Pad++ references (e.g., “a 200 MHz Pentium Pro running Linux” and “Maintaining smooth real-time interaction is crucial”) are not explicit formal requirements of the Pad++ browser (see: Bederson-4, pp. 1103-1104: “started constructing a list of basic technical requirements... **It is not meant as a formal definition of the requirements of a ZUI**...smooth real-time interaction is crucial...reference platform is a 200 MHz Pentium Pro running Linux”, p. 1128: “Pad++ is implemented in C++...continuing goal has been to make Pad++ as portable as possible...compile, and run Pad++ on all kinds of systems all around the world”; Exhibit G - Bederson Testimony, p. 293: “Do you agree that a fundamental requirement of Pad++ was the ability to perform zooming with smooth real time animation?...certainly a goal... **But I wouldn’t say that was a requirement for Pad++ to run**...operate without errors and do all the things that it was supposed to do except...would be slow”)(emphasis added). Appellant’s additional testing results are also not considered persuasive at least because they appear to substantially overlap the testing evidence previously conducted and argued before the PTAB as well as because they don’t appear to take into account that the Pad++ references also taught wherein the Pad++ browser could be applied flexibly when attempting to support any specific animation-frame rate requirements (see: Bederson-4, pp. 1117-1118: “better approach is to test each object for visibility...can significantly improve frame rates”; Bederson-5, p. 23: “Animations and zooming... **independent of processor speed, scene complexity, and window size**...rendering more or fewer frames, as time allows”; Programmer’s Guide, pp. 14-15: “Pad++ uses several techniques to maintain efficiency...Pad++ adjusts the frame rate during animations and zooming to maintain constant perceptual flow...takes 750 milliseconds”)(emphasis added).

E. Interactive Rendering of Text not Possible on any of the Nokia Communicator, Palm VII, or Apple Newton

Appellant argues (Appeal Brief: pp. 31-35) that high frame rates were critical for Pad++'s navigation scheme and one of the zooming interface requirements was maintaining smooth real-time interaction. Appellant notes that testing a demo page on a 200 MHz Pentium Pro PC with the numeric processor disabled resulted in the best frame rate of one frame in 11 seconds. Appellant further noted, via testing, that text rendering on the Nokia 9000 Communicator, Apple Newton, and Palm VII would be orders of magnitude too slow to be usable. Additionally Appellant argues, in light of testing shown in the Howard Declaration, that the Adjustable Frame Rate described in Bederson-5 would have negligible effect on the frame rate of a PDA or mobile phone, and certainly couldn't be used to speed up the frame rate by a factor of 10 or more, much less speed it up by a factor of 100 or more. Appellant notes a PHOSITA has to be enabled to practice the invention to support a *prima facie* case of obviousness and opines that the Examiner is asserting the hardware/software capabilities of these devices are irrelevant, which is refuted by each of Wolf, Howard, and Frostall, as well as Ben Bederson himself.

The Examiner respectfully disagrees with Appellant. First, as noted above, the PTAB clearly determined that the Pad++ browser was motivated to be and could be ported to PDAs at the time of the invention (see: Final Written Decision, pp. 18-19: "We are not persuaded...that one would not be motivated to port Bederson's Pad++ browser...because of technical difficulties

Art Unit: 3992

resulting from limited computing capacity and system incompatibilities”, p. 23: “Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers, and states that it was being designed for use on devices with small screens, such as PDAs...would be motivated to combine”). Additionally, the PTAB heard similar arguments regarding the rendering of text on a supposedly hardware/software deficient mobile device (see: Exhibit F - Reinman Declaration, Paragraph 35: “fundamental requirement of Pad++...to support real-time animations of at least 20,000 objects at a frame rate of 10 frames per second...Pad++ implemented its own fonts”, Paragraph 55: “The engine must support good fonts...At the time Bederson References were published, graphic capabilities on a PDA, such as the Zaurus device ...were limited...teaches away”, Paragraph 63: “Pad++ implemented its own fonts and associated rendering technique using floating point coordinates...a single page of text represents over 100,000 polygons which was near the limit of what a typical desktop computer could render at interactive rates...would be very slow”, and Paragraph 66: “Running pad++ on Zaurus would only serve to exacerbate these existing problems in meeting the real-time constraint”) and explicitly determined that the Pad++ browser could be ported to the mobile Zaurus device. While the mobile Zaurus device is not utilized herein, the Examiner notes that said device is not significantly more capable/powerful than the Nokia Communicator, Apple Newton (MessagePad 120), and Palm VII devices argued above.

Further, it is noted that certain portions of the specific hardware/software capabilities of the Pad++ references (e.g., “Maintain and render at least 20,000 objects with smooth interaction...Maintaining smooth real-time interaction is crucial...reference platform is a 200 MHz Pentium Pro”) are not explicit formal requirements of the Pad++ browser (see: Bederson-4,

Art Unit: 3992

pp. 1103-1104: “started constructing a list of basic technical requirements...**It is not meant as a formal definition of the requirements of a ZUI**...Maintain and render 20,000 objects with smooth interaction...Maintaining real-time interaction is crucial...reference platform is a 200 MHz Pentium Pro running Linux”, p. 1128: “Pad++ is implemented in C++...continuing goal has been to make Pad++ as portable as possible...compile, and run Pad++ on all kinds of systems all around the world”; Exhibit G - Bederson Testimony, p. 293: “Do you agree that a fundamental requirement of Pad++ was the ability to perform zooming with smooth real time animation?...certainly a goal...**But I wouldn’t say that was a requirement for Pad++ to run**...operate without errors and do all the things that it was supposed to do except...would be slow”)(emphasis added). Thus, the Examiner is not asserting that the hardware/software capabilities of the Nokia 9000 Communicator, Apple Newton, and Palm VII devices are irrelevant, but rather, similar to the PTAB’s determination with regard to the Zaurus mobile device, their specific hardware/software capabilities do necessarily preclude the appropriate combination of the references with the Pad++ references. Appellant’s additional testing results are also not considered persuasive at least because they don’t appear to take into account that the Pad++ references also taught wherein the Pad++ browser could be applied flexibly when attempting to support any specific animation-frame rate requirements (see: Bederson-4, pp. 1117-1118: “better approach is to test each object for visibility...can significantly improve frame rates”; Bederson-5, p. 23: “successfully loaded over 600 000 objects...maintained interactive rates of about 10 frames per second...Animations and zooming...**independent of processor speed, scene complexity, and window size**...rendering more or fewer frames, as time allows”; Programmer’s Guide, pp. 14-15: “Pad++ uses several techniques to maintain efficiency...Pad++

Art Unit: 3992

adjusts the frame rate during animations and zooming to maintain constant perceptual flow...takes 750 milliseconds”)(emphasis added). This functionality described in the Pad++ references would help alleviate problems related to processor speed, Web page complexity, and interactive text rendering generally argued by Appellant. Appellant’s testing (e.g., Exhibit A – Howard Declaration, Paragraphs 102-108) appears to unduly constrain all of the potential functionality taught in Pad++ that would be available to a PHOSTIA, and thus the new evidence does not overcome the same basic argument previously presented to the PTAB.

F. Pad++ Rendering of Text on Small, Low-Resolution Displays would be Unreadable and Unusable

Appellant argues (Appeal Brief: pp. 35-37) that rendered text using the text rendering techniques used by Pad++ would be unreadable for many, if not most Web page views. Via renderings of example Web pages, Appellant notes that the displayed text is clearly unreadable at a zoom level of 100%, and even when zoomed at 300%, most of the text is unreadable. Additionally, the Appellant notes that none of the claims in this reexamination are limited to use of a Palm IIIc shown in Figures 7A-B, 8A-B and 9A-B. Appellant further acknowledges that the Palm IIIc had insufficient memory and processing capabilities to implement any of the inventions in the reexamination claims, and has never asserted these claimed inventions could have been implemented on a Palm IIIc. Appellant states that the claimed invention could be implemented on much more powerful devices such as PDAs and Pocket PCs. Finally, Appellant argues that the proposed modification of any of the Nokia Communicator 9000, Apple Newton, and Palm VII device would result in device that was unusable on multiple levels (i.e., quick zooming functionality, readable font-rendering, and correct Web page rendering).

The Examiner respectfully disagrees with Appellant. Again, as noted above, the Pad++ references specifically teach wherein the Pad++ browser was designed to operate on PDAs (see: Bederson-4, p. 1128: “Pad++ is implemented in C++...continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems all around the world”; Bederson-5, p. 5: “Pad++...designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)”) and was noted to be especially attractive for systems which have small screens, such as handheld computers (i.e., PDAs) (see: Benderson-5, p. 6: “provide effective access to a large structure of information on a much smaller display”; Pad++ Tour, pp. 140-141: “Pad++...windowing system based on zooming. **Your whole desktop could be zoomable. This seems especially attractive for systems which have small screens, such as handheld computers (i.e., PDA’s)**”)(emphasis added). Thus, Pad++ teaches wherein a user could continually zoom in on the entire desktop of their handheld PDA such that they could access (read) a large structure of information (see: e.g., Bederson-3, p. 260: “Pages are scaled so that the page in focus is clearly readable”, p. 262: “a zoomable web browser using Pad++...objects can be placed at any position and at any scale”; Bederson-4: pp. 1101-1102: “Documents can be placed on the surface at any position, and also scaled to any size...great potential even on small screens”; Pad++ Tour, p. 93: “Simply zoom in on a sub-document to read it”). As noted above, even Appellant appears to agree that text rendered by Pad++ would be readable for some Web page views.

The Examiner agrees that none of the claim limitations in this reexamination are specifically limited to the Palm IIIc device shown in Figures 7A-B, 8A-B, and 9A-B. Dependent

Art Unit: 3992

claim 3 requires a “palm-held device” but this is not equivalent to the more specific Palm IIIc device. The Examiner only pointed out said implementation in response to Appellant’s previous arguments against the supposed text rendering deficiencies of the Nokia Communicator 9000, Apple Newton, and Palm VII devices. The point was that the disclosure of the ‘926 patent does not explicitly define any specific level of display resolution or processing power that would be necessary for a PDA or Pocket PC to possess in order to implement the functionality of ‘926 patent claims at the time of the invention. The representative embodiments in Figures 7A-B, 8A-B, and 9A-B appear to show the Palm IIIc with a display of only 160x160 pixels practicing limitations of the claimed invention with some text readability issues, but certainly, as argued by Appellant, the claims are not limited to such an embodiment.

G. Jazz and Piccolo

Appellant again argues (Appeal Brief: pp. 37-42) that Pad++ was never implemented on a PDA, mobile phone, or similar device and further notes that Ben Bederson abandoned Pad++ in 1998 and began working on his fourth generation ZUI called “Jazz.” Appellant notes that Bederson stated that originally it was very difficult to add new objects and interactions to Pad++. Subsequent to developing Jazz, Bederson began working on his fifth-generation ZUI called “Piccolo” that Bederson himself deemed “too slow” for implementing on a handheld device (2002 Pocket PC). Appellant further details additional work done by Ben Bederson and notes that when Bederson actually designed a ZUI for a PDA he abandoned the font-rendering and ZUI navigation approach used in Pad++ as well as had to limit the animation on the PDA due to limited processing power. Therefore, Appellant states that a PHOSITA with any experience

Art Unit: 3992

whatsoever in developing software on a PDA prior to June, 2001 would have immediately recognized the impossibility of implementing a usable version of Pad++ on a PDA such as the Nokia Communicator 9000, the Apple Newton, and the Palm VII. Appellant concludes that the Examiner's assertions are not based on factual evidence, but rather the irrefutable factual evidence is: (1) Ben Bederson had difficulty implementing a usable ZUI for a Pocket PC in 2002; and (2) Each of the Nokia Communicator 9000, the Apple Newton, and the Palm VII was at least 27 times slower than the Pocket PC used in Piccolo. Finally, Appellant argues that the PTAB did not consider any of the further noted evidence.

The Examiner respectfully disagrees with Appellant. The Pad++ references specifically teach wherein the Pad++ browser was designed to operate on PDAs (see: Bederson- 4, p. 1128: "continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems around the world"; Bederson-5, p. 5: "Pad++...operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)") and was noted to be especially attractive for systems which have small screens, such as handheld computers (i.e., PDAs) (see: Benderson-5, p. 6: "provide effective access to a large structure of information on a much smaller display"; Pad++ Tour, pp. 140-141: "Pad++...windowing system based on zooming. Your whole desktop could be zoomable. This seems especially attractive for systems which have small screens, such as handheld computers (i.e., PDA's)"). The Examiner notes that evidence of Ben Bederson's development of Jazz and the noted main limitations of Pad++ (see: Exhibit F – Reinman Declaration, Paragraphs 64-65: "Bederson...moved on from Pad++...started developing a new ZUI...called 'Jazz'...Bederson expressly noted the main

Art Unit: 3992

limitations of Pad++...was too slow...several significant design problems, and is difficult to extend...Pad++ difficult to port”) were previously before the PTAB, and as noted above, the PTAB determined that the Pad++ browser was motivated to be and could be ported to PDAs at the time of the invention (see: Final Written Decision, pp. 18-19: “We are not persuaded...that one would not be motivated to port Bederson’s Pad++ browser...because of technical difficulties resulting from limited computing capacity and system incompatibilities”, p. 23: “Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers, and states that it was being designed for use on devices with small screens, such as PDAs...would be motivated to combine”). Thus, regardless of Ben Bederson’s subsequent work developing two different ZUIs (Jazz and Piccolo), the Pad++ references clearly provide a motivation for implementing the Pad++ browser on a mobile device having a smaller display such as a Nokia Communicator device, a Palm device, and/or an Apple Newton device.

As noted above, the PTAB heard similar arguments regarding a supposedly hardware/software deficient mobile device (see: Exhibit F - Reinman Declaration, Paragraphs 54-72: “Moving Pad++ to a device with a small screen was not contemplated, tested, or examined. Nor did the Bederson References disclose or teach how to specifically implement Pad++ on a PDA, or any special considerations for doing so...Bederson himself characterizes Pad++ as being too slow and difficult to port to different systems...Zaurus device’s CPU ran at a much slower frequency...memory capacity...was significantly lower...no motivation to put their Pad++ on a weaker device which would not have the processing capability or memory capacity to handle Pad++...conducted an experiment...system’s performance was significantly degraded”) and explicitly determined that the Pad++ browser could be ported to the mobile Zaurus device.

Art Unit: 3992

While the mobile Zaurus device is not utilized herein, the Examiner notes that said device is not significantly more capable/powerful than the Nokia Communicator, Apple Newton (MessagePad 120), and Palm VII devices argued above.

Additionally, Appellant's new testing evidence is not considered persuasive because it appears premised on limitations that are not explicit formal requirements of the Pad++ browser (see: Bederson-4, pp. 1103-1104: "started constructing a list of basic technical requirements...**It is not meant as a formal definition of the requirements of a ZUI**...reference platform is a 200 MHz Pentium Pro running Linux", p. 1128: "Pad++ is implemented in C++...continuing goal has been to make Pad++ as portable as possible...compile, and run Pad++ on all kinds of systems all around the world"; Exhibit G - Bederson Testimony, p. 293: "Do you agree that a fundamental requirement of Pad++ was the ability to perform zooming with smooth real time animation?...certainly a goal...**But I wouldn't say that was a requirement for Pad++ to run**...operate without errors and do all the things that it was supposed to do except...would be slow")(emphasis added) and because it does not appear to take into account that the Pad++ references also taught wherein the Pad++ browser could be flexible when attempting to support any specific animation-frame rate requirements (see: Bederson-4, pp. 1117-1118: "better approach is to test each object for visibility...can significantly improve frame rates"; Bederson-5, p. 23: "Animations and zooming...**independent of processor speed, scene complexity, and window size**...rendering more or fewer frames, as time allows"; Programmer's Guide, pp. 14-15: "Pad++ uses several techniques to maintain efficiency...Pad++ adjusts the frame rate during animations and zooming to maintain constant perceptual flow...takes 750 milliseconds")(emphasis added).

H. An Obvious Rejection must Explain how the Teachings of the References would be Combined

Appellant argues (Appeal Brief: pp. 42-45) the Examiner makes conclusory statements based on inadequate evidence, and then states the IPR Board came to similar results. Appellant further argues that no evidence has been provided to how Netscape or Mosiac could have been combined with Pad++ or why a PHOSITA would have been motivated to do so or even have any expectation of success. Appellant additionally argues the Examiner has further failed to identify why a PHOSITA, at the time of the invention, would want to implement an HTML 1.0 Web browser on a PDA or mobile phone. When considering adding all the missing HTML tags to Pad++, Appellant argues that it is undisputed that it took a team of highly-skilled engineers at Mozilla at least 50 man-years to develop the Gecko rendering engine and opines that it is the Examiner's position that 50 man-years of development would not constitute undue experimentation. Appellant further notes that selecting which tags to implement could have been easily accomplished but actually implementing the tags in a browser rendering engine (supporting HTML 4/4.01) is entirely different. In summary, Appellant argues that a PHOSITA could not have modified Pad++ to implement all available HTML tags (a browser supporting HTML 4/4.01) at the time of the invention in view of the fact that they would need to compete against Microsoft and Netscape.

The Examiner respectfully disagrees with Appellant. First, as discussed above, the Examiner notes that the independent claims do not require processing all of the HTML tags

Art Unit: 3992

defined by HTML 2, 3.2, and/or 4/4.01. For example, independent claim 1 only requires “access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page.” As noted by Appellant, Pad++ combined with any of the mobile devices discussed above “would only be able to render an estimated 0.005% of Web pages in accordance with HTML standards.” Thus, by Appellant’s own admission Pad++ could properly render some Web pages. Therefore, while the simple Pad++ browser does not implement each and every feature of HTML, the Pad++ references do disclose preserving the layout, functionality, and design of the Web page (e.g., see: Pad++ Tour, p. 87: “Here is a screen snapshot showing Pad++ displaying an HTML document”, p. 89: “Here is a zoomed in view of the document”).

Second, the Pad++ references clearly teach the motivation (i.e., to “provide effective access to a large structure of information” and to provide a completely zoomable desktop) to implement an HTML 1.0 Web browser on a PDA or mobile phone. The Pad++ references specifically teach that the Pad++ browser was designed to operate on PDAs (see: Bederson-4, p. 1128: “continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems around the world”; Bederson-5, p. 5: “Pad++...designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)”) and was noted to be especially attractive for systems which have small screens, such as handheld computers (i.e., PDAs) (see: Benderson-5, p. 6: “provide effective access to a large structure of information on a much smaller display”; Pad++ Tour, pp. 140-141: “Pad++...windowing system based on zooming. Your whole desktop could be zoomable. This seems especially attractive for

Art Unit: 3992

systems which have small screens, such as handheld computers (i.e., PDA's"). Thus the motivation to combine the Pad++ browser with a PDA is clear.

While not a limitation of the independent claims, the Examiner notes that the PTAB heard similar arguments (see: Exhibit F – Reinman Declaration: Paragraphs 25-34: "I have downloaded and operated the Pad++ software as well as reviewed the source code for Pad++...many HTML tags that are not implemented by Pad++...In addition, there are many other HTML tags that are fundamental to content layout which are not even defined...At the time, HTML web browser rendering engines...employed hundreds of modules, and have been implemented by large teams of developers") regarding the Pad++ browser's inability to implement "many of the common HTML tags used for web pages" and the amount of developers needed to implement Netscape/Internet Explorer and reached a similar determination in the Final Written Decision (see: Final Written Decision, p. 22: "Although...nor Bederson implements each and every feature of HTML...Bederson disclose preserving the layout, functionality, and design of the part of the web page...underlying issue is not...whether...Bederson disclose implementing all the features of HTML...issue is whether...would have been obvious", p. 23: "The fact that Bederson did not implement all, or even many, of the known capabilities of HTML does not alter the fact that Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers", p. 26: "even if Bederson did not implement all available HTML tags, the selection of which tags to implement is a matter that would be well within the knowledge and abilities of a person of ordinary skill in the art, and would be obvious"). The PTAB's determination clearly indicates that selecting and implementing any available HTML tag would be well within the knowledge and abilities of a PHOSITA. Here, a PHOSITA would not be

Art Unit: 3992

implementing a Web browser from scratch or by needing “at least 50 man-years of development” as argued by Appellant, but would be relying on the well-known functionality of preexisting web browsers (e.g., Pad++, Mosaic, etc.) and the notoriously well-known HTML/CSS specifications. Even the disclosure of the ‘926 patent explicitly states that certain browsers provided open source code that is readily available for use by developers for conventional HTML-based Web content processing (see: ‘926 patent, column 17, lines 31-41: “recognized by those skilled in the art, the functionality...commonly performed by conventional browsers...performed by the Mozilla rendering engine, which comprises open source code....for use by developers”). The Examiner agrees that Pad++ didn’t want to get into the business of competing with the major Web browser companies, but as discussed above, instead it wanted to take advantage of the new features of HTML by utilizing (piggybacking on) the functionality of existing commercial ones (see: Pad++ Tour, pp. 139-140: “We build what was at the time a cutting edge browser - with HTML 1.0...taking advantage of the new features of HTML by building a new generation Web browser that piggybacks on existing commercial ones.”).

I. A PHOSITA would have no Motivation to Combine Pad++ with an Existing Browser on a PDA or Mobile Phone, and there would be no Reasonable Expectation of Success

Appellant argues (Appeal Brief: pp. 45-46) that Dr. Grimes alleges Pad++ could have been combined with an existing browser, without identifying any existing browser Pad++ could be combined with. As previously noted above, Appellant notes that Scott Forstall testified that you couldn’t just take a web browsing solution for a desktop and just port it to a mobile phone. Appellant argues that neither the present rejections nor the IPR Decisions provide any details on

Art Unit: 3992

how the prior art (Pad++ with an existing browser on a PDA) would be combined. Appellant argues that even if Pad++ could have been combined with an existing browser on a PDA or mobile phone, the combination would have been unusable (i.e., inoperable), useless, and thus the opposite of a useful improvement.

The Examiner respectfully disagrees with Appellant. The Examiner notes that the testimony of Scott Forstall (see: Exhibit H – Forstall Testimony & Exhibit F – Reinman Declaration, Paragraph 70: “Mr. Forstall testified that one could not just take a web browser solution for a desktop and just port it to a mobile phone”) was previously before the PTAB, and as noted above, the PTAB determined that the Pad++ browser was motivated to be and could be ported to PDAs at the time of the invention (see: Final Written Decision, pp. 18-19: “We are not persuaded...that one would not be motivated to port Bederson’s Pad++ browser...because of technical difficulties resulting from limited computing capacity and system incompatibilities”, p. 23: “Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers, and states that it was being designed for use on devices with small screens, such as PDAs...would be motivated to combine”).

Additionally, as noted above, the Pad++ references specifically teach that the Pad++ browser was designed to operate on PDAs (see: Bederson-4, p. 1128: “continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems around the world”; Bederson-5, p. 5: “Pad++...designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)”) and was noted to be especially attractive for systems which have small screens, such as handheld computers (i.e., PDAs) (see:

Art Unit: 3992

Benderson-5, p. 6: “provide effective access to a large structure of information on a much smaller display”; Pad++ Tour, pp. 140-141: “Pad++...windowing system based on zooming. Your whole desktop could be zoomable. This seems especially attractive for systems which have small screens, such as handheld computers (i.e., PDA’s)”). Thus the Pad++ references indisputably suggest that the Pad++ browser can and should be applied to PDAs. Further, while Ben Bederson admitted he didn’t implement Pad++ on a PDA, he also admitted that “conceptually, yes, it should run” on PDAs and even if it was running on a much slower machine “it might operate without errors and do all the things it was supposed to do except...it would be slow” (see: Exhibit G - Bederson Testimony, p. 181: “yes, it should run...don’t have any more information to know how easy or hard”, p. 293: “might operate without errors...would be slow”).

Regarding combining or modifying the Pad++ browser in view of existing browsers, the Examiner notes that the Pad++ reference itself suggests such a combination (see: Pad++ Tour, p. 86: “A zooming version of Mosaic and Netscape? This is not such a strange idea...the web browser we show here is still fairly primitive, you imagine a much more advanced version”, pp. 139-140: “We build what was at the time a cutting edge browser - with HTML 1.0...Instead, we are taking advantage of the new features of HTML by building a new generation Web browser that piggybacks on existing commercial ones”). Even the disclosure of the ‘926 patent teaches that it was common practice that certain existing web browsers provided their open source code so that it is was readily available for use by developers for conventional HTML-based Web content processing (see: ‘926 patent, column 17, lines 31-41: “recognized by those skilled in the art, the functionality...commonly performed by conventional browsers...performed by the Mozilla rendering engine, which comprises open source code....for use by developers”). Thus,

Art Unit: 3992

Appellant's general overarching argument that it would be too difficult for a PHOSITA to port Pad++ or a modified version of Pad++ to a PDA at the time of the invention is not persuasive. Said argument has previously been considered by the PTAB and was rejected. The limitations in the hardware capabilities of PDAs do not preclude the appropriate combination of PDA references with the Pad++ references for the various reasons previously discussed in this Examiner's Answer.

J. A Prior Art Reference must be Considered in its Entirety

Appellant argues (Appeal Brief: pp. 46-47) that certain features disclosed in the Pad++ references lead away from the claimed invention of the '926 patent, and/or otherwise could not be used to obtain the claimed invention.

The Examiner respectfully disagrees with Appellant. The Examiner notes that a vast majority of the listed teachings/features (e.g., Operating System Type, Text Rendering, TrueType Font Support, HTML Compliance, Graphics Library, Used in Commercial Applications, etc.) for the '926 patent are not actual claim limitations. Nothing in the disclosure of the Pad++ references teaches away (i.e., criticizes, discredits, or otherwise discourages) from the claimed invention of the '926 patent. In fact, as noted above, the Pad++ references specifically teach motivation for the Pad++ browser to operate on PDAs (see: Bederson-4, p. 1128: "continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems around the world"; Bederson-5, p. 5: "Pad++...operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants)", p. 6: "provide

Art Unit: 3992

effective access to a large structure of information on a much smaller display”; Pad++ Tour, pp. 140-141: “Pad++...windowing system based on zooming. Your whole desktop could be zoomable. This seems especially attractive for systems which have small screens, such as handheld computers (i.e., PDA’s)”).

X. Issues 5 and 9: Claim 29 is Patentably Distinct

Appellant argues (Appeal Brief: pp. 47-50) that in order to meet the claim limitation, in the context of claim 1 when properly construed, the browser rendering engine would need to meet the conformance requirements defined for CSS 2.0 (CSS2), which is a 338 page specification. Appellant argues that in the Bray reference there is no teaching or enabling disclosure regarding how to implement CSS in a Web browser rendering engine and that it would have been outside the capabilities of a PHOSITA to add CSS support to Pad++. Appellant further argues that it would take well in excess of 2 man years of development for a PHOSITA to add CSS support to Pad++ to enable Pad++ to render Web pages with CSS and meet the CSS conformance requirements. Additionally, Appellant argues that the CSS aspects of claim 29 are fully enabled through the disclosed use of the Mozilla (Gecko) rendering engine, which had support for HTML 4/4.01.

The Examiner respectfully disagrees with Appellant. Claim 29 merely requires wherein the HTML-based content includes CSS content defining layout and presentation attributes for the Web page. The Examiner notes that claim 29 does not specifically require, as suggested by Appellant in order to meet the claim limitation, meeting the conformance requirements defined

Art Unit: 3992

for CSS 2.0 (CSS2). At the time of the invention CSS was a notoriously well-known format that was compatible with HTML documents. Thus the motivation to combine the Pad++ references with Bray is clear (i.e., to enable delivery of CSS). More specifically, as discussed above, Pad++ broadly teaches a zoomable Web browser that “reads hypertext written in the Hypertext Markup Language (HTML), the language used to describe hypertext documents used by WWW browsers...Pad++ interface for accessing hypertext is similar to traditional systems, but zooming mechanisms are employed.” Bray more narrowly teaches Web browsers that can process HTML-based content that comprises cascading style sheet (CSS) data (e.g., “thing to do is to send the XML to a browser with a stylesheet, today in CSS...has a bunch of advantages...cuts down on the amount of data you have to transmit”, “one official, approved, stable...standard for stylesheets...CSS for short...CSS 1...CSS 2”, “rendering XML with CSS is nice”, etc.). Thus one of ordinary skill in the art would have been motivated to modify the Pad++ zoomable Web browser to accommodate the newer extensible markup standard, XML, along with the formatting standard CSS. The evolving markup language and stylesheets were known in the art and used to enhance the delivery of content on the new web browsers.

The Examiner is not persuaded by Appellant’s enablement and undue experimentation argument. The amount of experimentation that is considered reasonable will vary depending on the facts and circumstances of individual cases. In the instant case, as noted above, the compatibility between CSS and HTML documents was notoriously well-known at the time of the invention. While the Pad++ references explicitly teach how to parse and render HTML web pages, the Bray reference taught the CSS standard (see: Bray, p. 4: “there is only one official...production-quality standard for stylesheets...CSS for short...around since December

Art Unit: 3992

1996”) and how to tell your browser to use stylesheets (see: Bray, p. 3: “If you want to use stylesheets, you’ll have to tell the browser...here’s an example”). Appellant’s argument of “well in excess of 2 man years of development” is additionally found unpersuasive by the fact that the ‘926 patent is as detailed when discussing how to implement CSS in a Web browser as the Bray reference. The disclosure of the ‘926 patent merely mentions the utilization of cascading style sheets (CSS) in passing (see: ‘926 patent, column 2, lines 36-41, column 8, lines 50-52: “In Brief, HTML translator 58 translates...cascaded style sheet (CSS) layout content into scalable vector representation”). To the extent that the required implementation of CSS in claim 29 is fully enabled through the disclosed use of the Mozilla (Gecko) rendering engine in the ‘926 patent, the Bray reference is clearly similarly enabling via the disclosed use of Microsoft Internet Explorer 5 (IE5) which supports versions of the CSS and HTML standards (see: Bray, p. 1: “main focus of this article is publishing and browsing...in a standard-compliant way using IE5”, p. 3: “If you want to use stylesheets, you’ll have to tell your browser...here’s an example”, p. 4: “I found the IE’s CSS handling pretty good: i.e., everything I tried worked more or less first time...**worth your while to grab the CSS stylesheet that’s being used here and have a look at it; it’s not rocket science**”)(emphasis added).

XI. Issue 10: Claims 5 and 32: Real-Time Zooming

Appellant argues (Appeal Brief: pp. 50-52) that Patent Owner defined the meaning of “real-time” to mean the operation is performed in a few seconds or less and thus the Examiner’s assertion that “the limitation “real-time” is not narrowly defined (i.e., broad) in the claims,” is without merit. Appellant argues that the rejection of claims 5 and 32 is based on the bare

Art Unit: 3992

assertion that the zooming performance of Pad++ if implemented on a PDA such as the Nokia Communicator 9000, Apple Newton and Palm VII would have been sufficient to support the real-time zooming performance described in the Pad++ references. Appellant argues that the motivation is premised on the false statement that Pad++ was being designed for use on devices with small screens, such as PDAs, and is not based on any testing of Pad++, consideration of the design and source code of Pad++, nor any analysis of processor capabilities on PDAs. Appellant argues that performance of Pad++ on 200 MHz Pentium PC with a numeric processor enabled provides insufficient evidence to how Pad++ would perform on any other PDA or mobile phone. Appellant further asserts that overwhelming factual evidence has been provided showing that the real-time limitation could not be met.

The Examiner respectfully disagrees with Appellant. The Examiner notes that Appellant's purported definition of real-time, which is based on the prior prosecution history, appears to broader than the non-limiting construction detailed in the specification of the '926 patent (see: '926 patent, column 4, line 56-column 5, line 15: "fraction of a second. There is no perceptible wait for the user as the file is resized, or the window is repositioned"). Thus, Appellant's broader definition from the prosecution history will not be applied over what was taught in the specification of the '926 patent. Regardless, the definition does not overcome the teachings of Pad++ which clearly show real-time zooming operations (see: Bedernson-4, p. 1104: "Maintaining smooth real-time interaction", p. 1108: "built Pad++...and meet our goal of smoothly animated pans and zooms"; Bederson-5, p. 23: "successfully...maintained rates of about 10 frames per second...code to render zoomed images has been very carefully optimized

Art Unit: 3992

and allows real-time zooming”; Programmer’s Guide, p. 8: “Pad++ supports real-time zooming”).

As noted above, the Examiner does not agree that the motivation was premised on a false statement related to Pad++ being designed for use on devices with small screens, such as PDAs. The Examiner again notes that the reference platform of a 200 MHz Pentium Pro is not a formal requirement of the Pad++ browser (see: Bederson-4, p. 1103: “It is not meant as a formal definition of the requirements of a ZUI”) and thus the Pad++ references are not explicitly restricted by the processing capabilities and memory requirements as noted by Appellant via their submitted testing evidence. See section IX.E above wherein the additionally flexibility of Pad++ is discussed (e.g., Bederson-5, p. 23: “successfully loaded over 600 000 objects...maintained interactive rates of about 10 frames per second...Animations and zooming...**independent of processor speed, scene complexity, and window size**...rendering more or fewer frames, as time allows”)(emphasis added) with regard to Appellant’s additional testing evidence. The PTAB heard similar arguments (see: Exhibit F – Reinman Declaration: Paragraphs 66-71: “a weaker device which would not have the processing capability or memory capacity to handle Pad++ calculations...pad++ on Zaurus would only serve to exacerbate these existing problems in meeting the real-time constraint...too expensive to allow Pad++ to satisfy its real-time constraints”) and reached a similar determination in the Final Written Decision (see: Final Written Decision, pp. 18-19: “We are not persuaded...that one would not be motivated...because of technical difficulties resulting from limited computing capacity and system incompatibilities”, p. 23: “Pad++ demonstrates the concept, suggests it could be applied to the Netscape and Mosaic browsers, and states that it was being designed for use on devices

Art Unit: 3992

with small screens, such as PDAs...would be motivated to combine”) that the Pad++ browser was motivated to be and could be ported to PDAs at the time of the invention.

XII. Issue 11: Claims 12, 14, and 42 are Patentably Distinct

Appellant argues (Appeal Brief: pp. 52-62) that HTML-based columns, images, and paragraphs are not substantially equivalent elements and further notes that the HTML 4.01 specification defines how HTML IMG, PARAGRAPH, TABLE, DIV, and SPAN elements are to be differently processed and formatted. Appellant argues that in order to zoom on a column/paragraph and fit the content corresponding the selected column/paragraph across the touch-sensitive display, you would need a bounding box for the column or paragraph. Appellant opines that the Examiner has not addressed this requirement. Further, Appellant argues that each of the Pad++ graphical features are visible features, and thus Pad++ provides no teaching or suggestion of using invisible HTML container elements.

Additionally, Appellant argues that Pad++ does not enable a user to zoom on any HTML object on a Web page via a user input that selects the HTML object (e.g., column, image, paragraph, etc.). Appellant provides further evidence, via the Itsy prototype pocket computer, on the difficulty of implementing a desktop interface on a PDA and opines that Pad++ provides no teaching with respect to how to implement three-button mouse functionality with a single mouse button. Appellant argues that the inventors of the ‘926 patent solved this problem using a tap-based interface, under which tapping on a column, image, or paragraph zoomed in on the column, image or paragraph by detecting the bounding box for the column, image, or paragraph and scaling and offsetting the view of the web page such that the width of the bounding box was

Art Unit: 3992

fit across the width of the screen. With respect to the tap-to-zoom operation being performed automatically in response to tapping on a column (or paragraph or image), the Appellant notes that claim 1 requires instructions “that when executed by a processor” and claim 12 requires “execution of the instructions...by tapping on the column via the touch-sensitive display.”

Additionally, Appellant notes that the meaning of “fit across the display” was defined by stating it “is intended to cover the rendering of a column (in response to a zoom to column user input) such that the column is rendered to span the width of the applicable browser display area.”

Appellant argues that the Examiner ignores the “selected” limitation and improperly equates “fills most of the screen” with fit across the display.

Regarding paragraphs (e.g., <p>), Appellant additionally argues that Pad++ could not support zooming on a paragraph because: (1) Pad++ does not generate a bounding box for paragraphs; (2) Pad++ does not extract, generate or maintain an HTML paragraph object; and (3) All lines of text in a paragraph except for the last line would span the width of the page for any paragraph longer than a single line.

The Examiner respectfully disagrees with Appellant. Regarding HTML-based columns, images, and paragraphs, the Examiner is not arguing that said elements are substantially equivalent because they perform the same function, but rather they are considered substantially equivalent because each are notoriously well-known HTML elements with well-established formats and functions. As implied by the disclosure of the ‘926 patent, processing these HTML elements was known to developers and commonly practiced by conventional browsers (see: ‘926 patent, column 17, lines 31-41: “recognized by those skilled, the functionality performed...are

Art Unit: 3992

commonly performed by conventional browsers...Mozilla rendering engine...for use by developers”). The PTAB reached a similar determination regarding HTML tags in general in the Final Written Decision (see: Final Written Decision, p. 26: “even if Bederson did not implement all available HTML tags, the selection of which tags to implement is a matter that would be well within the knowledge and abilities of a person of ordinary skill in the art, and would be obvious”). The PTAB’s determination clearly indicates that selecting and implementing any available HTML tag would be well within the knowledge and abilities of a PHOSITA.

Additionally, the Examiner notes that the claims that require zooming in on columns and paragraphs (e.g., claims 12 and 14) simply do not require any of special processing of the claimed columns/paragraphs as argued by Appellant. The claims do not explicitly teach that the paragraph or column is “selected” via its HTML tag (e.g., <p>, <col>, etc.) in the HTML-based Web content and certainly don’t require “a bounding box for the column or paragraph” or automatically “detecting the bounding box for the column, image, or paragraph and scaling and offsetting the view of the web page such that the width of the bounding box was fit across the width of the screen.” The claims (e.g., claim 14) merely require “enabling a user to view a paragraph of the Web content...by tapping on the paragraph...the display is re-rendered...displayed to fit across a width of a display area.” Therefore, Appellant’s specific arguments regarding these supposedly missing features are not persuasive. The Examiner notes that if claims 12, 14, and 42 captured all aspects argued by Appellant they may overcome the teachings in the Pad++ references, but the claims are simply not that limiting.

The Examiner is also not persuaded that it would be difficult to implement the inputs of Pad++ on a PDA. As noted above, the Pad++ references specifically teaches that the Pad++

Art Unit: 3992

browser was designed to operate on PDAs (see: Bederson-4, p. 1128: “continuing goal has been to make Pad++ as portable as possible...and run Pad++ on all kinds of systems around the world”; Bederson-5, p. 5: “Pad++...designed to operate on platforms ranging from high-end graphics workstations to PDAs (Personal Digital Assistants”). Additionally, Ben Bederson admitted that providing touchscreen based inputs to a computer system running Pad++ would have been within the skill of a PHOSITA (see: Exhibit G - Bederson Testimony, p. 182: “no bearing on the question of how easy or hard it would have been to have Pad++ run on a PDA...could use a touchscreen as input to a computer system for Pad++...clearly yes...not an interesting question. The answer was obviously yes...any input device could be used...that ran software such as Pad++”).

The Pad++ references clearly teach wherein the zoomable web browser could process HTML documents that include many characters/text, line segments, images, bullet items, and horizontal rules (see: Bederson-1, p. 24: “Pad++ can read in...(HTML)”; Bederson-3, p. 264: “includes text...rules”; Bederson-4, p. 1104: “HTML page composed of many characters...images”). Additionally, the Pad++ references taught wherein the zoomable web browser could (1) place objects (e.g., HTML Web pages) at any position and at any scale (see: Bederson-3, 262: “object can be placed at any position...scale”; Bederson-4, p. 1128: “All graphical elements in Pad++ are instances of objects...ability to be moved and scaled”; Bederson-5, p. 14: “Clicking on the page...pan and zoom so that the page fills most of the screen”); (2) allow the user to zoom in and out of pages, enabling explicit control of how much context is viewed at any time (see: Bederson-1, p. 23: “Zooming and panning are the primary methods of navigation in Pad++”: Bederson-3, p. 262: “Zooming can involve simple geometric

Art Unit: 3992

scaling...allows the user to zoom in and out of pages, enabling explicit control”; Bederson-5, p. 7: “Pad++ supports...direct manipulation panning and zooming”; Pad++ Tour, p. 89: “Here is a zoomed in view of the document”, p. 93: “Simply zoom in on the sub-document to read it”); and (3) permit authors to manipulate and interact with any web page element (see: Bederson-3, p. 264: “manipulate and interact...any web page element”). Therefore, the zoomable web browser of Pad++ could zoom and/or scale any column or paragraph of text displayed in an HTML document such that it met Appellant’s non-limiting definition of “fit across the display” (see: Final Written Decision, pp. 25-26: “Bederson disclosed...using Pad++ to zoom in on a portion of Bederson’s home web page (an HTML document) selected by a user...Thus, Bederson discloses zooming on a portion of a web page”). Thus, a user of the Pad++ HTML browser could manually zoom in on (i.e., select) any portion of the HTML document they desired, including any column or paragraph of text, such that the column or paragraph of text “fit across the display.”

For example, the Pad++ Tour reference meets the limitation by more specifically showing dynamically zooming in on a paragraph of text (see: Pad++ Tour, p. 89) of an initially displayed HTML document (see: Pad++ Tour, p. 87), wherein said paragraph of text is displayed to fit across the display. In this case the Examiner agrees that the user of Pad++ Tour is zooming in on a portion of the entire document in order to provide a zoomed in view of the paragraph of text now fit across the display. However, the claims do not preclude this method and do not require only the paragraph or column be zoomed in on, rather the claim merely requires “the display is re-rendered such that at content corresponding to the selected paragraph is displayed to fit across the width of the display area.”

XIII. Issue 12: Claims 21 and 51

Appellant notes (Appeal Brief: pp. 63-65) that Pad++ includes a graphics renderer, which is used to render Pad++ objects to the screen. Appellant argues that the single-pass parser is separate and apart from the Pad++ graphics renderer and that a PHOSITA would not consider the single-pass parser of Pad++ to be a rendering engine within the well-known meaning of the term at the time of the invention. Appellant argues that the claimed rendering engine was much more sophisticated than the simple single-pass parser implemented in Pad++, and was used to process and interpret the original page layout, functionality, and design of a Web page as defined by its HTML-based Web content, including markup up content (HTML 4/4.01 tags, XML, image files) and formatting information (CSS).

The Examiner respectfully disagrees with Appellant. The Pad++ references explicitly teach wherein Pad++ comprises a rendering engine (see: Bederson-1, p. 24: “Pad++ can read in hypertext files written in Hypertext Markup Language (HTML)”); Bederson-4, p. 1128: “Pad++ consists of several components...Renderer: the render performs all the rendering on the screen...specify translation and scale”; Bederson-5, pp. 12-13: “Pad++ reads hypertext written in...(HTML)...Many different HTML documents loaded in Pad++”). Absent an explicit definition (how much more sophisticated does the single-pass parser of Pad++ need to be?) the Examiner believes a rendering engine is merely a program that renders marked up content and/or formatting information. The Examiner notes that Appellant’s citation to the Wikipedia definition of a web browser engine merely states that a web browser engine is a “computer program that

Art Unit: 3992

renders web pages” and that rendering “is the process of transforming an HTML document and its associated resources...into a visual representation on an electronic display.” Clearly, the Pad++ references meet this requirement (see: e.g., Bederson-1, p. 24: “A screen dump showing a series of World Wide Web home pages loaded into Pad++”; Bederson-4, p. 13: “Many different HTML documents loaded in Pad++”; Pad++ Tour, p. 86: “a simple Pad++ web browser”, p. 87: “Here is a screen snapshot showing Pad++ displaying an HTML document”).

Even in light of Appellant’s definition, Pad++ clearly teaches a Pad++ web browser used to process and interpret the original page layout, functionality, and design of a Web page as defined by its HTML-based content (see: Pad++ Tour, p. 86: “Pad++ application we look at is a simple Pad++ web browser”, p. 87: “screen snapshot showing Pad++ displaying an HTML document”, p. 89: “Here is a zoomed in view of the document”, p. 91: “relevant document is loaded into Pad++”). The Examiner notes the PTAB reached a similar determination that Pad++ taught a rendering engine that was used to process and interpret the original page layout, functionality, and design of a Web page as defined by its HTML-based content (see: Final Written Decision, p. 22: “both Zaurus and Bederson disclose preserving the layout, functionality, and design of the part of the web page that is translated”; p. 24: “Pad++ includes a renderer that performs all rendering”). The Examiner further notes that the level of complexity argued by Appellant (i.e., Pad++ could not process HTML 4/4.01 tags, CSS, and XML) is not a limitation of claims 21 and 51.

Art Unit: 3992

Conclusion

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/ADAM L BASEHOAR/
Primary Examiner, Art Unit 3992

Conferees:

/JOSHUA CAMPBELL/
Primary Examiner, Art Unit 3992

/ALEXANDER KOSOWSKI/
Supervisory Patent Examiner, Art Unit 3992

Requirement to pay appeal forwarding fee. In order to avoid dismissal of the instant appeal in any application or *Ex Parte* reexamination proceeding, 37 CFR 41.45 requires payment of an appeal forwarding fee within the time permitted by 37 CFR 41.45(a), unless Appellant had timely paid the fee for filing a brief required by 37 CFR 41.20(b) in effect on March 18, 2013.