

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FOLEON INC.,

Petitioner

v.

TURTL SURF & IMMERSE LIMITED

Patent Owner

IPR2026-00428

Patent No. 11,755,816

EXHIBIT 1003

**DECLARATION OF DR. BENJAMIN B. BEDERSON
IN SUPPORT OF PETITION FOR INTER PARTES REVIEW OF
U.S. PATENT NO. 11,755,816**

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Background and Qualifications	1
IV.	Relevant Patent Law and Legal Standards	11
A.	Date of Invention.....	12
B.	Person of Ordinary Skill in the Art (POSITA).....	12
C.	Claim Construction Standard	13
D.	Anticipation	14
E.	Obviousness.....	14
F.	Standard of Proof.....	20
V.	Technological Background.....	20
A.	Modular Document Generation.....	20
B.	Dynamic Web Technologies	22
VI.	Overview of the '816 Patent.....	24
VII.	Summary of the Prior Art	33
A.	Harrop.....	33
B.	Hailey	36
C.	Davis.....	38
D.	Lo.....	40
E.	Ohashi.....	41
VIII.	GROUNDS FOR UNPATENTABILITY	45

A.	Ground 1: Claims 1-4, 7, 11-14, and 17 are rendered obvious by Harrop, and Hailey	45
1.	Reasons to combine	45
2.	Claim 1	48
3.	Claim 2: The method of claim 1, wherein the reference is created based at least in part on the personalization parameter.....	64
4.	Claim 3: The method of claim 1, wherein the reference is used to update the one or more modular items in the derivative document.....	65
5.	Claim 4: The method of claim 1, wherein the derivative document comprises a second set of modular items that do not have embedded rules.....	68
6.	Claim 7: The method of claim 1, wherein the personalization parameter is received via a graphical user interface or via an application programming interface.....	70
7.	Claim 11	71
8.	Claim 12: The system of claim 11, wherein the reference is created based at least in part on the personalization parameter.....	74
9.	Claim 13: The system of claim 11, wherein the reference is used to update the one or more modular items in the derivative document.....	74
10.	Claim 14: The system of claim 11, wherein the derivative document comprises a second set of modular items that do not have embedded rules.....	74
11.	Claim 17: The system of claim 11, wherein the personalization parameter is received via a graphical user interface or via an application programming interface.....	74

B.	Ground 2: Claims 8-9 and 18-19 are rendered obvious by Harrop, Hailey, and Lo	74
1.	Reasons to Combine	75
2.	Claim 8: The method of claim 1, further comprising upon receiving the personalization parameter, rendering the derivative document in real-time.	77
3.	Claim 9: The method of claim 8, wherein the derivative document is rendered within a web browser by executing a script using a content renderer.	78
4.	Claim 18: The system of claim 11, wherein the one or more processors are further configured to render rendering the derivative document in real-time upon receiving the personalization parameter.	79
5.	Claim 19: The system of claim 18, wherein the derivative document is rendered within a web browser by executing a script using a content renderer.	80
C.	Ground 3: Claims 10 and 20 are rendered obvious by Hailey, Harrop and Davis.....	80
1.	Technical Background - User Tracking.....	80
2.	Reasons to Combine	81
3.	Claim 10: The method of claim 1, further comprising embedding a tracking feature in the derivative document to collect data indicative of user interaction with the derivative document.....	85
4.	Claim 20: The system of claim 11, wherein the derivative document is embedded with a tracking feature to collect data indicative of user interaction with the derivative document.....	86
D.	Ground 4: Claims 5- 6 and 15-16 are rendered obvious by Hailey, Harrop and Ohashi.....	86

1.	Reasons to Combine	86
2.	Claim 5: The method of claim 4, further comprising converting the derivative document into a master document by embedding rules into at least one of the second set of modular items.....	88
3.	Claim 6: The method of claim 5, further comprising removing the reference to break the link between at least one modular item and the derivative document.....	90
4.	Claim 15: The system of claim 14, wherein the one or more processors are further configured to convert the derivative document into a master document by embedding rules into at least one of the second set of modular items.	91
5.	Claim 16: The system of claim 15, wherein the one or more processors are further configured to remove the reference to break the link between at least one modular item and the derivative document.....	91
IX.	CONCLUSION.....	92

EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 11,755,816 to Mason
1002	Prosecution History of U.S. 11,755,816
1004	<i>Curriculum Vitae</i> of Benjamin B. Bederson
1005	U.S. Patent Publication No. 2005/0080814 A1 to Hailey et al.
1006	International Publication No. WO 03/075191 A1 to Harrop et al.
1007	U.S. Patent No. 5,796,952 to Davis et al.
1008	U.S. Patent No. 7,844,896 (“Ohashi”)
1009	U.S. Patent Publication No. 2005/0094205 A1 to Lo et al.
1010	U.S. Patent No. 4,712,174 to Minkler (“Minkler”)
1011	U.S. Patent 5,956,687 to Wamsley et al. (“Wamsley”)
1012	“How Users Read on the Web”
1013	<i>Information on RFC 3986</i> , RFC Editor,
1014	“How to use the Mail Merge feature in Word to create and to print form letters that use the data from an Excel worksheet,” Microsoft
1015	“How People Read Online: The Eyetracking Evidence”
1016	Rong Chen, Anne Rose, and Benjamin B. Bederson. 2009. How people read books online: mining and visualizing web logs for use information. In Proceedings of the 13th European conference on Research and advanced technology for digital libraries (ECDL'09). Springer-Verlag, Berlin, Heidelberg, 364–369
1017	NYTimes, Google Acquires Urchin Software, March 29, 2005

Exhibit No.	Description
1018	Nataliia Bielova. 2017. Web Tracking Technologies and Protection Mechanisms. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 2607–2609.
1019	Tim Berners-Lee et al., <i>The World-Wide Web</i> . 37 Communications of the ACM 76 (1994).
1020	Archived webpage regarding Web Parameter Tampering (April 27, 2020).
1021	Archived webpage regarding Information exposure through query strings in url description (April 22, 2020).
1022	U.S. Patent 8,661,342 to Lu et al. (“Lu”).
1023	<i>Foleon Inc. v. Turtl Surf & Immerse Limited</i> , IPR2025-01525, Paper 6 (PTAB Jan. 28, 2026).
1024	<i>Foleon Inc. v. Turtl Surf & Immerse Limited</i> , IPR2025-01525, EX2001 (PTAB Jan. 28, 2026).
1025	[RESERVED]
1026	Wayback Machine Archive of Alex Hopmann, “Story of XMLHTTP”, (June 17, 2006)
1027	Michael Singer, “Google Testing Free Webmail” (April 1, 2004)
1028	Bret Taylor, “Mapping your way” (February 8, 2005)
1029	Jesse James Garrett, “Ajax: A New Approach to Web Applications”, February 18, 2005
1030	https://en.wikipedia.org/wiki/JavaScript#History
1031	Document Object Model (DOM) Level 1 Specification (1998)
1032	Peter-Paul Koch, Digital Web Magazine, <i>The Document Object Model: an Introduction</i> , January 24, 2014

TABLE OF CLAIMS

Claim Limitation	Claim Language
1[PRE]	1. A method for creating an interactive document, the method comprising:
1[A]	(a) providing a first document comprising a first set of modular items, wherein at least a subset of the first set of modular items have embedded rules;
1[B]	(b) receiving a personalization parameter for generating a derivative document;
1[C]	(c) selecting one or more modular items from the first subset of modular items to be assembled into the derivative document, wherein the one or more modular items are selected when the personalization parameter meets the embedded rules in the respective one or more modular items; and
1[D1]	(d) creating a reference for linking the one or more modular items to the derivative document.
2	2. The method of claim 1, wherein the reference is created based at least in part on the personalization parameter.
3	3. The method of claim 1, wherein the reference is used to update the one or more modular items in the derivative document.
4	4. The method of claim 1, wherein the derivative document comprises a second set of modular items that do not have embedded rules.

5	5. The method of claim 4, further comprising converting the derivative document into a master document by embedding rules into at least one of the second set of modular items.
6	6. The method of claim 5, further comprising removing the reference to break the link between at least one modular item and the derivative document.
7	7. The method of claim 1, wherein the personalization parameter is received via a graphical user interface or via an application programming interface.
8	8. The method of claim 1, further comprising upon receiving the personalization parameter, rendering the derivative document in real-time.
9	9. The method of claim 8, wherein the derivative document is rendered within a web browser by executing a script using a content renderer.
10	10. The method of claim 1, further comprising embedding a tracking feature in the derivative document to collect data indicative of user interaction with the derivative document.
11[PRE]	11. A system for creating an interactive document, the method comprising:

11[A]	A memory for storing a set of software instructions, and one or more processors configured to execute the set of software instructions to:
11[B]	(a) provide a first document comprising a first set of modular items, wherein at least a subset of the first set of modular items have embedded rules;
11[C]	(b) receive a personalization parameter for generating a derivative document;
11[D]	(c) select one or more modular items from the first subset of modular items to be assembled into the derivative document, wherein the one or more modular items are selected when the personalization parameter meets the embedded rules in the respective one or more modular items; and
11[E]	(d) create a reference for linking the one or more modular items to the derivative document.
12	12. The system of claim 11, wherein the reference is created based at least in part on the personalization parameter.
13	13. The system of claim 11, wherein the reference is used to update the one or more modular items in the derivative document.
14	14. The system of claim 11, wherein the derivative document comprises a second set of modular items that do not have embedded rules.
15	15. The system of claim 14, wherein the one or more processors are further configured to convert the derivative document into a master

	document by embedding rules into at least one of the second set of modular items.
16	16. The system of claim 15, wherein the one or more processors are further configured to remove the reference to break the link between at least one modular item and the derivative document.
17	17. The system of claim 11, wherein the personalization parameter is received via a graphical user interface or via an application programming interface.
18	18. The system of claim 11, wherein the one or more processors are further configured to render rendering the derivative document in real-time upon receiving the personalization parameter.
19	19. The system of claim 18, wherein the derivative document is rendered within a web browser by executing a script using a content renderer.
20	20. The system of claim 11, wherein the derivative document is embedded with a tracking feature to collect data indicative of user interaction with the derivative document.

I. INTRODUCTION

1. I, Benjamin B. Bederson, offer this declaration in support of the Petition for Inter Partes Review of U.S. Patent No. 11,755,816 (“the ’816 Patent”). I am over the age of 18, competent to make this declaration, and have personal knowledge of the facts set forth below. If called to testify, I could and would testify honestly, under oath, to the matters set forth herein.

2. I am not an attorney. I have not been asked to offer any legal opinions. I have been informed and understand the law to be applied for determining invalidity, which I explain in places below. I have applied the law told to me in developing my technical opinions in this Declaration.

3. All statements made in this declaration regarding the disclosures and teachings of the prior art references and/or the state of the relevant technology are from the perspective of a person of ordinary skill in the art (“POSITA”).

II. BACKGROUND AND QUALIFICATIONS

4. My curriculum vitae, which includes a more detailed summary of my background, experience, and publications, is attached as Exhibit 1004.

5. I received a Bachelor of Science degree in Computer Science with a minor in Electrical Engineering from Rensselaer Polytechnic Institute (“RPI”) in 1986. I received a Master of Science degree and a Ph.D. in Computer Science from New York University (“NYU”) in 1989 and 1992, respectively.

6. Since 1998, I have been a Professor of Computer Science at the University of Maryland (“UMD”), where I have had joint appointments at the Institute for Advanced Computer Studies and the College of Information Studies (Maryland’s “iSchool”) and am currently Professor Emeritus. I was also Associate Provost of Learning Initiatives and Executive Director of the Teaching and Learning Transformation Center from 2014 to 2018. I am a member and previous director of the Human-Computer Interaction Lab (“HCIL”), the oldest and one of the best known Human-Computer Interaction (“HCI”) research groups in the country.

7. I am also co-founder and co-director of the International Children’s Digital Library (“ICDL”), a web site launched in 2002 that provides the world’s largest collection of freely available online children’s books from around the world with an interface aimed to make it easy for children and adults to search and read children’s books online. I was also co-founder and Chief Scientist of Zumobi, Inc. from 2006 to 2014, a Seattle-based startup that is a publisher of content applications and advertising platforms for smartphones. I was also co-founder and prior Chief Technology Officer of Hazel Analytics from 2014 to 2023, a data analytics company. In addition, I have for more than 25 years consulted for numerous companies in the area of user interfaces, including EPAM, Lockheed Martin, Logitech, Microsoft, NASA Goddard Space Flight Center, the Palo Alto Research Center, and Sony.

8. For more than 30 years, I have studied, designed, and worked in the field of computer science and human-computer interaction. My experience includes 30 years of teaching and research, with research interests in human-computer interaction and the software and technology underlying today's interactive computing systems. This includes the design and implementation of user interfaces and software applications using a wide range of technologies including web servers and websites, databases, embedded systems, and mobile devices, including smart phones and PDAs.

9. At UMD, my research has been in the area of HCI, a field that relates to the development and understanding of computing systems to serve users' needs. Researchers in this field are focused on making universally usable, useful, efficient, and appealing systems to support people in their wide range of activities. My approach is to balance the development of innovative technology that serves people's practical needs. I have combined my hardware and software skills throughout my career in HCI research, building various interactive electrical and mechanical systems that couple with software to provide an innovative user experience. Example systems that I have built include Pad++ (1994 software for zoomable user interfaces), KidPad (1997 software to support children collaboratively writing stories), PadPrints (1998 software to graphically show interactive web browsing histories), PhotoMesa (2001 software for end users to

browse personal photos), Pocket PhotoMesa (2004 software for browsing of personal photos on a mobile device), DateLens (2002 software for end users to use their mobile devices to efficiently access their calendar information), LaunchTile (2005 “home screen” software for mobile devices to allow users to navigate apps in a zoomable environment), SpaceTree (2001 software for end users to efficiently browse very large hierarchies), ICDL (as described above), and StoryKit (a 2009 iPhone app for children to create stories).

10. Pad++ included a custom web browser that I built from scratch and distributed as open source software at the Pad++ web site. That web browser was described in numerous publications including “Pad++: A Zoomable Graphical Interface for Exploring Alternate Interface Physics.”¹ I implemented a rendering engine from scratch for that web browser which supported the version of HTML available at the time (HTML 1.0) as well as an embedded scripting language. That web browser included the ability to display embedded web pages that could be zoomed along with the rest of the pages. The following images are from that custom web browser that I built, which used a “zoomable user interface” so one page could be shown larger while other recently visited pages would be displayed smaller.²

¹ Bederson, B. B. & Hollan, J. D. (1994). Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics, ACM Conference on User Interface Software and Technology, 17-26.

² Bederson, B. B. et al. (1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics, 7, Journal of Visual Languages & Computing.

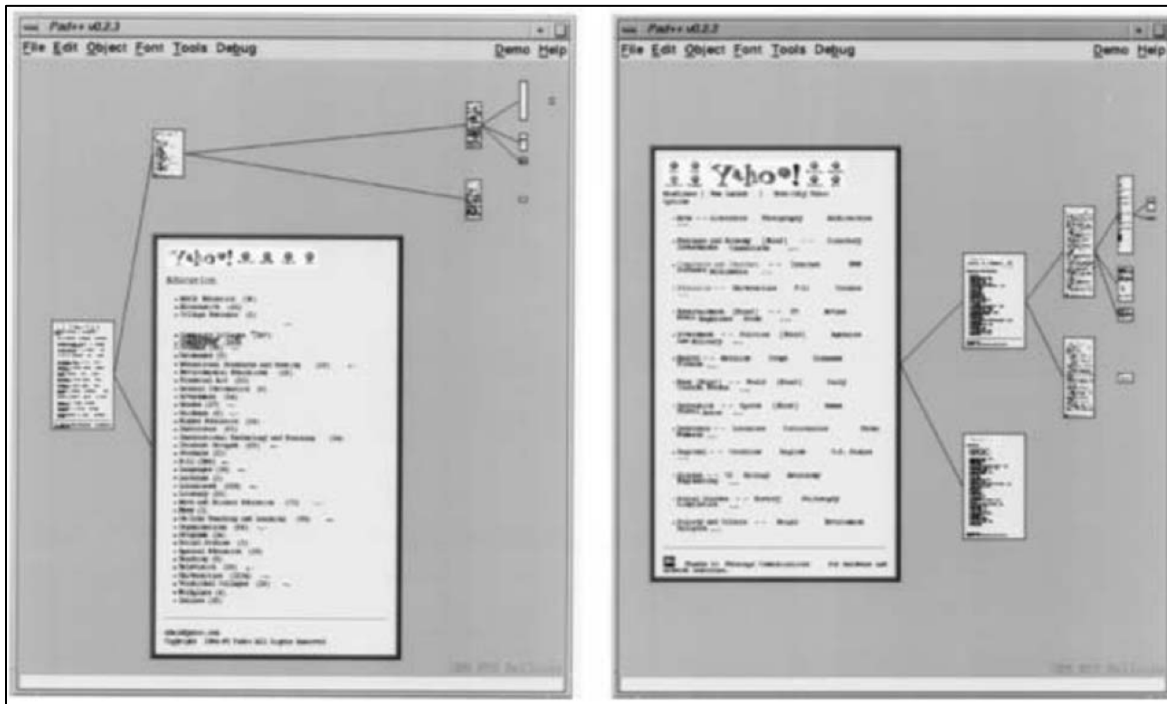


Figure 5 from “Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics”

11. By 1998, my team had combined Pad++ with Netscape Navigator (a standard web browser of the time) to offer a graphical web history, visually showing what pages a user had visited with an application we called “PadPrints.”³ We did this by implementing a web “proxy,” which was a standard way at the time to modify the built-in behavior of web browsers. Some proxies work by intercepting all web traffic coming from and going to a web browser. Once intercepted, the proxy can do whatever it wants with the traffic, caching it for better performance, simply

³ Bederson, B. B. et al. (1998). PadPrints: Graphical Multiscale Web Histories. In Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology (UIST '98). Association for Computing Machinery, New York, NY, USA, 121–122. <https://doi.org/10.1145/288392.288582>

passing the traffic through and monitoring it, modifying the data itself, or holding that connection and opening new connections. PadPrints' proxy worked by monitoring the traffic and letting PadPrints know what the user was doing so PadPrints could display the web history.

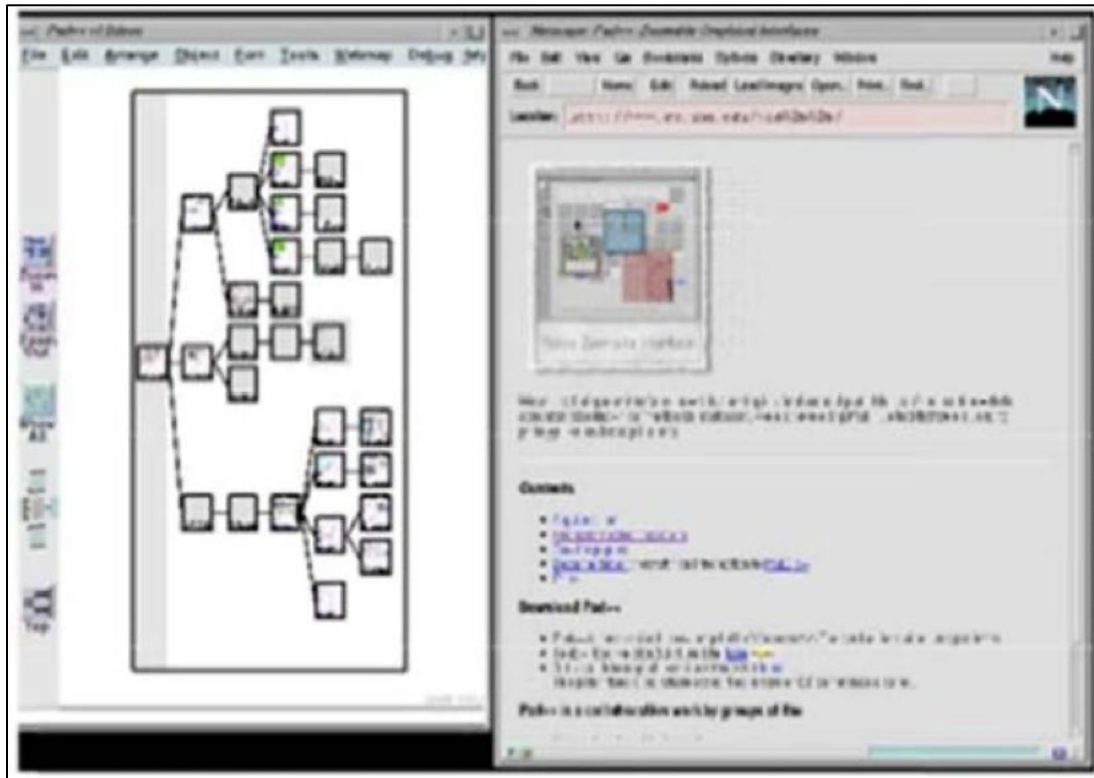


Figure 1 from PadPrints paper

12. At the International Children's Digital Library (ICDL), I have since 2002 been the technical director responsible for the design and implementation of the web site, www.childrenslibrary.org (originally at www.icdlbooks.org). In particular, I have been closely involved in designing the user interface as well as the software architecture for the web site since its inception in 2002.

13. The ICDL includes a display of search results that include thumbnails of book covers where 8 of the book covers are shown at a time. Pressing the right arrow uses AJAX web technology to dynamically retrieve and display the next 8 book covers from the server without having to reload the entire page. We switched from a similar version of this interface to this AJAX-based approach in about 2006.

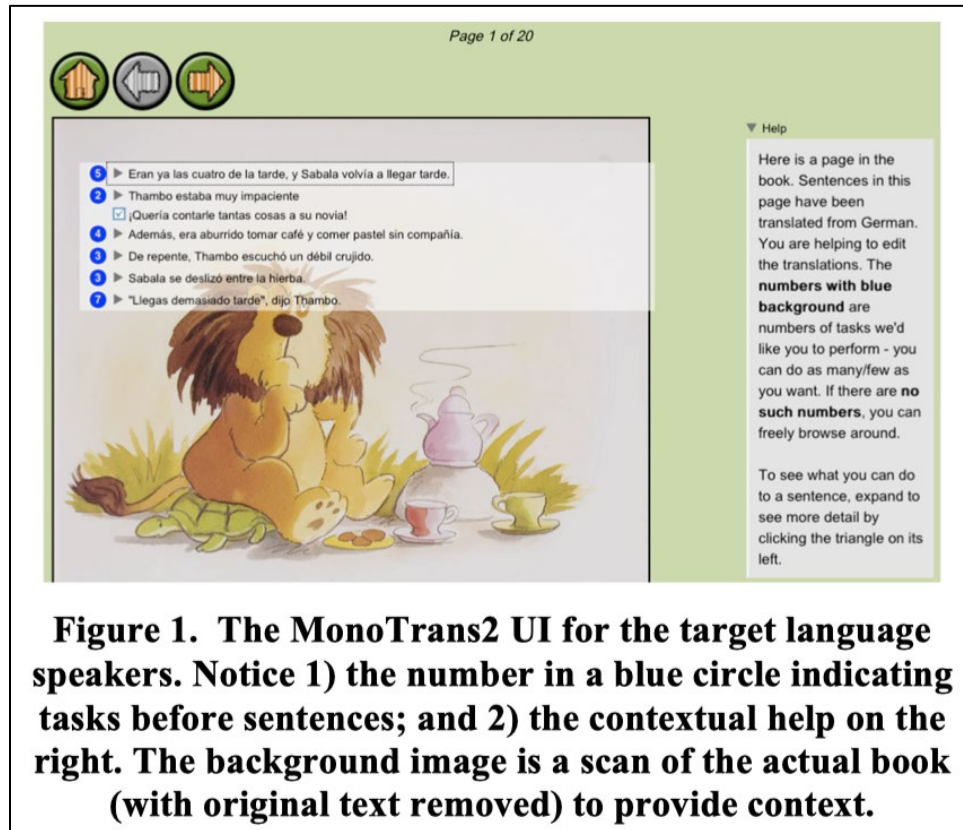


Screenshot taken February 19, 2017 by Benjamin B. Bederson from <http://www.childrenslibrary.org/icdl/SimpleSearchCategory?ids=25&langid=&pnum=1&cnum=1&text=&lang=English&ilang=English>

14. Beginning in the mid-1990s, I have been responsible for the design and implementation of numerous other web sites in addition to the ICDL. For example, I designed and built my own professional web site when I was an Assistant Professor of Computer Science at the University of New Mexico in 1995 and have continued to design, write the code for, and update both that site (which I moved to the University of Maryland in 1998, currently at <http://www.cs.umd.edu/~bederson/>) as well as numerous project web sites, such as Pad++, <http://www.cs.umd.edu/hcil/pad++/>.

15. I started a new web-based project in 2009 to bring people together on a website to collaboratively translate text for children's books from one language to another. This resulted in numerous publications, including a 2011 paper⁴ that described a second generation of this work. In the numerous versions of this project, we used several web-based collaboration techniques, including by synchronous and asynchronous collaboration – allowing people to see the work of others on their own views of our website either immediately as others were doing the work (synchronously) or later after they had completed the work (asynchronously).

⁴ Bederson, B. B. et al. (2011). MonoTrans2: a new Human Computation System to Support Monolingual Translation. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). Association for Computing Machinery, New York, NY, USA, 1133–1136. <https://doi.org/10.1145/1978942.1979111>



16. My work has been published extensively in more than 160 technical publications, and I have given about 100 invited talks, including 9 keynote lectures. I have won a number of awards including the Brian Shackel Award for “outstanding contribution with international impact in the field of HCI” in 2007, and the Social Impact Award in 2010 from Association for Computing Machinery’s (“ACM”) Special Interest Group on Computer Human Interaction (“SIGCHI”). ACM is the primary international professional community of computer scientists, and SIGCHI is the primary international professional HCI community. I have been honored by both professional organizations. I am an “ACM Distinguished Scientist,” which “recognizes those ACM members with at least 15 years of professional experience

and 5 years of continuous Professional Membership who have achieved significant accomplishments or have made a significant impact on the computing field.” I am a member of the “CHI Academy,” which is described as follows: “The CHI Academy is an honorary group of individuals who have made substantial contributions to the field of human-computer interaction. These are the principal leaders of the field, whose efforts have shaped the disciplines and/or industry, and led the research and/or innovation in human-computer interaction.” The criteria for election to the CHI Academy are: (1) cumulative contributions to the field; (2) impact on the field through development of new research directions and/or innovations; and (3) influence on the work of others. I have received two “Test of Time” awards from IEEE InfoVis for my 2001 and 2002 work on visualizing hierarchies, including those containing digital photos.⁵ I received the Janet Fabri Memorial Award for Outstanding Doctoral Dissertation for my Ph.D. work in robotics and computer vision.

17. I have appeared on radio shows numerous times to discuss issues relating to user interface design and people’s use and frustration with common technologies, web sites, and mobile devices. My work has been discussed and I have been quoted by mainstream media around the world over 120 times, including by

⁵ <https://ieevis.org/year/2021/info/awards/test-of-time-awards>; <https://ieevis.org/year/2022/info/awards/test-of-time-awards>.

the New York Times, the Wall Street Journal, the Washington Post, Newsweek, the Seattle Post-Intelligencer, the Independent, Le Monde, NPR's All Things Considered, New Scientist Magazine, and MIT's Technology Review.

18. I have designed, programmed, and publicly deployed dozens of user-facing software products that have cumulatively had millions of users. I am the co-inventor of 14 U.S. patents and 20 U.S. patent applications.

III. SUMMARY OF MATERIALS REVIEWED AND CONSIDERED

19. All of the opinions contained in this Declaration are based on the documents I reviewed and my knowledge and professional judgment. In forming the opinions expressed in this Declaration, while drawing on my experience and knowledge regarding graphical user interfaces, I read and considered the '816 Patent, its prosecution history, as well as any other material referenced herein.

20. My opinions are guided by my appreciation of how a person of ordinary skill in the art ("POSITA") would have understood the claims of the '816 Patent at the time of the filing, which I have been asked to assume is July 6, 2020.

IV. RELEVANT PATENT LAW AND LEGAL STANDARDS

21. I am not an attorney. I understand that in writing this Declaration I am obliged to follow existing applicable law. I have therefore been asked to apply the following legal principles to my analysis, and I have done so.

A. Date of Invention

22. I have been asked to use the date of the filing of the earliest application to which priority is claimed (*i.e.*, July 6, 2020) as the date of invention for purposes of my analysis.

B. Person of Ordinary Skill in the Art (POSITA)

23. I understand that my assessment of the claims of the '816 Patent must be undertaken from the perspective of what would have been known or understood by a person having ordinary skill in the art, reading the '816 Patent on its relevant filing date and in light of the specification and file history of that patent. I often refer to such a person as a POSITA in this Declaration.

24. I am informed and understand that various factors can be considered in determining a POSITA. I am informed and understand that those factors include: (1) the educational level of the inventor(s); (2) the type of problems encountered in the art; (3) prior art solutions to those problems; (4) the rapidity with which innovations are made; (5) sophistication of the technology; and (6) education level of active workers in the field.

25. I understand that Foleon, Inc. proposed that a POSITA in the field of the '816 Patent as of July 6, 2020, would have been someone knowledgeable about and familiar with familiar with electronic document assembly systems and methods pertinent to the '816 Patent. A POSITA would have had a bachelor's degree in

Computer Science or equivalent training, and approximately two years of programming experience developing client/server systems that create and render documents. Lack of work experience can be remedied by additional education, and vice versa.

26. I agree with this level of ordinary skill for the '816 Patent. Based on my experience, I understand and know of the capabilities of a person of this skill level as of the time the '816 Patent was filed. I am familiar with how a POSITA would have understood and used the terminology found in the '816 Patent at the time of its filing, and with the state of the art at that time.

27. My analysis and opinions presented herein regarding the claims of the '816 Patent have been and are based on the perspective of a POSITA as defined above as of July 6, 2020, the assumed priority date for the '816 Patent. As described above and as shown in my CV, I have extensive experience in the field of electronics in the relevant timeframe.

C. Claim Construction Standard

28. In making this Declaration, I have been asked to review the terms in the '816 Patent claims to determine what a POSITA would have understood those terms to have meant at the time of the alleged invention. In addition, I have been informed that, in an *inter partes* review proceeding, the claim terms under consideration are to be construed using the same claim construction standard used in a civil action

under 35 U.S.C. § 282(b), which I understand is articulated in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc), and its progeny. I further understand that under the *Phillips* standard, the claim terms are given their ordinary and customary meaning, as would be understood by a POSITA at the time of filing, *i.e.* July 6, 2020, consistent with the specification and the prosecution history. My opinions regarding the meaning of claim terms of the '816 Patent are based on the construction of the claim terms using the *Phillips* standard in view of the specification and the prosecution history of the '816 Patent.

D. Anticipation

29. I understand that to be valid, a patent claim must be novel, and that a claim is invalid if it is anticipated by a single prior art reference. I further understand that a reference anticipates if it discloses each and every element of the claim and enables a POSITA to make and use the claimed invention without undue experimentation.

E. Obviousness

30. I also understand that even if a claim of a patent is not anticipated, that claim is still invalid if the differences between the claimed subject matter and the prior art are such that the subject matter as a whole would have been obvious to a POSITA at the time the invention was made.

31. I understand further that the condition “at the time the invention was made” in an obviousness determination is imposed to rule out impermissible hindsight. Accordingly, permissible reasoning takes into account only knowledge which was within the level of ordinary skill in the art at the time the claimed invention was made and does not include knowledge gleaned from the applicant’s disclosure in the patent under analysis. I also understand that an opining expert is required to analyze prior art from the perspective of a POSITA and may not simply provide his or her own personal conclusions.

32. I also understand that an obviousness determination includes several factual inquiries, including: (1) determining the scope and content of the prior art; (2) ascertaining the differences between the claimed invention and the prior art; (3) resolving the level of ordinary skill in the pertinent art; and (4) taking into consideration any secondary indicia (objective evidence) of non-obviousness.

33. I also understand that such objective evidence of non-obviousness may include: (1) a long felt but unsolved need that was satisfied by the claimed invention; (2) commercial success attributable to the claimed invention; (3) unexpected results achieved by the claimed invention; (4) praise by experts of the claimed invention with factual support; (5) taking of licenses under the patent by others for reasons related to the alleged non-obviousness of the claimed invention; (6) the failure of others; (7) the teaching away of others; (8) skepticism by experts; and (9) evidence

that competitors in the marketplace have copied the invention instead of using the prior art. I also understand that, in order to establish non-obviousness, there must be a demonstrated relationship, or nexus, between any such secondary indicia and the claimed invention, *i.e.*, there must be objective evidence of non-obviousness attributable to the claimed invention.

34. I further understand that near simultaneous invention by two or more equally talented inventors working independently may or may not be an indication of obviousness when considered in light of all the circumstances.

35. I understand that a conclusion of obviousness can be based on a combination of multiple prior art references. I understand that rationales that may support a conclusion of obviousness include:

36. (A) Combining prior art elements according to known methods to yield predictable results;

37. (B) Simply substituting one known element for another to obtain predictable results;

38. (C) Using known techniques to improve similar devices (methods, or products) in the same way;

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

40. (E) Using an obvious to try solution, *i.e.*, choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;

41. (F) Applying known work in one field of endeavor to yield variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art;

42. (G) Applying some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference disclosures to arrive at the claimed invention.

43. I further understand that an analysis of obviousness should recognize that market demand, rather than scientific literature, often drives design trends.

44. I also understand that if a technique has been used to improve one device, and if a POSITA would recognize the technique would improve similar devices in the same way, then using that technique would be obvious to a POSITA unless actual application of that technique is beyond his or her skill.

45. I also understand that practical and common-sense considerations should guide an analysis of obviousness, because familiar items may have obvious uses beyond their primary purposes. I further understand that applying common sense does not require a specific hint or suggestion in a particular reference. Rather, applying common sense requires a reasoned explanation that avoids conclusory generalizations and false deductions.

46. I also understand that, when addressing a problem, a POSITA will often be able to fit together the disclosures of multiple publications. In this regard, I understand that an analysis of obviousness may consider inferences and creative steps that a POSITA would normally employ.

47. I also understand that a particular combination of prior art may be proven obvious merely by showing that it was obvious for a POSITA to try that combination. For example, when there is a design need or a market pressure to solve a problem, and/or there exists a finite number of identified, predictable solutions with a reasonable expectation of success, a POSITA has good reason to pursue the known options within his or her technical grasp. I understand that if this leads to anticipated success, it is likely that the result is not of innovation but of ordinary skill and common sense.

48. I also understand that a combination of familiar elements according to known methods is presumed to be obvious when that combination does no more than yield predictable results. Work known in one field of endeavor may prompt variations of that work for use in either the same field or a different field, based on design incentives and other market forces. If a technique has been used to improve some device, and a POSITA can implement a predictable variation of that technique, then that variation is likely unpatentable.

49. I further understand that to be proper for use in an analysis of obviousness, a reference must teach art that is analogous to the claimed invention. I also understand that under a correct analysis, any need or problem known in the field of endeavor at the time of filing and addressed by the claimed invention can provide a reason for combining the elements in the manner claimed.

50. I understand that a claim can be obvious in light of a single reference, without the need to combine references, if the elements of the claim that are not found explicitly or inherently in that single reference can be supplied by the common sense or technical knowledge of a POSITA. For example, combining two embodiments disclosed adjacent to each other in a prior art patent likely does not require a leap of inventiveness.

51. I also understand that a claimed invention may be obvious if it merely involves simple substitution of one known element for another to obtain predictable results.

52. I understand further that, in an analysis of obviousness, two items of prior art need not be like two puzzle pieces that must fit together perfectly. For example, a claimed invention may be found obvious if a POSITA would view rearrangement as an obvious matter of design choice.

53. I also understand that an analysis of obviousness requires a comparison between the properly construed claim language and the prior art on a limitation-by-limitation basis.

F. Standard of Proof

54. I understand that the standard to prove unpatentability in an inter partes review proceeding is by a preponderance of the evidence, which means more likely than not.

V. TECHNOLOGICAL BACKGROUND

A. Modular Document Generation

55. Creating documents from modular or variable content has been known for decades. For example, in 1987, U.S. Patent No. 4,712,174 to Minkler describes “[a] computer based system for generating text from a predetermined data base, in either prose or poetic form, in response to a plurality of input data provided by an operator in an interactive mode of operation with the computer.” EX1010-Minkler, Abstract. Even four decades ago, Minkler describes that “it is well known in the prior art to generate textual material by combining prewritten sentences, paragraphs or sections of text retrieved from data base files.” EX1010-Minkler, 1:12-13. In particular, Minkler’s system “provide[s] a pseudorandom computer printout of text in response to a plurality of input data items related to the intended recipient and/or sender of that text.” EX1010-Minkler, 2:17-20. In the 90’s systems for generating

documents from variable data related to a specific field were also known. For example, U.S. Patent 5,956,687 to Wamsley et al. (“Wamsley”) describes a system that “automatically generate[s] a demand letter and calculate[s] settlement amounts from information gathered in the record during pursuit of [a] claim.” EX1011-Wamsley, abstract.

56. As described in more detail below, both Hailey and Harrop, both published in the early 2000s, also describe systems for generating documents from variable or modular data or content. For example, Harrop teaches “[a] document assembly or document automation system includes an assembler for generating an instance document on the basis of a source document and one or more logic source documents referenced by the source document.” EX1006-Harrop, abstract. Similarly, Hailey teaches that “[d]ocument production systems that insert customer-specific content into a pre-fabricated document are known.” Hailey’s system, among other things, “create[es] documents by assembling document components and applying precedence and rules to create a desired document.” EX1005-Hailey, ¶2.

57. More generally, a range of products in the early 2000s generated documents from templates. For example, many word processing programs supported a “mail merge” feature which used a word processing document as a template with fields marked with a delimiter. *See, e.g.*, EX2022.

58. As recognized by Minkler, Wamsley, Hailey, Harrop, and others, the use of assembly rules to instruct or influence which content or text to include in a generated document. Minkler teaches an algorithm with a series of assembly rules that direct which words, how many syllables, which pronouns, etc. should be included in the generated poem. *See* EX1010-Minkler, 16:53-17:32. Wamsley teaches generating a demand letter with rules for which paragraphs and information to include in the letter. EX1011-Wamsley, 23:6-39. As discussed in sections X and XI below, Harrop and Hailey also describe using assembly rules to determine what content to include in a generated document.

B. *Dynamic Web Technologies*

59. The World Wide Web (“WWW”) works in part with HTML-based web pages that are displayed within Web browsers that fetch those HTML pages from web servers using the standard Hypertext Transfer Protocol (HTTP).

60. While the WWW was originally created to allow the viewing and navigation of documents, it quickly became apparent that the Web could be used for additional purposes, including for interaction with content. JavaScript was designed at Netscape in 1995 as one way to provide that functionality. *See* EX1030 And over the next few years, a competitive marketplace largely driven by Netscape and Microsoft resulted in multiple variations of JavaScript and the Document Object Model (“DOM”) encouraged this interaction. And as described above, I created my

own browser using the TCL scripting language to provide for interaction within a web page.

61. The DOM provides a way of accessing individual document elements so they can be modified (for example, enabling an image object to be dynamically changed) while the JavaScript programming language had facilities for manipulating the DOM. As a 1998 specification from the World Wide Web Consortium (W3C) explained:

This specification defines the Document Object Model Level 1, a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents. The Document Object Model provides a standard set of objects for representing HTML and XML documents, a standard model of how these objects can be combined, and a standard interface for accessing and manipulating them.

EX1031 at 2.

62. The same document provides a description of the web page facilitated by the DOM:

With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model, with a few exceptions....

EX1031 at 10.

63. Despite the W3C's effort to standardize the DOM, Netscape's "Navigator" and Microsoft's "Internet Explorer" web browsers provided competing

versions of the DOM that were similar, but not identical. This resulted in web developers having to make their web pages work with multiple versions of web browsers and DOMs, as described in a 2001 article (which also demonstrated that the DOM and JavaScript were very well used by that point). *see* EX1032

64. As web developers started to create interactive documents, the need developed to be able to dynamically fetch information from web servers into documents that were displayed within web browsers. One approach to this was developed by Microsoft in late 1998. Microsoft called it XMLHTTP and used it to improve their Outlook Web Access website. *See* EX1026. Google went on to do something similar with their creation of Gmail in 2004 (*see* EX1027) and Google Maps in 2005. *see* EX1028. Jesse Garrett coined the term “AJAX” (Asynchronous JavaScript and XML) to describe this idea, (*see* EX1029) and the term stuck.

VI. OVERVIEW OF THE '816 PATENT

65. The '816 Patent is directed to document assembly systems and methods for automatically assembling digital content or documents based on a set of modular content and parameters corresponding to the ultimate reader of the assembled document. EX1001, 1:35-41; 2:7-17. These assembled documents may take various forms including “report[s], brochures, digital magazines, case studies, product guides, sales playbooks, sales proposals, whitepapers, newsletters, company handbooks, annual reports, presentations, and various others that may be utilized in

various other fields.” Id., 4:42-47. The patent interchangeably refers to these assembled documents as “derivative” or “assembled” documents. EX1001, 6:4-6 (“In some cases, the personalized document assembly may also be referred to as derivative document which are utilized interchangeably throughout the specification.”).

66. The patent describes these generated documents as being “interactive” documents and broadly defines “interaction.” For example, the patent describes interaction when a user has “read, edited, shared, [and/or] saved” a document. EX1001, 23:13-28 (“ In some cases, the interaction data may be analyzed to extract metrics and usage analytics measuring the degree to which slides/pages were provided or made available to and viewed by customers (*e.g.*, the number of times a content item was pitched to customers), to extract performance metrics measuring the *degree to which readers have engaged with (e.g., read, edited, shared, saved)* the interactive content (*e.g., the dwell time on an item, such as the amount of time that the content was displayed in a content viewer* (*e.g.*, web browser or other application), to extract impact analytics indicative of the content's impact on business metrics (*e.g.*, average revenue per pitch of a content item that may be obtained from other systems, such as customer relationship management (CRM) systems, marketing automation systems, etc.), and various other analytics.”). In another example, interaction amounts to a user “brows[ing], filter[ing], or select[ing]

a content item (*e.g.*, page, video) for viewing.” EX1001, 25:12-14 (“the personalized document may include interactive features for a reader to actively interact with one or more content items. For instance, *a reader may be permitted to browse, filter, or select a content item (e.g., page, video) for viewing.*”). Thus, the patent broadly defines “interaction” to include even passive activities like selecting a document to read or watching a video.

67. The patent’s discussion of “interaction” is related to its described collection of “data indicative of” a user’s interaction with a document. EX1001, 23:12-13 (“For instance, data may be collected indicative of user interaction with the document.”). While the patent describes a wide spectrum of interaction, it offers no description of what the collected data looks like—whether it is a particular type of data, stored in a specific document, or otherwise has a particular form. Despite the amorphous nature of the collected data, the patent describes using this collected data “indicative of user interaction with a document” to analyze “usage analytics”, “performance metrics,” “impact analytics,” and “various other analytics” to “personalize further interactive content/document[s].” EX1001, 23:13-28.

68. Related to the discussion of use “interaction,” the patent describes collection of “data indicative of” a user’s interaction with a document. EX1001, 23:16-17 (“For instance, data may be collected indicative of user interaction with the document.”) However, despite broadly describing various categories or types of

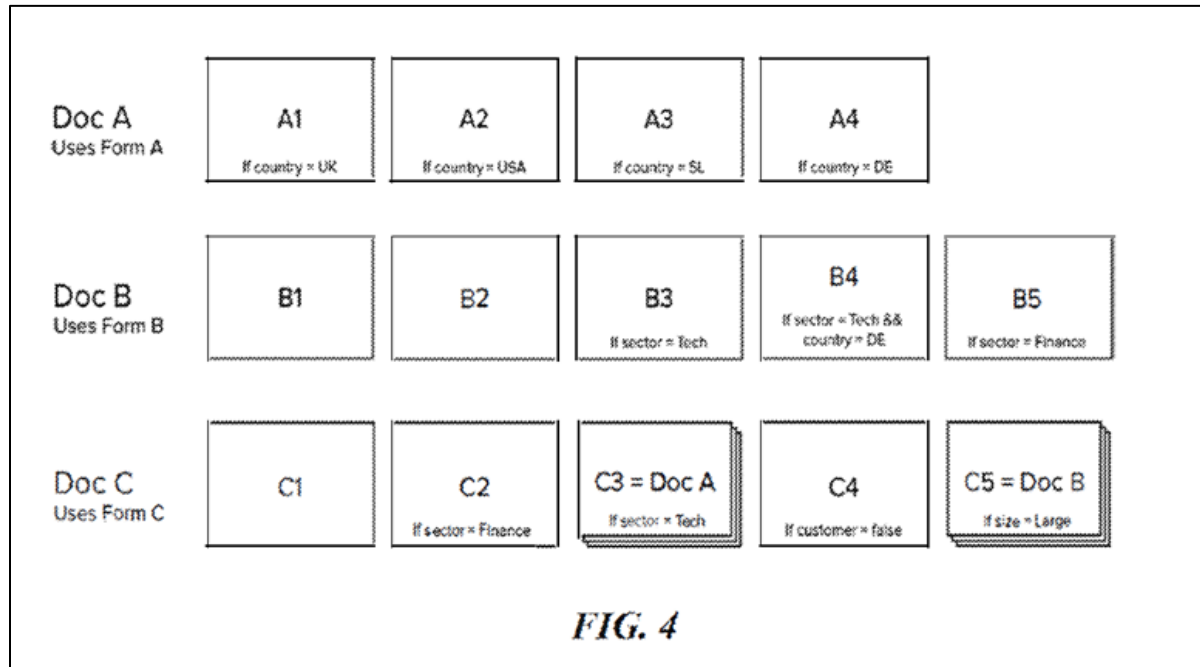
interactions that the system looks for, the patent does not describe what the collected data looks like. That is, the patent does not describe whether the collected data is a particular type of data, stored in a specific document, or otherwise has a particular form. Considering that a POSITA would be well aware of the various ways data can be stored in an electronic form, it is not surprising that the patent does not describe these details and, instead, appears to rely on a POSITA's knowledge to fill gaps in the patent's disclosure.

69. Regarding the creation of personalized interactive documents, the '816 Patent describes using a "content creation and management system 121" which "permits users (*e.g.*, creators, personalizers, analyzers, administrators, etc.) to create content and content items, generate a set of rules associated with a content item, customize a document[,], and manage various aspects of the platform." EX1001, 8:18-23. Content items—also referred to as "modular content items"—may be "a page, a slide, a video, an image, a chart, [and/or] a text file...." EX1001, 15:49-53 ("A content item such as a page, a slide, a video, an image, a chart, a text file of a document may be stored with one or more different representations (*e.g.*, mobile rendering, desktop rendering, rendering in a browser, rendering in a desktop application, etc.)).

70. The modular content items "may be automatically selected from one or more documents to be assembled into a personalized document assembly based on

pre-determined assembly rules” which “define the conditions to meet in order to include a content item into a derivative/output document.” EX1001, 5:53-58. The patent refers to these “conditions” defined in the assembly rules as “personalization parameters.” EX1001, 5:58-65, 20:3-8. For example, the personalization parameter may require a certain geographic location (e.g., state) or revenue for the content associated with the rule to be displayed in the generated document. EX1001, 16:46-49. A user may select personalization options in a personalization field provided to the user via a graphical user interface where the selections allow a document to be customized to a specific customer or reader. EX1001, 8:30-40. Thus, the personalization parameters are variables which influence which content are included in the generated document.

71. The patent describes that the assembly rules may be “embedded” in a modular content item and offers Figure 4 (reproduced below) as an illustrative depiction of “at least one document page embedded with one or more assembly rules.” EX1001, 3:49-51.



72. As seen above, page A1 includes the rule “if country = UK” such that page A1 would be included in the generated document if the user location is “UK.” EX1001, 19:3-6. Figure 4 depicts the rule “if country = UK” as appearing within the box that defines modular item A1. The accompanying description in the specification states “For instance, page A1 has embedded rule ‘if country=UK’ that when the condition is met (i.e., the personalization request indicates country=UK) page A1 will be selected or displayed.” EX1001, 19:3-6. Beyond the depiction of an “embedded” rule shown in figure 4 and other similar figures, the patent does not give an example (e.g., example code, file type, pointer, etc.) of how an assembly rule would be “embedded” in a content item. In other words, the illustration in figure 4 at best describes a concept where a rule is part of a module, but does not prescribe a certain coding or data structure requirement. Further, while the ’816 patent includes

claims which recite limitations directed to a derivative document which exclude embedded rules, the '816 patent provides no explanation, example, or discussion of a derivative document either including or excluding these assembly rules.

73. The patent describes a “content renderer for rend[er]ing the assembled document.” EX1001, 22:52-54. The function of content renderer is described at a high level including that the content renderer may “execute a script (e.g., JavaScript);” may “render the content or document as a web-page in a web browser;” and/or may “render content according to a brand theme/style mapping.” EX1001, 22:52-64. However, the patent does not provide any example of what a “content renderer” may be, other than describing the intended function.

74. Figure 15 of the patent (reproduced below) purports to depict a “user interface” that “display[s] content live when a user completes a personalization form.”

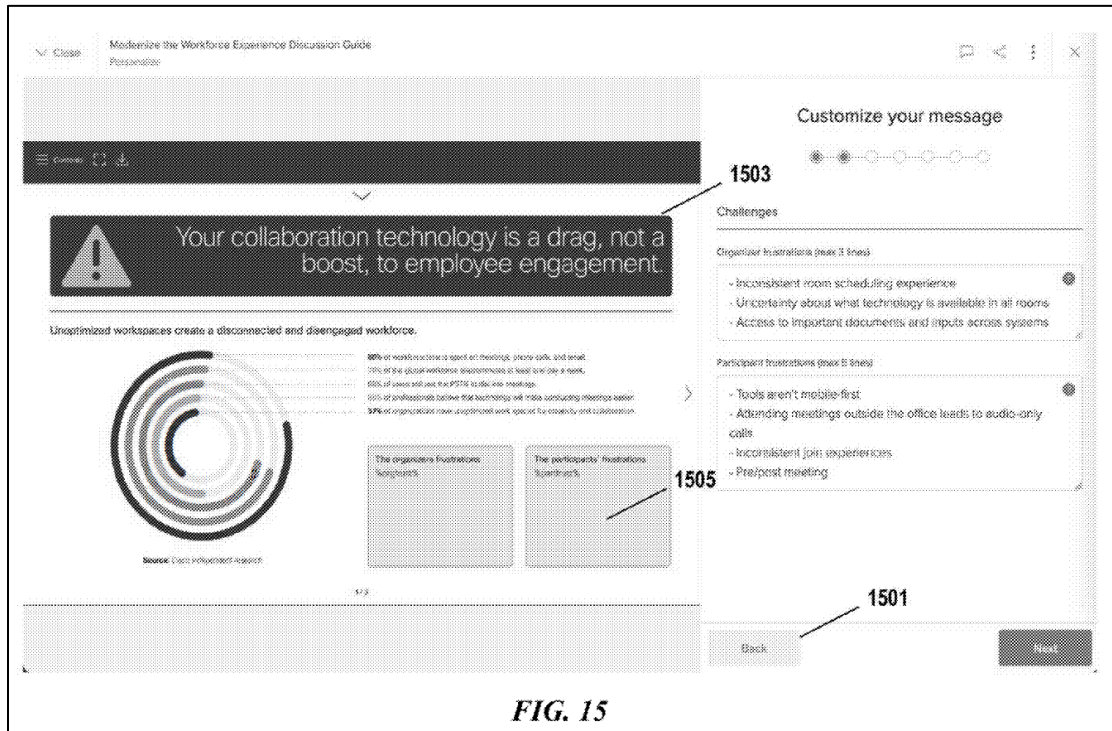


FIG. 15

EX1001, Fig. 15, 22:40-43. The patent states that “[u]pon receiving an input in a field of the personalization form, the document 1503 may automatically navigate to the relevant page and highlight the content (e.g., blocks 1505) that is related to the field.” EX1001, 22:45-48. The patent does not describe whether “navigation” requires any specific action (e.g., scrolling, zooming, panning, etc.) other than showing the user the portion of the document that is related to the field of the personalization form.

75. Another aspect of the ’816 Patent is the creation of a “link or connection between the source document and the derivative document.” EX1001, 6:9-11. When an output document is created “the personalization parameters (e.g., fields in the personalization form) that were provided during the personalization process and a

reference to the source document may be stored in the system as a link or connection to the source document(s).” EX1001, 19:35-40. When the same output document is accessed by a reader, “the source document may be loaded and the stored link may be used to dynamically apply changes from the source documents.” EX1001, 19:40-43. Aside from referring to the “stored link” and the “link or connection,” the patent offers no elaboration of what form the “link or connection” takes (*e.g.*, whether it is a data structure, a table, a dynamic reference, etc.). Further, the patent offers no description or explanation for how the personalization parameters are stored either as part of or along with the described “link or connection.”

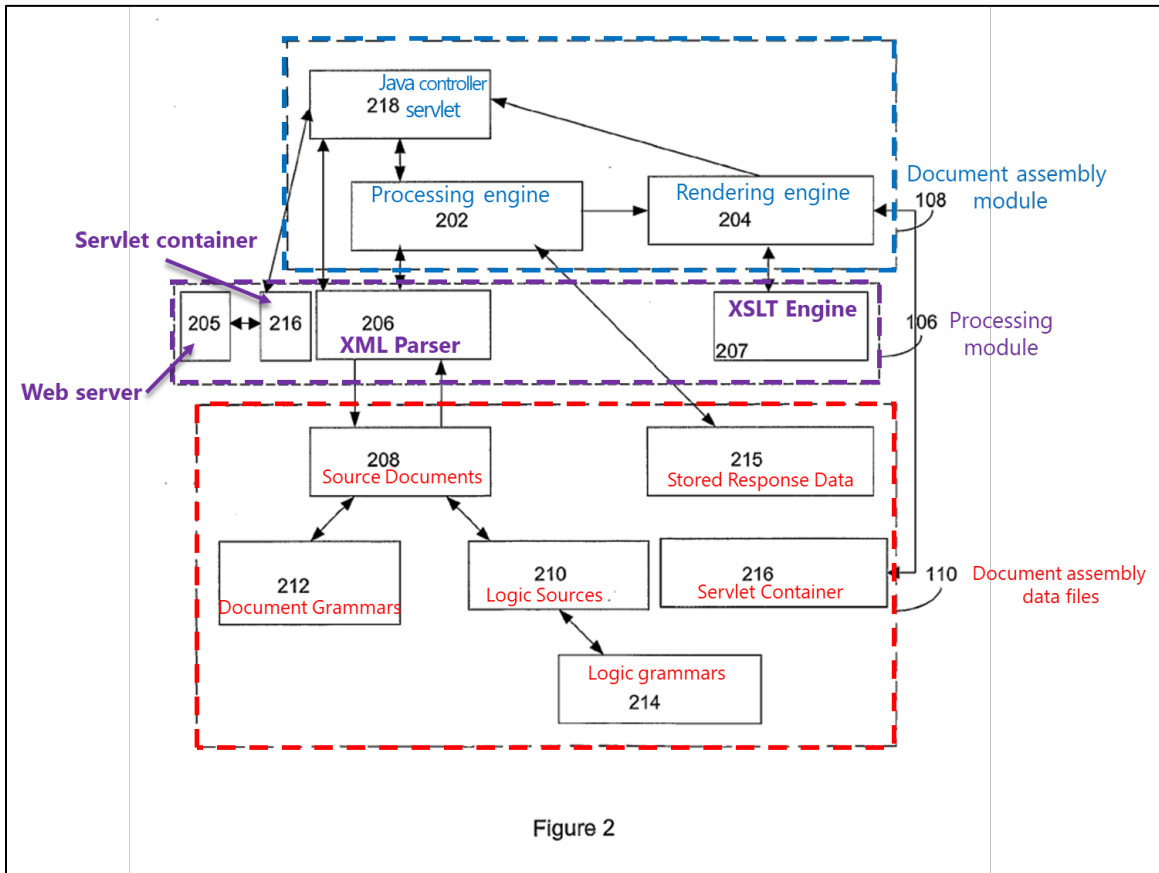
76. Lastly, the '816 patent describes that “the system may be capable of converting a personalized document into a master document” by “break[ing] the connection between the personalized document and its original master document.” The patent describes that “breaking the connection” prevents updates to the original master document from affecting the converted derivative documents. EX1001, 20:14-29. The patent describes that an aspect of converting the derivative document into a master document may involve “hardcoding” stored fields as assembly rules. EX1001, 20:25-29;

VII. SUMMARY OF THE PRIOR ART

A. *Harrop*

77. Just like the '816 Patent, Harrop teaches a computer-based document assembly system that generates different output documents (called “instance documents”) from XML source documents and personalized information provided by a user or database. EX1006-Harrop, 8:1-20.

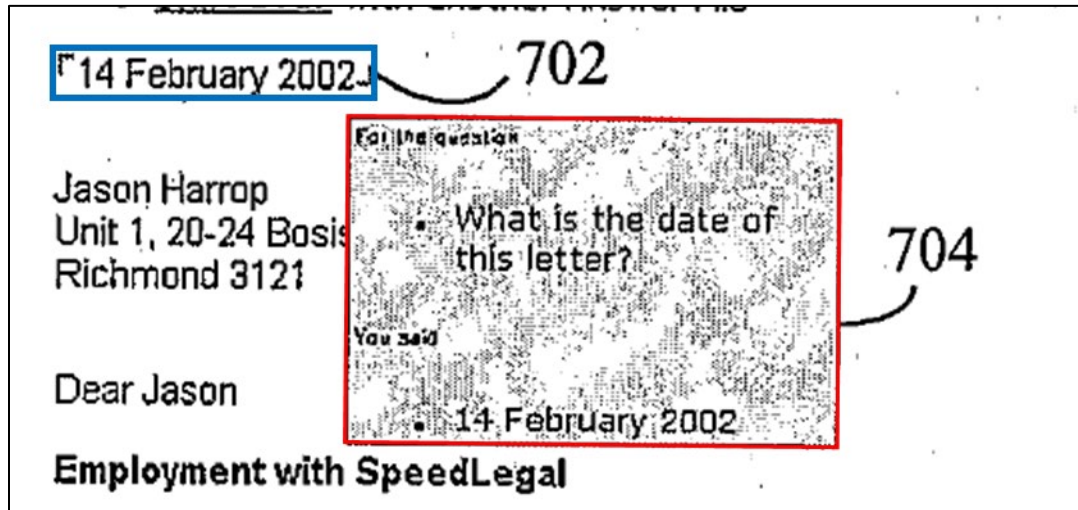
78. In one embodiment, a user may use their computer’s web browser to access the document assembly system. Over a network connection the assembly system accesses assembly modules 108, invokes a processing engine 202 to generate an instance document, and uses rendering engine 204 to send the rendered document to the user’s web browser. EX1006-Harrop, 7:8-19. Figure 2 of Harrop (reproduced below and annotated to include the names of each component) depicts many of the components of Harrop’s system. EX1006-Harrop, Fig. 2 (reproduced, annotated, below).



79. Harrop's system uses XML source documents 208 and information provided by a user or a data source to generate one or more instance documents. Ex1006-Harrop, 8:1-5. Where a user is the data source, the processing engine 202 uses logic sources 210 to generate questions and solicit necessary responses from the user; together, Harrop refers to the question and answers as "interview data" stored as "response data 215." EX1006-Harrop, 8:5-7, 9:15-19, 11:23-26, 12:8-12. The logic sources function as rules which, when evaluated by the processing engine and combined with interview data, instruct which text from the XML source document should be included in the generated instance document. EX1006-Harrop, 9:11-15. Harrop describes that logic sources 210 may be separate from the source

documents 208 or the logic sources may be incorporated in the source documents through embedded references to the logic sources. Ex1006-Harrop, 20:4-12.

80. As part of the question-and-answer interview process, Harrop's processing engine can reset questions as requested by the user and re-display the question to solicit a new answer. EX1006-Harrop, 15:10-15; 68:7-9, 68:20-22. Once a new answer is provided (and all questions are answered), a new instance document may be generated. EX1006-Harrop, 15:15-16. Specifically, Harrop describes "hypertext anchors" as part of a generated HTML instance document that, when a user moves their mouse over the hypertext anchor, a pop-up window showing the question and answer that led to the text being included. EX1006-Harrop, 15:8-10. Further, when the hypertext anchor is clicked "the processing engine 202 generates a new interview round including the relevant question, allowing the user to provide a different response to that question." EX1006-Harrop, 15:10-13. Figure 7 (reproduced-in-part below and annotated) depicts the **hypertext anchor** and the **pop-up window** showing the relevant question and answer. Fig. 7 (annotated).



Specifically, Harrop describes that the pop-up 704 is generated on mouse-over based on “JavaScript and the anchor element’s HTML onMouseOver event.” EX1006-Harrop, 68:15-19.

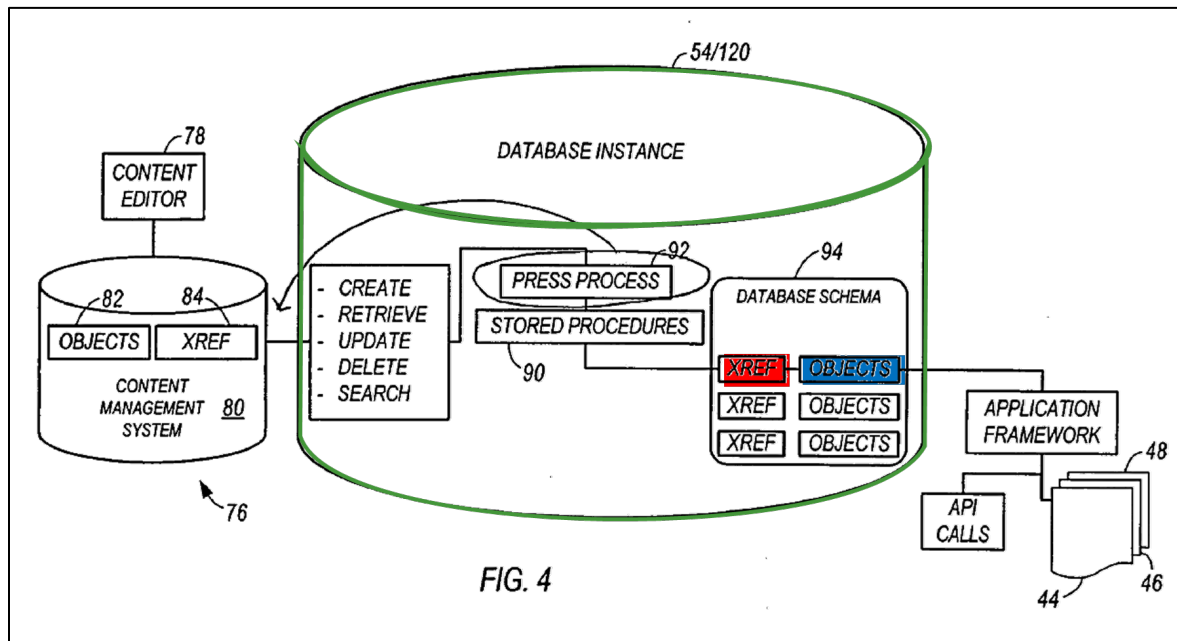
81. After the processing engine 202 has resolved all the processing-instructions logic, the rendering engine 204 renders the instance document into a desired format, such as HTML. EX1006-Harrop, 9:30-10:7. Additionally, in one embodiment, Harrop’s system can generate a document “in HTML (without controls), XML, or in rich text format (RTF).” EX1006-Harrop, 15:20-23.

B. Hailey

82. Just like the ’816 Patent, Hailey teaches a document generation system that uses input information regarding the document needed (e.g., “type of transaction, number of parties, amount of loan, price, governing law, etc.”) and generates an appropriate output file based on the input information. EX1005-Hailey,

¶7. Hailey's output file (or document) may be in one of a variety of file formats including PDF and HTML. EX1005-Hailey, ¶42.

83. Hailey's output documents are built from content and rules (cross-references called "Xref") held within a knowledge base. EX1005-Hailey, ¶¶44-45. Figure 4 illustrates one example of how the content (objects), rules (Xrefs), and knowledge base 54/120 can be arranged in Hailey's system. EX1005-Hailey, ¶64, Fig. 4 (annotated below).



84. As seen in figure 4 above, the knowledge base 54/120 is depicted as a “database instance.” *Id.*, Fig. 4, ¶95 (“embodiments of the invention may include a database or knowledge base”). In addition, Hailey describes that the xref-object pairs shown in database schema 94 are stored as associated pairs. EX1005-Hailey, ¶64.

85. In one example, Hailey describes one type of input information (location) where the appropriate consumer protection clause is included in the instance document depending on the location of the transaction (*e.g.*, New York). EX1005-Hailey, ¶5.

C. Davis

86. Davis teaches “a method for tracking the use and interaction of a user with a resource...by use of a tracking program embedded in the resource.” EX 1007-Davis, 4:3-5. Davis teaches that, even in 1997, including user tracking features in an HTML document was known. Specifically, Davis teaches at least one embodiment where “the tracking program is embedded in an HTML document (such as a Web site, a Web page, or part of a Web page—*e.g.* an ‘ad banner.’” EX1007-Davis, 5:13-16.

87. Davis teaches embedding a tracking program in an HTML document that is sent from a server to a client with the tracking program executing on the client machine. EX1007-Davis, 8:6-8. The tracking program “may monitor various indicia. such as time, mouse events, keyboard events, and the like, in order to track a user’s interaction with the Web page.” *Id.*, 8:6-13. Thus, Davis explains, that in one embodiment “the tracking program may simply monitor the amount of time the user spends interacting with the Web page, or may monitor details of choices (such as links) made by individual users within a particular Web page.” *Id.*, 8:13-20. In

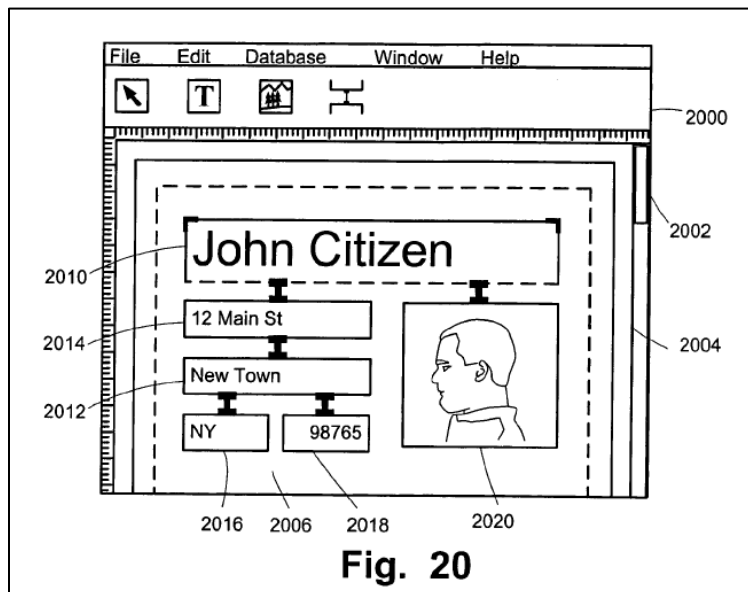
another embodiment, “[t]he tracking program may monitor operation of a peripheral input device connected to the client machine, such as a keyboard or mouse, keep a record of which choices, if any, are made by a user, and may monitor the length of time the user has displayed the Web page in addition to the time spent interacting with a particular part of it.” *Id.*, 8:30-38. Davis explains that tracking these user interactions “provide[s] meaningful data to the server concerning the user's interaction with the Web page.” EX1007-Davis, 8:64-9:2.

88. Davis teaches that “[t]he tracked information may be used to assemble resources geared toward the user's interests.” EX1007-Davis, 14:47-48. For example, “[b]ased upon the historic user profiles created in the server database, downloading of information to the same client on a subsequent visit to the same or different Web page may be done on a more intelligent basis.” EX1007-Davis, 14:48-52. The tracked information can be used to create user profiles that include “details of choices created by individual users within a particular resource.” EX1007-Davis, 4:24-32. Davis teaches using this tracked information to generate resources (*e.g.*, HTML documents) based on, for example, a historic user profile. *Id.*, 4:33-36. Thus, just like Harrop and Hailey, Davis also teaches a system that creates customized documents based on personalization parameters (*e.g.*, a user's profile).

D. Lo

89. Lo teaches a document creation and printing system that allows multiple documents to be created from a single document template, “where each document is customized for its intended readers.” EX1009-Lo, ¶¶6, 84. Lo’s customized documents are “are created by inserting variable data into a document template.” *Id.*, ¶7.

90. Further, Lo teaches using “live proofing” to “display[] variable data merged with the variable document template interactively and on-demand.” *Id.*, ¶428. As shown in figure 20 (reproduced below), Lo’s live proofing shows “what part of [a] record would look like if printed” including “extra borders 2004 and colors 2006...to make visible the constraints defined within the variable document template.” EX1009-Lo, ¶428, Fig. 20.

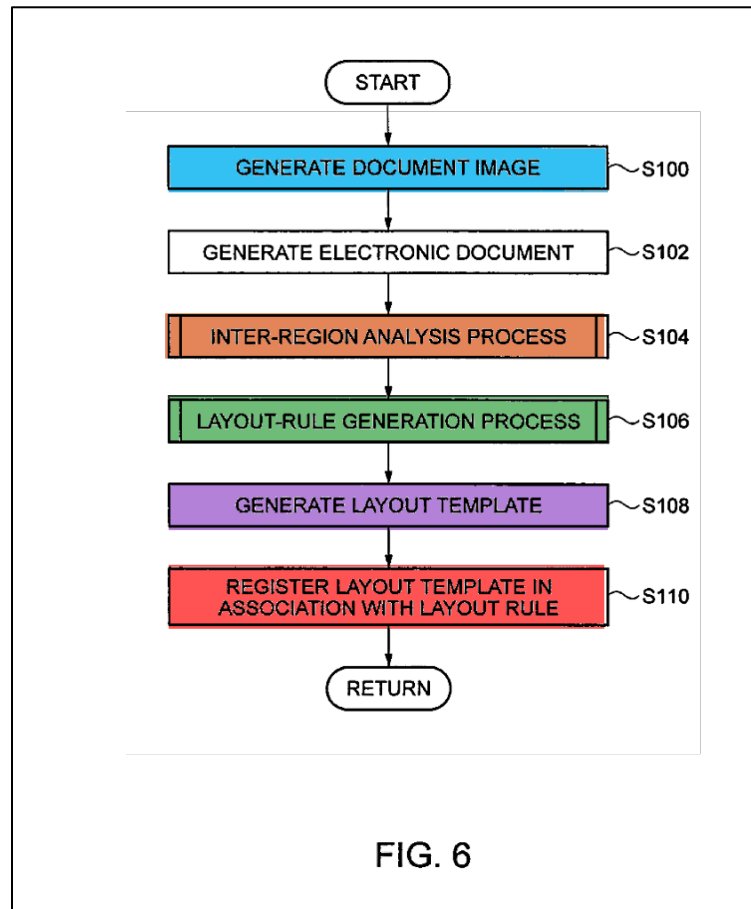


91. As seen in Fig. 20 above, Lo’s “live proofing” view shows only the portion of the document that is relevant to the edited variables in the document template. More specifically, Lo teaches that a “record” is a set of data used to produce one instance document. EX1009-Lo, ¶87. Thus, Lo teaches display what a “part” of a record (*i.e.*, data set) will look like—as opposed to displaying the entire data set. Lo explains that “[l]ive proofing is advantageous over a separate merging process because errors in the design of the template are likely to be found earlier through interactive navigation.” EX1009-Lo, ¶429.

E. Ohashi

92. Similar to the ’816 Patent, Ohashi teaches a computer-based document assembly system that uses templates to generate output documents. EX1008-Ohashi, 1:49-59. Ohashi teaches a method and system configured to automatically create layout templates based on existing documents by analyzing the content, positional relationships, font, colors, and other attributes of the existing document. EX1008-Ohashi, 16:38-46, 19:3-41. Ohashi teaches that “it takes much time and labor to produce layout templates manually one by one,” and that Ohashi’s system provides a “convenient [way] to extract a layout from an existing laid-out document to form a layout template automatically.” EX1008-Ohashi, 1:60-64. As seen in Figure 6 (annotated below), Ohashi’s system **generates an image** of an existing document (e.g., via a scanner) at **step S100**, teaches a process, **analyzes the regions**

of the document image at **step S104**, generates layout, **generates layout-rules** based on the analysis at **step S106**, **generates a layout template** at **step S108**, and finally **associates the layout rules and layout templates** with each other at **step S110**. See EX1008-Ohashi, 19:3-41.



EX1008-Ohashi, Fig. 6 (annotated).

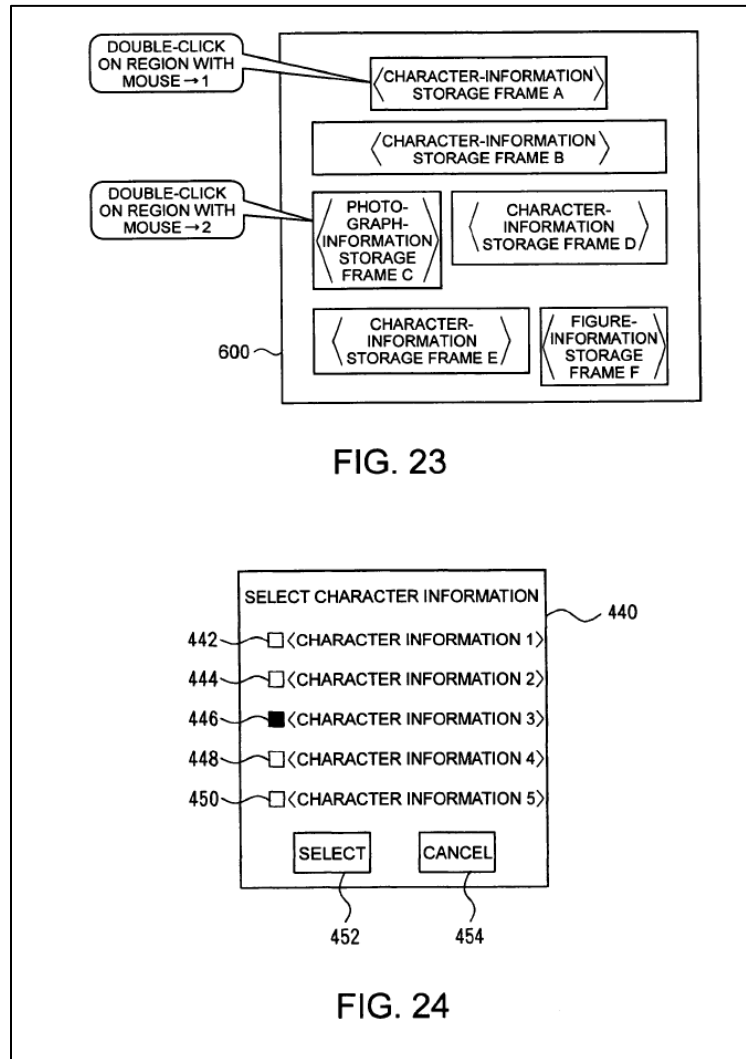
93. Thus, Ohashi takes an existing document and converts the document into a layout template that includes embedded rules (associated layout rules) through analysis of the content (images, headings, colors, etc.) and their positional relationships. Similar to the '816 patent, Ohashi teaches that in the generated layout

template various attributes of the generated template may be stored as structured data in an XML format:

the number, size, layout position, and types of information storage frames, the size, type, and color of the font of character information, character spacing and line pitch, and the quality, size, aspect ratio, and trimming of photograph information or figure information, can be set for each information storage frame, *which are described by a markup language such as an extensible markup language (XML)*.

EX1008-Ohashi, 17:43-50 (emphasis added).

94. Similar to the '816 patent, Ohashi's system allows for users to edit the generated layout by editing which content is included in a template object. For example, figures 23 and 24 (reproduced below) illustrates a user double clicking on <Character-Information Storage Frame A> and selecting the character information that should be associated with the layout object. EX1008-Ohashi, 29:38-64.



EX1008-Ohashi, Figs. 23 and 24. Figure 25 shows options for the <Photograph-Information Storage Frame C> shown in Figure 23. EX1008-Ohashi, 29:65-30:16. Thus, Ohashi teaches that the generated layout templates may be further customized by a user.

VIII. GROUNDS FOR UNPATENTABILITY

A. Ground 1: Claims 1-4, 7, 11-14, and 17 are rendered obvious by Harrop, and Hailey

95. As explained in greater detail in the following paragraphs (¶¶99-161), it is my opinion that document-assembly systems were well known to a POSITA at the time of filing of the '816 Patent, and any limitations recited in claims 1-4, 7, 11-14, and 17 of the '816 Patent would have been obvious to a POSITA in view of Harrop and Hailey.

1. Reasons to combine

96. A POSITA would have been motivated to combine Harrop and Hailey to arrive at the features of claims 1-4, 7, 11-14, and 17 for numerous reasons presented below.

a) Analogous Art

97. In my opinion, Harrop and Hailey are in the same field of endeavor as the '816 Patent, namely document assembly and creation. *See* EX1001, Abstract; 1:33-53. Hailey, for example, is directed to the creation of documents by assembling document components and applying rules to create a derivative document. EX1005-Hailey, Abstract, ¶2. Harrop is directed to document assembly and systems and methods for document assembly, or the generation of a derivative document from one or more source documents. EX1006-Harrop at Abstract, 1:4-23. Further, the '816 Patent generally describes document creation and assembly systems and

methods for use on the Internet. *See* EX1001, 4:51-5:22. Harrop and Hailey also relate to the application of document creation systems and methods for use on the Internet, including standard hyper-text markup language (“HTML”) documents. *See* EX1005-Hailey, ¶42 (output file in HTML format); EX1006-Harrop, 6:28-7:6 (document assembly system can be accessed by a user over the Internet), 15:18-28 (output document in HTML format).

b) Motivation to Combine

98. In my opinion, a POSITA would find it obvious to combine the teachings of Hailey and Harrop.

99. A POSITA would have recognized that the systems of Hailey and Harrop are closely related and would have found it obvious to combine various teachings within each reference to improve the system of the other reference. For example, a POSITA would have found it obvious to include Harrop’s teaching of linking portions of the interactive document, the master document, and the personalization parameters through hypertext anchors in a generated HTML document. In particular, a POSITA would have appreciated Harrop’s described benefit of the feature as a way for users to inspect portions of a final document quickly (*i.e.*, via a pop-up) and edit those sections by adjusting the input variables. EX1006-Harrop, 67:28-68:18. Hailey describes a similar user-input interface in its origination platform and teaches soliciting further answers from a user when more

information is needed to generate a document. EX1005-Hailey, ¶¶40-41. Accordingly, it is my opinion that a POSITA would have been motivated to include Harrop's teaching of hypertext anchors which generate a pop-up with associated questions and answers, and which reset associated answers when clicked to allow a user to make relatively quick changes to the content included in a portion of a generated document.

c) Reasonable Expectation of Success

100. In my opinion, the references include similar arrangements of well-understood components and use predictable technology in routine and conventional ways.

101. For example, a POSITA would have found it routine and predictable to implement Harrop's hypertext anchors in the system of Hailey through known technologies like URL parameters and/or JavaScript functions. Indeed, Harrop itself teaches that the pop-ups associated with the hypertext anchors are generated through "JavaScript and the anchor element's HTML onMouseOver event..." EX1006-Harrop, 68:15-19. Similarly, Hailey teaches reviewing a generated (assembled) document and, if information is missing (*e.g.*, a loan amount), querying the assembly engine in order to request the missing information from the user or database. EX1005-Hailey, ¶40. In other words, Hailey also provides a way for the system to correct a generated document by querying a user for additional information. In my

opinion, a POSITA would have recognized these well-known and commonly used web-based document languages and would have found it routine to implement the same or similar JavaScript and/or HTML coding in Hailey's generated HTML output file to accomplish the same or similar pop-up generation and reset of associated variables/questions. A POSITA would have considered the combination of these teachings nothing more than the implementation of well-known concepts in well-known systems that had been known for decades prior to the '816 Patent.

2. Claim 1

a) Limitation 1[PRE]: A method for creating an interactive document

102. In my opinion, the Hailey-Harrop combination renders the elements of the preamble obvious by teaching methods for creating interactive documents. Further, a POSITA would have understood that Hailey and Harrop each teach methods for creating interactive documents and that the Hailey-Harrop combination renders the elements of the preamble obvious.

103. For example, Hailey and Harrop each teach creating documents for display in a web browser. *See, e.g.*, EX1005-Hailey, ¶42 (“The output file (or document) may be in one of a variety of file formats as is illustrated by three exemplary file format blocks (or documents) 44, 46, and 48 (.pdf, HTML, and other)"); EX1006-Harrop, 15:2-6 (“the instance [*i.e.* generated] document is then displayed by the web browser, as shown in FIG. 7.”). A POSITA would have

understood that when a document is displayed in a web browser, a user may interact with the document in various ways including scrolling, activating links, interacting with animations and/or dynamic objects, etc. and that many of those interactions may be present in a PDF document as well. Accordingly, it is my opinion that a POSITA would have understood that Harrop and Hailey both teach creating interactive documents.

104. As discussed above, the '816 Patent describes “interaction” as including user actions such as reading, editing, saving, browsing, filtering, and/or selecting video or web page. EX1001, 6:23-39, 25:10-15; ¶¶54-64, *supra* (incorporated here). Accordingly, it is my opinion that Hailey and Harrop teach this feature at least in as much detail as the '816 Patent.

b) Limitation 1[A]: (a) providing a first document comprising a first set of modular items, wherein at least a subset of the first set of modular items have embedded rules

105. The Hailey-Harrop combination renders this limitation obvious. Each of Hailey and Harrop teach this limitation.

106. In my opinion, a POSITA would have recognized that Hailey teaches providing content (modular items) stored in a knowledge base (first document) and to create output files (interactive documents) based on various input values (personalization parameters). EX1005-Hailey, ¶39. Similarly, the '816 Patent

describes that the first (or master) document may be “stored as structured data in a document database.” *See, e.g.*, EX1001, 2:1-6; 13:36-14:24; 15:35-48.

107. In my opinion, a POSITA would understand that Hailey’s knowledge base which contains the content is the “first document” as claimed and that the various items of content are pulled from this database (i.e., the knowledge base). Hailey teaches that the items of content are stored in database 54/120 as groups of “objects” and associated embedded rules (“XREF”). EX1005-Hailey, ¶64, Fig. 4. These grouped rules (XREF) and objects form one or more modular items that are used to generate interactive documents 44, 46, 48. EX1005-Hailey, ¶64 (“The content is used ... to create documents such as the documents 44, 46, and 48.”).

108. Harrop teaches generating an instance document (interactive document) based on logic sources [RM13.1][CC13.2][RM13.3](modular items) that include reusable text options which are pulled from one or more XML source documents (first documents) based on the values of variables relevant to the circumstances (e.g., based on information specific to the parties involved and the circumstances of their relationships). EX1006-Harrop, 8:1-20, 8:22-9:23. The logic sources “are themselves structured XML documents that define questions, conditions, and reusable text objects.”; EX1006-Harrop, 8:28-29. In my opinion, POSITA would understand that these logic sources are modular items defining rules and various text options for a derivative document.

EX1003 - Declaration of Dr. Benjamin B. Bederson

109. In addition, Harrop also teaches embedding the processing instructions in the XML modules in the source documents 208 (*i.e.*, master document) with the logic sources 210. EX1006-Harrop, 20:4-20. In one example of an XML source document provided by Harrop (annotated below), the logic source is embedded in a processing instruction: `<?SpeedLegal?>`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Letter SYSTEM "file:///home/speedlegal/dtd/xhtml1-letter/dtd/xhtml1-letter.dtd">
<Letter>
  <?SpeedLegal
    <LogicSources>
      <LogicSource ID="LogicSource_1" uri="/files/logic/LS1WD.lgc"/>
      <LogicSource ID="LogicSource_2" uri="/files/logic/LS2WD.lgc"/>
      <PartyMapping>
        <PartyMap From="Contractor" To="Employee">
          <PartyDetailMap From="Shortname" To="Firstname"/>
          <PartyDetailMap From="Fullname" To="Firstname"/>
        </PartyMap>
      </PartyMapping>
    </LogicSource>
  </LogicSources>
  ?>
  <?SpeedLegal
    <MetaInformation>
      <rdf:RDF xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax-ns#">
```

```
<LetterHead>
  <Date/>
  <Salutation/>
</LetterHead>
<LetterBody>
  <p>
    <?SpeedLegal
      <Condition IDREF="PaidLeave" LogicSource="LogicSource_1"/>
    ?>
    <?SpeedLegal
      <PartyReference IDREF="Employee" LogicSource="LogicSource_1"
        Style="Pronoun" Type="PartyDetail_1"/>
    ?> shall be eligible for paid leave.</p>
  </p>
```

EX1006-Harrop, 48:15-38, 18:18-31. In my opinion, a POSITA would have understood that the above example shows embedded assembly rules (*i.e.*, the logic sources and their application) which instruct the processing engine how to assemble

the appropriate text once the appropriate questions are answered. The first extract above shows two logic sources (LogicSource_1 and LogicSource_2) being embedded. Then the second extract above shows the application of the logic from LogicSource_1. It shows that the text “shall be eligible for paid leave” will only be included if the “PaidLeave” property is true. It also includes the employee’s name in the generated text. Accordingly, a POSITA would have recognized from these extracts that Harrop teaches embedding an assembly rule (*i.e.*, logic sources and their application) in a modular item (*i.e.*, XML processing instruction module) in a first document (*i.e.*, XML source document).

110. Thus, both Hailey and Harrop teach providing a first document (database / XML source documents) comprising a first set of modular items (Xref-content groups / Logic Sources) where at least a subset of the modular items have embedded rules (XREF rules / Logic Sources). Accordingly, in my opinion, Hailey and Harrop each teach Limitation 1[A] and the Hailey-Harrop combination renders this limitation obvious. *Id.*

c) Limitation 1[B]: (b) receiving a personalization parameter for generating a derivative document;

111. In my opinion, Hailey-Harrop combination renders this limitation obvious because each of Hailey and Harrop teach this limitation. As explained above, the ’816 patent describes the personalization parameters as variables or

options which a user can select to personalize a generated document. EX1001, 8:30-35; §VII.A, *supra* (incorporated here).

112. Hailey teaches that the content (*i.e.* objects and XREF) are pulled from the database (first document) based on various data input (*i.e.* personalization parameters) into the system:

[T]he document generation module 22 performs four basic functions. The document generation module 22 ***validates incoming data*** (as is represented by a validation module 24), ***applies precedence*** (as is represented by precedence module 26), ***applies rules*** to the content (as is represented by a rules application module 28), and ***performs document assembly*** (as is represented by an assembly module 30).

EX1005-Hailey, ¶40.

113. As an example, Hailey describes processing location data (*e.g.*, Texas or California) input into origination platform 12 to select and generate an output file that includes appropriate content for the relevant location. EX1005-Hailey, ¶40-41. Hailey further describes soliciting additional input data from origination platform 12 to resolve any remaining questions and complete document assembly. EX1005-Hailey, ¶40 (“For example, if the amount of a loan has not been specified in the input data originally provided by the origination platform 12, a question module 32 queries the origination platform 12 to request needed information or data 34.”), Fig. 1. Accordingly, the origination platform receives data (*i.e.*, personalization parameters) which influence, at least in part, which content is assembled into the final output file.

114. A POSITA would have recognized that Harrop also teaches generating an instance document (interactive document) based on XML modules (modular items) that include text options which are pulled from one or more XML source documents (master documents) based on the values of variables relevant to the circumstances (*e.g.*, based on information specific to the parties involved and the circumstances of their relationships). EX1006-Harrop, ¶¶8:1-20, 8:22-9:23.

115. In one example, Harrop teaches using a user's answers to interview questions to "determine values for variables" which "determine content for the resulting instance document." EX1006-Harrop, 14:7-11. "The user's responses are used to assign values to variables, such as setting Boolean variables to a value of true or false for determining whether an associated portion of text is to be included in a generated instance document." EX1006-Harrop, 9:19-22. Thus, a POSITA would have understood that whether a portion of text is to be included in the output document is at least partly based on the answers provided by a user (*i.e.*, personalization parameters). In my opinion, the answers provided by a user are an example of receiving a "personalization parameter for generating a derivative document."

116. Harrop provides an illustrative example regarding the creation of a letter of employment for a prospective employee ("Jason"). EX1006-Harrop, 14:21-31, Fig. 6. Figure 6 illustrates one example of an interview question at reference

number 604: “What are the core hours when Jason is expected to attend the office?”; with the user (*e.g.*, HR admin) answers: “10am – 12 pm and 2pm – 4 pm.” *Id.*, Fig. 6. Once all the questions about Jason and his employment are answered, Harrop’s system “generates at step 510 an instance document in HTML format on the basis of responses provided by the user, and the instance document is then displayed by the web browser, as shown in Figure. 7.” EX1006-Harrop, 15:1-6.

117. Thus, it is my opinion that a POSITA would have recognized that Harrop uses the variables corresponding to the questions asked and answers provided by a user as personalization parameters. The answers are applied to the appropriate XML module (modular item) to resolve an associated logic and the appropriate a portion of text is pulled from the XML source document (*i.e.*, master document) and the text is included in the instance document (*i.e.*, interactive document) is at least partly based on the values (*i.e.*, answers) of those variables (*i.e.*, personalization parameters).

118. Thus, it is my opinion that a POSITA would have understood that both Hailey and Harrop teach receiving personalization parameters which are used in generating a derivative document and, therefore, Hailey and Harrop each teach Limitation 1[B] and the Hailey-Harrop combination render this limitation obvious.

d) Limitation 1[C]: (c) selecting one or more modular items from the first subset of modular items to be assembled into the derivative document wherein the one or more

modular items are selected when the personalization parameter meets the embedded rules in the respective one or more modular items; and

119. In my opinion, a POSITA would have found that the Hailey-Harrop combination renders this limitation obvious.

120. Hailey teaches selecting modular items (*i.e.* objects and XREF) from the database (first document) based on the evaluation of various data input (*i.e.* personalization parameters). EX1005-Hailey, ¶40 (“[T]he document generation module 22 performs four basic functions. The document generation module 22 validates incoming data..., applies precedence..., *applies rules to the content*..., and *performs document assembly*...”). As discussed above, Hailey describes an example of processing location data (*e.g.*, Texas or California) input into origination platform 12 to select and generate an output file that includes appropriate content for the relevant location. EX1005-Hailey, ¶40-41. Hailey further describes soliciting additional input data from origination platform 12 to resolve any remaining questions and complete document assembly. EX1005-Hailey, ¶40 (“For example, if the amount of a loan has not been specified in the input data originally provided by the origination platform 12, a question module 32 queries the origination platform 12 to request needed information or data 34.”), Fig. 1. Accordingly, the data (location, loan amount, etc.) input into the origination platform are personalization parameters which influence whether certain content is assembled into the final

output file. In my opinion, a POSITA would understand that Harrop uses the input data to evaluate whether an embedded rule is met and, depending on how the rule is met, selects which content from the modular item to include in the generated document.

121. Similarly, Harrop teaches selecting text options from logic sources in XML source documents based on received values of variables. EX1006-Harrop, 8:1-20, 8:22-9:23. In one example, Harrop teaches using a user's answers to interview questions to "determine values for variables" which "determine content for the resulting instance document." EX1006-Harrop, 14:7-11. "The user's responses are used to assign values to variables, such as setting Boolean variables to a value of true or false for determining whether an associated portion of text is to be included in a generated instance document." EX1006-Harrop, 9:19-22. A POSITA would understand that "determin[ing] content for the resulting instance document" and "determining whether an associated portion of text is to be included" are descriptions of applying the values of variable (i.e., personalization parameters) to rules in order to determine the appropriate content. Thus, whether a portion of text is to be included in the output document is based on how user's responses (*i.e.*, personalization parameters) meet a rule associated with the variables.

122. Harrop provides an illustrative example regarding the creation of a letter of employment for a prospective employee ("Jason"). EX1006-Harrop, 14:21-

31, Fig. 6. Figure 6 illustrates one example of an interview question at reference number 604: “What are the core hours when Jason is expected to attend the office?”; with the user (e.g., HR admin) answers: “10am – 12 pm and 2pm – 4 pm.” *Id.*, Fig. 6. Once all the questions about Jason and his employment are answered, Harrop’s system “generates at step 510 an instance document in HTML format on the basis of responses provided by the user, and the instance document is then displayed by the web browser, as shown in Figure 7.” EX1006-Harrop, 15:1-6.

123. Thus, Harrop uses the variables corresponding to the questions asked and answers provided by a user as personalization parameters. The answers are applied to the appropriate logic sources (modular item) to resolve an associated logic rule and the appropriate a portion of text is pulled from the XML source document (i.e., master document) and the text included in the instance document (i.e., interactive document) is based on how the values (i.e. answer) meets the logic rule.

124. Thus, it is my opinion that a POSITA would have understood that both Hailey and Harrop teach selecting one or more modular items based on how one or more personalization parameters (information related to the parties / variables holding answers as values) meets an associated rule. Accordingly, Hailey and Harrop each teach Limitation 1[C] and the Hailey–Harrop combination render this limitation obvious. *Id.*

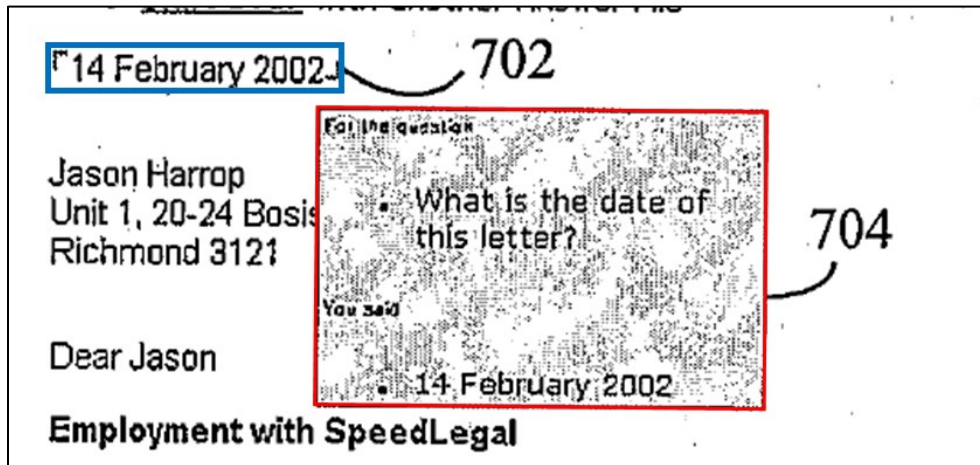
e) Limitation 1[D]: (d) creating a reference for linking the one or more modular items to the derivative document.

125. In my opinion, a POSITA would have found that the Hailey-Harrop combination teaches this limitation.

126. Limitation 1[D] recites creating a reference for linking the modular items and the derivative document. The '816 Patent describes that “the personalization parameters (e.g., fields in the personalization form) that were provided during the personalization process and a reference to the source document *may be stored in the system as a link or connection* to the source document(s)” so that changes made to source documents can be reflected when rendering derivative documents. EX1001, 19:36-40 (emphasis added). The '816 Patent does not describe any form or data structure for the “link or connection.”

127. Harrop teaches creating a reference linking the modular items included in the instance document (interactive document) and the XML source document. For example, Harrop teaches rendering an HTML instance document where certain text is rendered as “a hypertext anchor” to indicate the question and answer that resulted in that particular text being included in the rendered document. EX1006-Harrop, 67:28-68:5. Specifically, Harrop teaches that hovering (or mousing over) the hypertext anchor “displays the question and response that resulted in the anchor text being included.” EX1006-Harrop, 68:1-2. Figure 7 of Harrop depicts one example

of a **pop-up window** showing the question and answer associated with the **included hyperlinked text**.



EX1006-Harrop, Fig. 7 (reproduced-in-part, annotated).

128. Further, Harrop teaches that when a hypertext anchor is clicked on by a user, the processing engine 202 is invoked to “reset” the questions associated with the linked text. EX1006-Harrop, 68:14-18. The processing engine 202 along with java controller servlet 218, and rendering engine 204 are components of document assembly modules 108, which, in turn, exist on server 102. *See* EX1006-Harrop, 6:28-29 (“As shown in Figure 1, a document assembly system includes a server 102 having a number of modules 106 to 112”), 7:1-3 (“As shown in Figure 1, a document assembly system includes a server 102 having a number of modules 106 to 112.”), 7:8-9 (“The document assembly modules 108 include a Java controller servlet 218, a processing engine 202 and a rendering engine 204.”) Thus, invoking the processing

engine means that the hypertext anchor provides a link from the user's device (e.g., web browser) to the server.

129. The processing engine 202 uses XML source documents 208 and logic sources 210 to resolve all logic in the source document including all questions and required answers. *See* EX1006-Harrop, 8:22-9:9, 9:30-10:7, 14:13-19, 15:1-6. In addition, Harrop describes that “[i]f, at step 512, the user selects the hypertext anchor 702, the processing engine 202 generates a new interview round including the relevant question, allowing the user to provide a different response to that question.” EX1006-Harrop, 15:10-13. Once a user enters new responses and any additionally required responses are also provided, “[a] new HTML rendering of the instance XML document is then generated at step 510 and displayed by the web browser” which will contain different content depending on the user's new answers or responses. EX1006-Harrop, 15:8-16. Step 510 shown in Figure 5 refers to the newly rendered document as “generate HTML *version* of instance document.” EX1006-Harrop, Fig. 5. Further, Harrop teaches that the referred web browser operates on the user's computing system 114 which is separate from the server 102. EX1006-Harrop, 7:11-13 (“A user of the computer system 114 can access the document assembly system using a web browser application executing on the computer 114), Fig. 1 (showing the computing system 114 being separate from the server 102). Harrop also teaches that rendering engine 204, which exists on the server 102, is

responsible for generating the HTML version of the instance document. EX1006-Harrop, 10:4-5 (“Initially, the rendering engine 204 generates an HTML rendering of the instance document that includes controls for further processing of the document.”). Accordingly, a POSITA would understand that when the rendering engine updates the HTML of the instance document based on a user’s changed responses, the processing engine and the rendering engine include a link from the modular items that were changed to the instance document so that the rendering engine can update the HTML version of the instance document to reflect any changes. Indeed, the ’816 patent describes updating a derivative document in response to “an update in the input data (e.g., a field update in the personalization form...” which are displayed on user devices to elicit responsive inputs from users. EX1001, 10:25-28, 20:3-13. Thus, the ’816 patent describes updating a derivative document via a similar process to what Harrop describes.

130. A POSITA would understand that while Harrop does not specify whether the updated HTML rendering of the instance document involves updating only changed portions of the document or presenting a newly created HTML rendering, those are the two common approaches to performing such updates. Thus it would be at least obvious to use well-known technology such as AJAX (as described above in the sections on my qualifications and technology background) to update the instance document. Doing so includes a “reference for linking” to the

HTML rendering on the user's web browser since the server must be able to send the updated HTML back to the user's device that contains the instance document the hyperlink anchor that was initially clicked on resulting in invoking the processing engine. In other words, a POSITA would understand that information exists which informs the server 102 which user device and document requested the processing server to be invoked to reset certain questions and answers, and the server would use that identification information to send the updated HTML back to the user's device. At the very least, a POSITA would find it obvious to use information to link the user's active web browsing session to the server and vice versa in order to ensure that updates to the rendered document reach the proper intended instance document.

131. Thus, in my opinion, based on Harrop's teachings, a POSITA would understand that when Harrop's processing engine is invoked to reset the relevant questions, the processing engine would access the previously processed XML source document and that Harrop includes a reference for linking from the modular items in the source document to the HTML rendering of the instance document in order to update the user's web browser with any changes to the content included in the instance document. Accordingly, Harrop teaches a "creating a reference linking the one or more modular items to the derivative document linking the one or more modular items to the derivative document," as claimed.

132. As discussed above, it is my opinion that a POSITA would have found it obvious to include Harrop's link in the system of Hailey to connect generated interactive documents, the modular items included in the interactive document, and the master documents at least for the reasons discussed above. *See* §VIII.A.1, *supra*. Accordingly, limitation 1[D] is obvious in view of the Hailey-Harrop combination.

3. Claim 2: The method of claim 1, wherein the reference is created based at least in part on the personalization parameter.

133. In addition to all the limitations of claim 1, it is my opinion that the Hailey-Harrop combination further renders claim 2 obvious. *See* §VIII.A.ii, *supra* (incorporated here). As discussed regarding limitation 1[D], Harrop teaches creating a reference as a hypertext anchor which includes the question and the user's associated answer. *See* §VIII.A.ii.e, *supra* (incorporated here); EX1006-Harrop, 6:28-7:31, 14:7-11. When the hypertext anchor is hovered over the associated question and answer is displayed in a pop-up window. *See, e.g.*, EX1006-Harrop, Fig. 7.

134. It is my opinion that a POSITA would understand that Harrop's system creates the hypertext anchor such that the anchor and/or the HTML code associated with the anchor include information that allows Harrop's processing engine 202 to access the relevant portion (*i.e.*, the question) of the XML source document and the user's answer to the relevant question. Harrop's web browser uses this information

to generate the pop-up window which shows the relevant question-and-answer. The collection of information and the hypertext anchors that Harrop's system uses to perform the functions associated with the hypertext anchors creates a reference linking the modular item (e.g., the question and associated content) and the generated instance document. Thus, Harrop's system creates a reference that is based on personalization parameters (e.g., the user's answer) since the reference (hypertext anchor) includes information necessary to generate a pop-up window including the user's answer. It is thus my opinion that in combination with the disclosures of Hailey and Harrop discussed in reference to claim 1, the Hailey-Harrop combination renders claim 2 obvious as well.

4. Claim 3: The method of claim 1, wherein the reference is used to update the one or more modular items in the derivative document.

135. In addition to all the limitations of claim 1, the Hailey-Harrop combination further renders claim 3 obvious. *See* §VIII.A.2, *supra* (incorporated here). Harrop further teaches that the link (i.e., the connection between the interactive document, master document, and variables/answers) is used to apply a change to one or modular items based on a change to the master document. EX1006-Harrop, 10:5-7, 15:2-6, 15:8-16. That is, it is my opinion that a POSITA would understand that Harrop's system is designed to render the instance document based on the XML source document and that any changes to the XML source document

would result in a change to the instance document the next time the instance document is generated and rendered. EX1006-Harrop, 10:5-7, 15:2-6, 15:8-16 (resetting questions, querying new answers, rendering a new HTML instance document). For example, if a user clicks the hyperlink anchor in Harrop's HTML instance document, the question will be reset, and a user will have to provide a new answer. EX1006-Harrop, 15:8-16. If the source XML associated with the reset question has been modified since the HTML instance document was first rendered, then the re-rendered instance document—which now contains a new answer—will reflect the change made to the XML source document. Thus, Harrop teaches using the link (reference) to update one or modular items (content) in the instance document (derivative document).

136. To the extent that Patent Owner asserts that Harrop does not expressly teach that its content renderer uses Harrop's link to "update the one or more modular items in the derivative document," based on the teachings of Harrop a POSITA would have found it obvious to modify Harrop to update the modular items that are included in the derivative document based on a change to the master document Harrop teaches using reference links to incorporate XML code module into a given XML source to automatically update the resulting instance document based on changes in the XML module. EX1006-Harrop, 55:22-24 ("Alternatively, the user can import into the document a link to the module—this can point to a current

version of the module, and any modifications subsequently made to the module are automatically reflected in the resulting document.”). A POSITA would have understood that Harrop teaches the advantage of using links to source documents—in this case, an XML module—so that changes to the source can be automatically updated in the resulting document. In my opinion, a POSITA considering Harrop’s disclosure of this benefit and Harrop’s teaching of linking the master document, instance document, and personalization parameters, as discussed regarding limitation 1[C] (*see* §VIII.A.2.d), *supra*, incorporated here), would have found it obvious to use Harrop’s links to automatically update an instance document and the modular items (i.e. text) contained therein, based on changes to the source document.

137. Therefore, it is my opinion that it would have been obvious to a POSITA to include in the systems of Harrop and Hailey a rendering engine that uses a reference to update the modular items in the derivative document based on a change to the master document. This combination of teachings is applicable to both Harrop—since it contains the teachings and suggestions—as well as Hailey for the reasons discussed in §VIII.A.1 above (incorporated here). The Hailey-Harrop combination renders claim 3 obvious.

5. Claim 4: The method of claim 1, wherein the derivative document comprises a second set of modular items that do not have embedded rules.

138. As discussed above, the Hailey-Harrop combination renders obvious all the limitations of claim 1, and further renders claim 4 obvious. §VIII.A.2, *supra* (incorporated here).

139. Claim 4 further limits claim 1 such that the derivative document (the generated document) contains a second set of modular items that “do not have embedded rules.” Although the ’816 Patent describes the embedding of assembly rules in modular items in a source or master document, the ’816 Patent provides no description or example of a derivative document that includes modular items which *do not* include embedded rules. *See* EX1001, 3:49-51. Despite the lack disclosure regarding this claimed feature, a POSITA would have been able to understand the plain and ordinary meaning of this limitation based on the claim language by itself and, as discussed below, would have found the claim to be obvious over the Hailey-Harrop combination.

140. Hailey and Harrop both describe using source (master) documents and variables (personalization parameters) to select modular content and generate output (derivative) documents as HTML and PDF documents. *See* §VIII.A.2.c, *supra* (incorporated here); EX1005-Hailey, ¶¶39, 40-41, 64, Figs. 1, 4.; EX1006-Harrop, 8:1-9:23, 10:9-17, 14:7-11, 14:21-15:6, Fig. 7. The selected modular content that is

included in the generated document forms the “second set of modular items” claimed. Further, a POSITA would understand that a generated HTML or PDF document would include the selected modular content based on personalization parameters but would not include the assembly rules used to select the content and assemble the generated document. In my opinion, it would be obvious to *exclude* the assembly rules from the modular content items after they are evaluated and only populate the generated document with the object or text content from the modular items for several reasons. For example, a POSITA would recognize that including assembly rules in a generated document would unnecessarily complicate the document, including limiting the file types that the generated document use. Additionally, it is my opinion that a POSITA would understand that including the assembly rules in the generated document would introduce a security risk because an end-user could then potentially manipulate the value of parameters in the generated document relying on the rules to modify the document in ways unintended by the owner of the master document. *See, e.g.*, EX1020. Additionally, URL parameters can expose sensitive information, which is undesirable. *See, e.g.*, EX1021. Moreover, it is my opinion that a POSITA would understand that the typical approach would be to use the assembly rules to decide what content is included in the final generated document and leave the rule itself in the source/master document. In other words, a POSITA would have found it obvious to use assembly

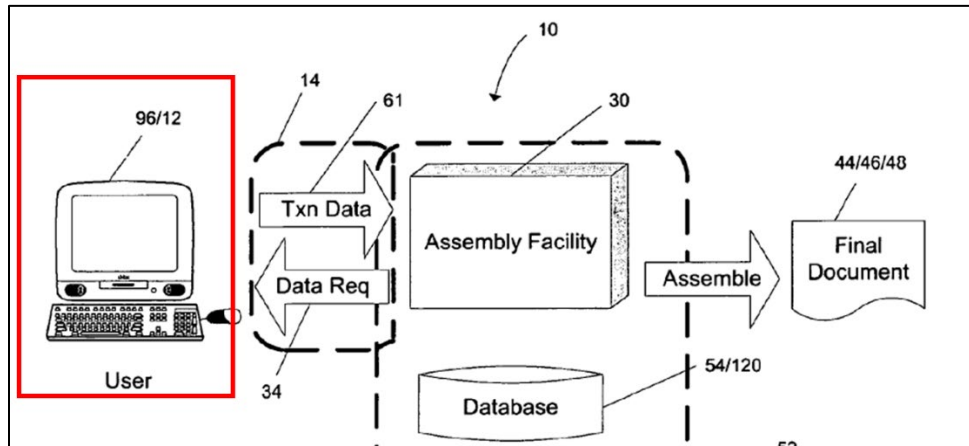
rules from modular items to select content and create an interactive document, and not import those assembly rules into the final, generated document.

6. Claim 7: The method of claim 1, wherein the personalization parameter is received via a graphical user interface or via an application programming interface.

141. In my opinion, the Hailey-Harrop combination renders obvious all the limitations of claim 1 and further renders claim 7 obvious. §VIII.A.2, *supra* (incorporated here).

142. Harrop teaches that a web browser (i.e., a GUI) displays interview questions to solicit answers from a user and generate an instance document. EX1006-Harrop, 14:21-31, Fig. 6. A web browser is or at least includes a graphical user interface (GUI).

143. In addition, Hailey teaches that an origination platform 12 is used to solicit data from a user, which data (personalization parameter) is used to select the appropriate content for the generated document. *See* §VIII.A.2.c)-d), *supra* (incorporated here); EX1005-Hailey, ¶¶39-41, 64, Fig. 1. As shown in figure 4A (annotated below), Hailey describes that a computer is used to input data into the origination platform 96/12. EX1005-Hailey, Fig. 4A, ¶66 (“A user 96 interacting with the origination platform 12 enters transaction data 61”). A POSITA would understand that the computer of Hailey includes a graphical user interface to receive data or selections by the user.



144. Accordingly, it is my opinion that the Hailey-Harrop combination renders the limitations of claim 7 obvious.

7. Claim 11

145. Claim 11 is directed to a system for creating an interactive document and recites limitations that are substantially identical to the recitations in claim 1. Indeed, Claim 11 even includes an apparent typo stating “the *method* comprising,” which may be a copy-paste error made by the patent applicant. Thus, it is my opinion that claim 11, for at least the same reasons as claim 1, is unpatentable over Hailey and Harrop. VIII.A.ii, *supra* (incorporated here).

a) **Limitation 11[PRE]: A system for creating an interactive document, the method comprising:**

146. As discussed regarding limitations 1[PRE], the Hailey-Harrop combination renders this limitation obvious. See §VIII.A.2.a), *supra* (incorporated here).

b) Limitation 11[A]: A memory for storing a set of software instructions, and one or more processors configured to execute the set of software instructions to:

147. In my opinion, the Hailey-Harrop combination renders this limitation obvious because each of Hailey and Harrop teach this limitation.

148. Hailey teaches implementing its document creation system “in software which may be stored as instructions on a computer readable medium ... and executed by a microprocessor or a similar device.” EX1005-Hailey, ¶99.

149. Harrop discloses the example of a computer server such as a Sun Fire® 15K server, and processing modules 106, 108, and 112 of the document assembly system implemented as software modules stored on hard disk storage 104 of the server 102.” EX1006-Harrop, 7:23-26.

150. In my opinion, a POSITA would recognize any of these example environments as teaching a memory for storing a set of software instructions, and one or more processors configured to execute the set of software instructions to perform one or more operations. Thus, Hailey and Harrop each teach Limitation 11[A] and, therefore, the Hailey-Harrop combination renders the limitation obvious).

- c) Limitation 11[B]: (a) provide a first document comprising a first set of modular items, wherein at least a subset of the first set of modular items have embedded rules;**

151. As discussed regarding limitation 1[A], it is my opinion that the Hailey-Harrop combination renders this limitation obvious. *See* §VIII.A.2.b), *supra* (incorporated here).

- d) Limitation 11[C]: (b) receive a personalization parameter for generating a derivative document;**

152. As discussed regarding limitations 1[B], it is my opinion that the Hailey-Harrop combination renders this limitation obvious. *See* §VIII.A.2.c), *supra* (incorporated here).

- e) Limitation 11[D]: (c) select one or more modular items from the first subset of modular items to be assembled into the derivative document, wherein the one or more modular items are selected when the personalization parameter meets the embedded rules in the respective one or more modular items; and**

153. As discussed regarding limitations 1[C], it is my opinion that the Hailey-Harrop combination renders this limitation obvious. *See* §VIII.A.2.d), *supra* (incorporated here).

- f) Limitation 11[E]: (d) create a reference for linking the one or more modular items to the derivative document.**

154. As discussed regarding limitations 1[D], it is my opinion that the Hailey-Harrop combination renders this limitation obvious. *See* §VIII.A.2.e), *supra* (incorporated here).

8. Claim 12: The system of claim 11, wherein the reference is created based at least in part on the personalization parameter.

155. As discussed regarding claim 2, it is my opinion that the Hailey-Harrop combination renders this claim obvious. *See* §VIII.A.3, *supra* (incorporated here).

9. Claim 13: The system of claim 11, wherein the reference is used to update the one or more modular items in the derivative document.

156. As discussed regarding claim 3, it is my opinion that the Hailey-Harrop combination renders this claim obvious. *See* §VIII.A.4, *supra* (incorporated here).

10. Claim 14: The system of claim 11, wherein the derivative document comprises a second set of modular items that do not have embedded rules.

157. As discussed regarding claim 4, it is my opinion that the Hailey-Harrop combination renders this claim obvious. *See* §VIII.A.5, *supra* (incorporated here).

11. Claim 17: The system of claim 11, wherein the personalization parameter is received via a graphical user interface or via an application programming interface.

158. As discussed regarding claim 7, it is my opinion that the Hailey-Harrop combination renders this limitation obvious. *See* §VIII.A.6, *supra* (incorporated here).

B. Ground 2: Claims 8-9 and 18-19 are rendered obvious by Harrop, Hailey, and Lo

159. In my opinion, the features of claims 8-9 and 18-19 are taught by Lo and rendered obvious when combined with the teachings of Hailey and Harrop.

1. Reasons to Combine

a) Analogous Art

160. Lo is analogous art to the Hailey-Harrop combination, as well as the '816 Patent. Lo is in the same field of endeavor, namely document assembly and creation. EX1009-Lo, ¶2 (“methods and processes for creation, editing, and of documents composed of text and graphics and, in printing particular, to the creation, editing and printing of variable data documents.”). Accordingly, a POSITA would have considered Lo to be analogous art to Hailey and Harrop, and the '816 Patent.

b) Motivation to combine

161. In my opinion, a POSITA would have been motivated to combine the teachings of Lo with the teachings of Hailey and Harrop. Lo teaches that “[l]ive proofing is advantageous over a separate merging process because errors in the design of the template are likely to be found earlier through interactive navigation.” EX1009-Lo, ¶381; *see* §VII.D, *supra* (incorporated here). Similarly, a POSITA would have recognized that Hailey and Harrop each appreciate the importance of previewing and/or catching errors before a final output document is generated. For example, Harrop describes document previews (*i.e.*, pop-up windows) to assist a user in adjusting the final document. EX1006-Harrop, 15:8-16, 67:28-68:18. Likewise, Hailey’s system includes a mechanism to allow a user a “change or replace

paragraphs or sections, add content, etc.” in a generated document. EX1005-Hailey, ¶69.

162. In my opinion, a POSITA would have been motivated to combine Lo’s teaching of live-proofing previews with the teachings of Hailey and Harrop to enable a user to find errors in document design at an earlier stage, as taught by Lo.

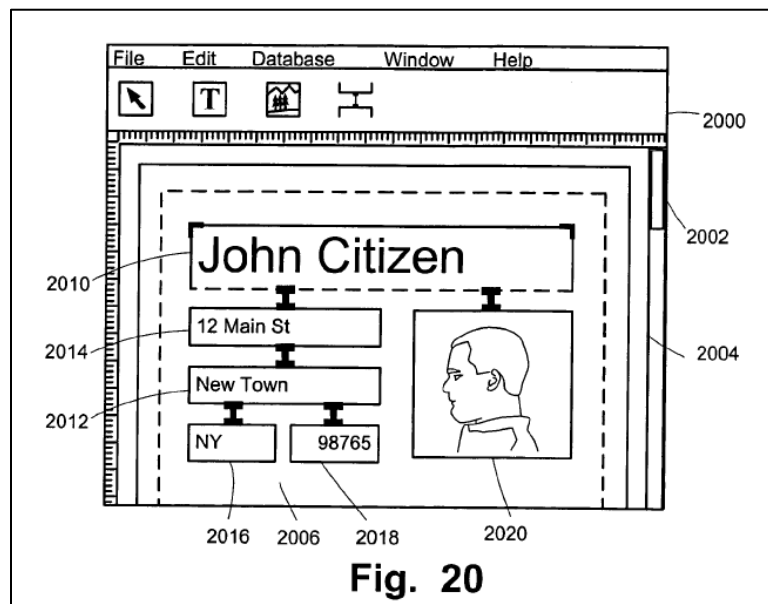
c) Reasonable Expectation of Success

163. In my opinion, a POSITA would have had a reasonable expectation of success in modifying the systems of Hailey and Harrop to include the live-proofing previews taught by Lo. The implementation of previews into a GUI would have required nothing more than incorporating known concepts (*i.e.*, previews in a GUI) into known systems (*i.e.*, the interfaces taught by Hailey and Harrop). In particular, Hailey and Harrop teach user interfaces through which a user inputs data or answers questions relative to the variables that influence which content is included in the generated documents. *See* §VIII.A.6, *supra* (incorporated here). A POSITA would have found it routine to modify the already-existing user interfaces of Hailey and Harrop to include Lo’s teachings regarding the live-proofing preview feature.

2. Claim 8: The method of claim 1, further comprising upon receiving the personalization parameter, rendering the derivative document in real-time.

164. As discussed above, the Hailey-Harrop combination renders claim 1 obvious. *See* §VIII.A.2, *supra* (incorporated here). In my opinion, claim 8 is rendered obvious over Hailey and Harrop in further view of Lo.

165. Lo teaches “live proofing” a template document to “display[] variable data merged with the variable document template interactively and on-demand.” EX1009-Lo, ¶428. Figure 20 of Lo shows “what part of [a] record would look like if printed” including “extra borders 2004 and colors 2006...to make visible the constraints defined within the variable document template.” EX1009-Lo, ¶¶380, 428-429. Fig. 20.



166. Thus, Lo teaches rendering the interactive document in real-time through displaying a preview of an interactive document (as shown in figure 20) where the preview reflects a user input regarding a personalization parameter (*e.g.*, Lo's "variable data").

167. For the reasons discussed above, it is my opinion that a POSITA would have found it obvious to include Lo's live-proofing feature in the Hailey-Harrop combined system to provide the user with a real-time preview of how the generated interactive document would change depending on changes to input variables. *See* §VIII.B.1, *supra* (incorporated here). Accordingly, the Hailey-Harrop combination as modified by Lo renders claim 8 obvious.

3. Claim 9: The method of claim 8, wherein the derivative document is rendered within a web browser by executing a script using a content renderer.

168. As discussed above, the Hailey-Harrop combination in view of Lo renders claim 8 obvious. In my opinion, the same combination renders claim 9 obvious. *See* §VIII.B.2, *supra* (incorporated here).

169. Lo teaches that his user interface and layout engine, which include the live-proofing feature, may be "incorporated as software executable on a web server that can serve up documents customized for the viewer." EX1009-Lo, ¶85. Similarly, Harrop teaches rendering the derivative document as an HTML file to display in a user interface. EX1006-Harrop, 67:28-31 ("The instance [HTML]

document can be rendered....”). Harrop further teaches that “[w]here the questions are put to a user via their web browser, the [java] controller servlet 218 can put all the questions to the user in a single HTML form...” EX1006-Harrop, 14:16-18, 14:14-15 (“The processing engine 202 provides the questions to 15 which it requires responses to the Java controller servlet 218.”). A POSITA would understand that a Java controller would execute JavaScript in order to put questions to a user “in a single HTML form.” Thus, Harrop teaches executing a script using a content renderer.

170. In my opinion, a POSITA would readily understand that displaying HTML documents on a display device includes rendering visual content from the HTML document. Accordingly, a POSITA would understand that the combined teachings of Hailey, Harrop, and Lo include using a web browser as a content renderer to display the derivative documents.

4. Claim 18: The system of claim 11, wherein the one or more processors are further configured to render rendering the derivative document in real-time upon receiving the personalization parameter.

171. For the same reasons discussed regarding claims 11 and 8 above, it is my opinion that the Hailey-Harrop combination in view of Lo renders claim 18 obvious. *See* §§VIII.A.7, VIII.B.2, *supra* (incorporated here).

5. Claim 19: The system of claim 18, wherein the derivative document is rendered within a web browser by executing a script using a content renderer.

172. For the same reasons discussed regarding claims 11 and 9 above, it is my opinion that the Hailey-Harrop combination in view of Lo renders claim 19 obvious. *See* §§VIII.A.7, VIII.B.3, *supra* (incorporated here).

C. Ground 3: Claims 10 and 20 are rendered obvious by Hailey, Harrop and Davis

1. Technical Background - User Tracking

173. Since (practically) the beginning of the web, user tracking has been known and implemented ubiquitously. In my opinion, a POSITA would have known of several commonly used ways to track how people use the web. At the simplest level, a POSITA would know of and how to use “web logs” which are logs of every URL request to a web server that are kept by essentially all web server systems. These logs can be analyzed to determine which pages are being viewed, from where, for how long, by how many different people, etc. There are numerous services that support such analysis of web logs. For example, Google Analytics has provided analysis of web logs since at least 1995. EX1017.

174. As described in this 2017 summary of a tutorial about web tracking technologies, a wide variety of tracking technologies “from simple cookies to advanced cross-device fingerprinting” have been known in the art for decades. EX1018, Abstract.

175. As discussed above, Davis, a patent filed in 1997, describes the tracking of how people use documents including monitoring “time, keyboard events, mouse events, and the like, in order to track choices and selections made by a user”. EX1007-Davis, Abstract; §VII.C *supra* (incorporated here). Davis describes performing this tracking “by use of a tracking program embedded in the resource and executable by a client.” EX1007-Davis, 4:6-8. Davis explains that “the tracking program is embedded in an HTML document (such as a Web site, a Web page, or part of a Web page—e.g. an ‘ad banner’). EX1007-Davis, 5:14-17. It also describes that “the tracking program of the present invention may be implemented in the JAVA programming language.” EX1007-Davis, 10:48-50.

2. Reasons to Combine

a) Analogous Art

176. In my opinion, a POSITA would consider Harrop, Hailey, Davis and the '816 patent to be analogous art. As discussed in ¶97 above, Harrop, Hailey, and the '816 patent are analogous art. Davis is also analogous art to Hailey, Harrop, and the '816 patent. For example, Davis teaches generating a document (HTML web page) based on personal information (*e.g.*, a user profile). EX1007-Davis, 14:46-52. Further, Davis also relates to the application of document creation systems and methods for use on the Internet, including standard hyper-text markup language (“HTML”) documents, just like the '816 patent. *See* EX1007-Davis, 4:1-10, 4:33-

36, 5:14-18 (collecting user tracking information and using it to generate a customized HTML document).

b) Motivation to combine

177. In my opinion, Davis teaches embedding tracking features in a derivative document to collect data indicative of user interaction with the document. EX1007-Davis, 5:14-18; *see* ¶¶86-88, *supra* (incorporated here). Davis teaches embedding a tracking program in an HTML document to gain information regarding user preferences, collect information regarding specific user interactions and create user profiles based on the tracked information. *See* EX1007-Davis, 4:3-8 (“a tracking program embedded in the resource and executable by a client.”), 4:19-23, (“creating a database of user profiles for use by advertisers and/or marketers to determine the effectiveness and value of network-based advertisements and/or marketing resources”), 4:33-36 (“assembling a resource, such as a Web page or a highly targeted ad banner, in accordance with a historic user profile.”). Specifically, Davis’s tracking feature monitors various user interactions including “time, mouse events, keyboard events,” (*id.*, 8:13-20) as well as “details of choices created by individual users within a particular resource.” *Id.*, 4:24-32. Davis teaches using these tracked interactions to assembly a customized resource (*e.g.*, HTML webpage) based on a user’s profile which includes the tracked information. *Id.*, 4:33-36.

178. Accordingly, it is my opinion that a POSITA would have been motivated to include in the systems of Hailey and/or Harrop a tracking feature that collects data indicative of user interactions with interactive documents, as taught by Davis, to use that collected data to improve the respective systems in various ways. For example, a POSITA would have been motivated to track user interaction with the interactive documents generated by Hailey and Harrop to glean information regarding user preferences to respective variations of user documents including preferences as to the formatting, size, color, file type, delivery method, etc. of the interactive documents. For example, a POSITA would understand that tracking a user's responses to questions posed by Hailey or Harrop's system regarding a user's name, entity, or job can be tracked using the tracking program described by Davis and that information can be reused in Hailey or Harrop's systems when assembling new interactive documents. Further, as described above in the Technological Background section (§§20-24, incorporated here), a POSITA would have been familiar with numerous reasons to track user interaction with interactive documents because designers of interactive systems improve the quality of those systems by understanding what works well and what does not for their users. Measuring how people use those systems by seeing what they read, what they click on, what they look at first, etc. inform the designer whether the system is working as intended. For example, I have done this myself numerous times when I wanted to determine which

of two possible interface designs was faster and more accurate, and I learned to do this as a best practice within the field of human-computer interaction.

c) Reasonable Expectation of Success

179. It is also my opinion that the references include similar arrangements of well-understood components and use predictable technology in routine and conventional ways.

180. In my opinion, a POSITA would have found it routine and predictable to include the user tracking feature of Davis in the systems of Harrop and Hailey. Each of Davis, Harrop, and Hailey describe generating or assembling documents in an HTML format. Further, Davis teaches that his tracking feature can be embedded as part of an HTML document. EX1007-Davis, 5:14-15 (“In one embodiment of the present invention, the tracking program is embedded in an HTML document”). In my opinion, a POSITA would have understood that implementing Davis’s user tracking program would have merely required implementing known code functions (*e.g.*, Davis’s tracking program) in a known way (*e.g.*, using known HTML or JavaScript functions) in known systems (*i.e.*, HTML documents).

3. Claim 10: The method of claim 1, further comprising embedding a tracking feature in the derivative document to collect data indicative of user interaction with the derivative document.

181. As discussed above, the Hailey-Harrop combination renders claim 1 obvious. *See* §VIII.A.2, *supra* (incorporated here). Claim 10 is obvious over Hailey and Harrop in further view of Davis.

182. Davis teaches a tracking feature that collects data indicative of user interaction with the interactive document. For example, Davis teaches embedding a tracking program in an HTML document, the tracking program monitoring and collecting information regarding how a user interacts with the HTML document. *See* EX1007-Davis, 8:13-20 (“the tracking program may simply monitor the amount of time the user spends interacting with the Web page or may monitor details of choices (such as links) made by individual users within a particular Web page.”). Davis uses the tracked information to generate personalized documents (*e.g.*, HTML documents). EX1007-Davis, 4:33-36 (“...another object of the present invention to provide means for assembling a resource, such as a Web page or a highly targeted ad banner, in accordance with a historic user profile.”).

183. As discussed above, it is my opinion that a POSITA would find it obvious to include a tracking feature to collect data indicative of user interaction—such as the one taught by Davis—in the systems of Hailey and/or Harrop.

Accordingly, the combined teachings of Hailey, Harrop, and Davis render obvious claim 10. *See* §VIII.C.2, *supra*.

4. Claim 20: The system of claim 11, wherein the derivative document is embedded with a tracking feature to collect data indicative of user interaction with the derivative document.

184. For the same reasons discussed regarding claim 10 above, it is my opinion that the Hailey-Harrop combination in view of Davis renders claim 20 obvious. *See* §§VIII.A.7, VIII.C.3, *supra* (incorporated here).

D. Ground 4: Claims 5- 6 and 15-16 are rendered obvious by Hailey, Harrop and Ohashi

1. Reasons to Combine

a) Analogous Art

185. Ohashi is analogous art to the Hailey-Harrop combination, as well as the '816 Patent. Ohashi is in the same field of endeavor, namely document assembly and creation. EX1008-Ohashi, 1:52-59 (“ Specifically speaking, a layout template that specifies a layout in which multiple information storage frames are arranged in a specified layout region is produced in advance, and at a layout job, layout elements (e.g., character information and image information) being laid out are stored in each information storage frame according to the layout template...”). Accordingly, a POSITA would have considered Ohashi to be analogous art to Hailey and Harrop, and the '816 Patent.

b) Motivation to Combine

186. In my opinion, a POSITA would have been motivated to combine the teachings of Ohashi with the teachings of Hailey and Harrop. Ohashi teaches a system and method which provides “convenient [way] to extract a layout from an existing laid-out document to form a layout template automatically.” EX1008-Ohashi, 1:60-64; *see* §VII.E, *supra* (incorporated here). Similarly, a POSITA would have recognized that Hailey and Harrop each appreciate the advantage of creating document templates from which customized documents can be made. EX1006-Harrop, 1:8-11 (“Document assembly refers to the generation of an instance document from one or more source documents. In general, a source document is a generic template document, and additional information specific to the relevant circumstances is required to generate an instance document from one or more source documents.”); EX1005-Hailey, ¶3 (“Document production systems that insert customer-specific content into a pre-fabricated document are known”).

187. Accordingly, a POSITA would have been motivated to combine Ohashi’s teaching of automatically creating layout templates based on an existing document with the teachings of Hailey and Harrop to enable a user to create templates based on a preferred document that the user possesses but lacks layout rules.

a. Reasonable Expectation of Success

188. In my opinion, a POSITA would have found it routine and predictable to include the automatic layout template creation feature of Ohashi in the systems of Harrop and Hailey. Each of Ohashi, Harrop, and Hailey describe the use of structured document templates which include content and rules for assembling the content written in XML format. *See, e.g.*, EX1005-Hailey, ¶8; EX1006-Harrop, 8:22-23; EX1008-Ohashi, 17:43-50. A POSITA would have understood that implementing Ohashi's automatic layout template system have merely required implementing known code functions (*e.g.*, Ohashi's automatic layout template system) in a known way (*e.g.*, using known XML expressions) in known systems (*i.e.*, document assembly systems of Harrop and Hailey which use XML).

2. Claim 5: The method of claim 4, further comprising converting the derivative document into a master document by embedding rules into at least one of the second set of modular items.

189. As discussed in §§VIII.A.2, VIII.A.5, *supra* (both incorporated here), the Hailey-Harrop combination renders obvious all the limitations of claims 1 and 4. In my opinion, claim 5 is rendered obvious over Hailey and Harrop in further view of Ohashi.

190. Ohashi teaches converting a rendered derivative document (*e.g.*, a document image) into a master document (*e.g.*, layout template) by embedding rules into the modular items (content) of the derivative document. Specifically, Ohashi

teaches analyzing the attributes of various content in a document image to automatically create a layout template. EX1008-Ohashi, 19:6-35; *see also* §VII.E, *supra* (incorporated here). Ohashi teaches that layout rules are created based on the content in the document image and that those layout rules are stored in a database and are associated with the layout template so that the rules and layout template are used in conjunction with each other. *See* EX1008-Ohashi, 16:64-17:8, 17:43-50, 18:49-54. A POSITA would understand that associating the layout rules and layout templates is the same as “embedding” the rules in the modular items (content) of the document image from which the layout template is made. Indeed, the ’816 patent describes that “embedding” rules may simply be associating rules and content together:

For instance, page A1 has embedded rule ‘if country=UK’ that when the condition is met (i.e., the personalization request indicates country=UK) page A1 will be selected or displayed. It should be noted that the example is for illustration purpose only. ***For example, a ruleset can be associated with a content item via a mapping relationship (e.g., specified by rule ID and content ID) or be directly stored with the content data structure.***

EX1001, 19:3-10 (emphasis added). Accordingly, Ohashi teaches converting a rendered document (document image) to a master document (layout template) by embedding rules (layout rules) in a set of modular items (content).

191. For the reasons discussed in §VIII.D.1, *supra* (incorporated here), a POSITA would have found it obvious to combine Ohashi’s teaching of

automatically creating layout templates based on an existing document with the teachings of Hailey and Harrop to enable a user to create templates based on a preferred document that the user possesses but lacks layout rules. Accordingly, the Hailey-Harrop combination as modified by Ohashi renders claim 5 obvious.

3. Claim 6: The method of claim 5, further comprising removing the reference to break the link between at least one modular item and the derivative document.

192. In my opinion, the Hailey/Harrop/Ohashi combination renders obvious all of the limitations of claim 5 and further renders claim 6 obvious. §§VIII.A.2 VIII.A.4, VIII.D.2, *supra* (each incorporated here).

193. To the extent that “the link” refers to “reference for linking” defined in limitation 1[D], as explained below, the Hailey-Harrop combination renders claim 6 obvious.

194. As discussed above, it is my opinion that Harrop teaches creating a link (hypertext anchor and associated question and answer data), aka a “reference for linking” between modular items from the xml source document and the instance document. *See* §VIII.A.2.e), *supra* (incorporated here). Harrop further teaches that after the instance document is generated (which document includes the hypertext anchors) and “[w]hen the user is satisfied with the instance document,” the user may press a button to generate an output document. EX1006-Harrop, ¶69. The output document may be generated in “Adobe® portable document format (PDF),” “HTML

(without controls), XML, or in rich text format (RTF).” *Id.* (emphasis added). In my opinion, a POSITA would understand that Harrop’s teaching of generating a pdf, HTML document “without controls”, XML, or RTF document means that the output documents will not include the hypertext anchors which link the modular items to the derivative document. Accordingly, Harrop teaches that these hypertext anchors (i.e., the claimed “link”) are removed when the output document is created. Accordingly, the Hailey-Harrop combination renders the limitations of claim 6 obvious.

4. Claim 15: The system of claim 14, wherein the one or more processors are further configured to convert the derivative document into a master document by embedding rules into at least one of the second set of modular items.

195. As discussed regarding claims 5, 11, and 14, it is my opinion that the Hailey/Harrop/Ohashi combination renders this claim obvious for the same reasons discussed regarding claim 5. *See* §§VIII.A.7, VIII.A.10, VIII.D.2, *supra* (incorporated here).

5. Claim 16: The system of claim 15, wherein the one or more processors are further configured to remove the reference to break the link between at least one modular item and the derivative document.

196. As discussed regarding claim 6, it is my opinion that the Hailey/Harrop/Ohashi combination renders this limitation obvious. *See* §VIII.D.3, *supra* (incorporated here).

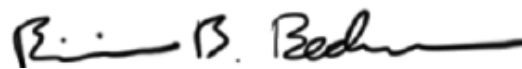
IX. CONCLUSION

197. For these reasons, it is my opinion that claims 1-20 of the '816 Patent are not novel and, within my understanding as an external technical expert, appear to be unpatentable.

198. I understand that this declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also understand that I may be subject to cross-examination concerning this declaration, and I will appear if required, at the time allotted for cross-examination.

199. I hereby declare that all of the statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code. I reserve the option to modify my conclusions in the event that authentic contradictory information should become known at a future time.

Executed on this 9th day of July, 2026.



Dr. Benjamin B. Bederson, Ph.D.