UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

FOLEON INC.,

Petitioner

v.

TURTL SURF & IMMERSE LIMITED

Patent Owner

_____

IPR2026-00404
Patent No. 11,379,651

EXHIBIT 1003

DECLARATION OF DR. BENJAMIN B. BEDERSON IN SUPPORT OF PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 11,379,651

TABLE OF CONTENTS

I. Introduction....11

II. Background and Qualifications....11

IV. Relevant Patent Law and Legal Standards....21

A. Date of Invention....22

B. Person of Ordinary Skill in the Art (POSITA)....22

C. Claim Construction Standard....23

D. Anticipation....24

E. Obviousness....24

F. Standard of Proof....30

V. Technological Background....30

VI. Overview of the '651 Patent....32

VII. Summary of the Prior Art....40

A. Harrop....40

B. Hailey....43

C. Davis....45

D. Lo....47

VIII. GROUNDS FOR UNPATENTABILITY....48

A. Ground 1: Claims 1-6, 8-10, and 13-20 are rendered obvious by Harrop, and Hailey....48

1. Reasons to combine....49

2. Claim 1....52

3. Claim 2: The method of claim 1, wherein the first modular item is stored in a structured data that defines the first set of rules. 68

4. Claim 3: The method of claim 1, wherein the first set of rules define a link between the first modular item and another document or a rule for assembling the first modular item into a derivative document. 69

5. Claim 4: The method of claim 3, wherein the value of the one or more parameters is specified by a user via a user interface. 70

6. Claim 5: The method of claim 1, wherein the user interface displays one or more personalization fields that are automatically generated based on the first set of rules. 71

7. Claim 6: The method of claim 5, wherein at least one of the one or more personalization fields is generated based on the second set of rules associated with the second modular item...73

8. Claim 8: The method of claim 6, wherein the value of the one or more parameters is specified by a user in a text file.....74

9. Claim 9: The method of claim 1, wherein the first set of rules includes a mapping relationship that maps the first modular item to a style for rendering the first modular item....75

10. Claim 10: The method of claim 1, wherein the derivative document is displayed to a reader using the mapped style. 76

11. Claim 13: The method of claim 1, further comprising storing the one or more parameters and a reference to the first document for updating the derivative document. 77

12. Claim 14: The method of claim 1, further comprising converting the derivative document into a master document by storing the one or more parameters as a set of rules associated with the derivative document. 81

13. Claim 15 83

14. Claim 16: The method of claim 15, further comprising providing a first user interface for receiving a user input to generate the first set of rules. 85

15. Claim 17: The method of claim 16, wherein the first user interface displays a plurality of fields for defining the first set of rules. 86

16. Claim 18: The method of claim 15, further comprising providing a second user interface for receiving user input to customize the one or more modular items. 87

17. Claim 19: The method of claim 15, wherein the first set of rules embedded with the at least one [modular] item of the first document includes a mapping relationship that maps the at least one item to the style. 87

18. Claim 20: The method of claim 15, further comprising providing a user interface for receiving the instruction for personalizing the first document. 87

B. Ground 2: Claims 11 and 12 are rendered obvious by Hailey and Harrop in further view of Davis 88

1. Technology Background – User Tracking....88

2. Reasons to Combine 89

3. Claim 11: The method of claim 10, further comprising collecting data about the reader's interaction with the derivative document to extract insight data. 92

4. Claim 12: The method of claim 11, further comprising receiving an instruction for personalizing the first modular item or the second modular item....93

C. Ground 3: Claim 7 is rendered obvious by Harrop and Hailey in further view of Lo....93

1. Reasons to Combine 94

2. Claim 7: The method of claim 6, further comprising: upon receiving an input in a given personalization field, displaying at least a portion of the content from the first document that is related to the given personalization field. 95

IX. CONCLUSION.....97

EXHIBITS

Exhibit No.	Description
1001	U.S. Patent No. 11,379,651 B2 to Mason
1002	Prosecution History of U.S. 11,379,651
1004	*Curriculum Vitae* 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	[RESERVED]
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	*Information on RFC 3986*, 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., *The World-Wide Web*. 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	*Foleon Inc. v. Turtl Surf & Immerse Limited*, IPR2025-01525, Paper 6 (PTAB Jan. 28, 2026).
1024	*Foleon Inc. v. Turtl Surf & Immerse Limited*, IPR2025-01525, EX2001 (PTAB Jan. 28, 2026).
1025	Prosecution History of U.S. Patent No. 11,361,144

TABLE OF CLAIMS

Claim Limitation	Claim Language
1[PRE]	A method for automatically assembling an interactive document comprising:
1[A]	embedding a first set of rules into a first modular item of a first document, wherein the first document comprises a first set of modular items;
1[B]	receiving one or more parameters for personalizing the first document to generate a derivative document;
1[C1]	selecting to display or hide at least the first modular item from the first document based at least in part on a value of the one or more parameters and the first set of rules
1[C2.A]	pull a second modular item from a second document based at least in part on the value of the one or more parameters and a second set of rules embedded with the second modular item,
1[C2.B]	wherein the second document comprises a second set of modular items, and wherein the first document and the second document are master documents that one or more of the first set of modular items and one or more of the second set of modular items are embedded with rules; and
1[D1]	generating the derivative document by assembling the second modular item with the first document or by displaying or hiding the at least first modular item from the first document,
1[D2]	wherein the derivative document comprises a plurality of modular items that are not embedded with rules.
2	The method of claim 1, wherein the first modular item is stored in a structured data that defines the first set of rules.

3	The method of claim 1, wherein the first set of rules define a link between the first modular item and another document or a rule for assembling the first modular item into a derivative document.
4	The method of claim 3, wherein the value of the one or more parameters is specified by a user via a user interface.
5	The method of claim 1, wherein the user interface displays one or more personalization fields that are automatically generated based on the first set of rules.
6	The method of claim 5, wherein at least one of the one or more personalization fields is generated based on the second set of rules associated with the second modular item.
7	The method of claim 6, further comprising: upon receiving an input in a given personalization field, displaying at least a portion of the content from the first document that is related to the given personalization field.
8	The method of claim 6, wherein the value of the one or more parameters is specified by a user in a text file.
9	The method of claim 1, wherein the first set of rules includes a mapping relationship that maps the first modular item to a style for rendering the first modular item.

10	The method of claim 1, wherein the derivative document is displayed to a reader using the mapped style.
11	The method of claim 10, further comprising collecting data about the reader's interaction with the derivative document to extract insight data.
12	The method of claim 11, further comprising receiving an instruction for personalizing the first modular item or the second modular item.
13	The method of claim 1, further comprising storing the one or more parameters and a reference to the first document for updating the derivative document.
14	The method of claim 1, further comprising converting the derivative document into a master document by storing the one or more parameters as a set of rules associated with the derivative document.
15[pre]	A method for personalizing an interactive document comprising:
15[A.1]	(a) creating one or more modular items of a first document, wherein at least one modular item is embedded with a first set of rules defining a link between the at least one modular item and a second document or
15[A.2]	a first set of rules for assembling the at least one modular item into a derivative document
15[B]	(b) receiving an instruction for personalizing the first document;
15[C.1]	(c) assembling a plurality of modular items to generate the derivative document of the first document,

15[C.2]	wherein the plurality of modular items comprise the at least one item displayed or hidden from the given document based on the first set of rules and the instruction or
15[C.3]	the plurality of modular items are pulled from the first document and the second document based at least in part on the first set of rules embedded with the at least one modular item of the first document, wherein the first document and the second document are master documents that one or more modular items of the second document are embedded with rules; and
15[D]	(iii) rendering the derivative document using a style associated with the first document and
15[E]	wherein the plurality of modular items of the derivative document are not embedded with rules.
16	The method of claim 15, further comprising providing a first user interface for receiving a user input to generate the first set of rules.
17	The method of claim 16, wherein the first user interface displays a plurality of fields for defining the first set of rules.
18	The method of claim 15, further comprising providing a second user interface for receiving user input to customize the one or more modular items.
19	The method of claim 15, wherein the first set of rules embedded with the at least one item of the first document includes a mapping relationship that maps the at least one item to the style.
20	The method of claim 15, further comprising providing a user interface for receiving the instruction for personalizing the first 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. 11,379,651 ("the '651 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.

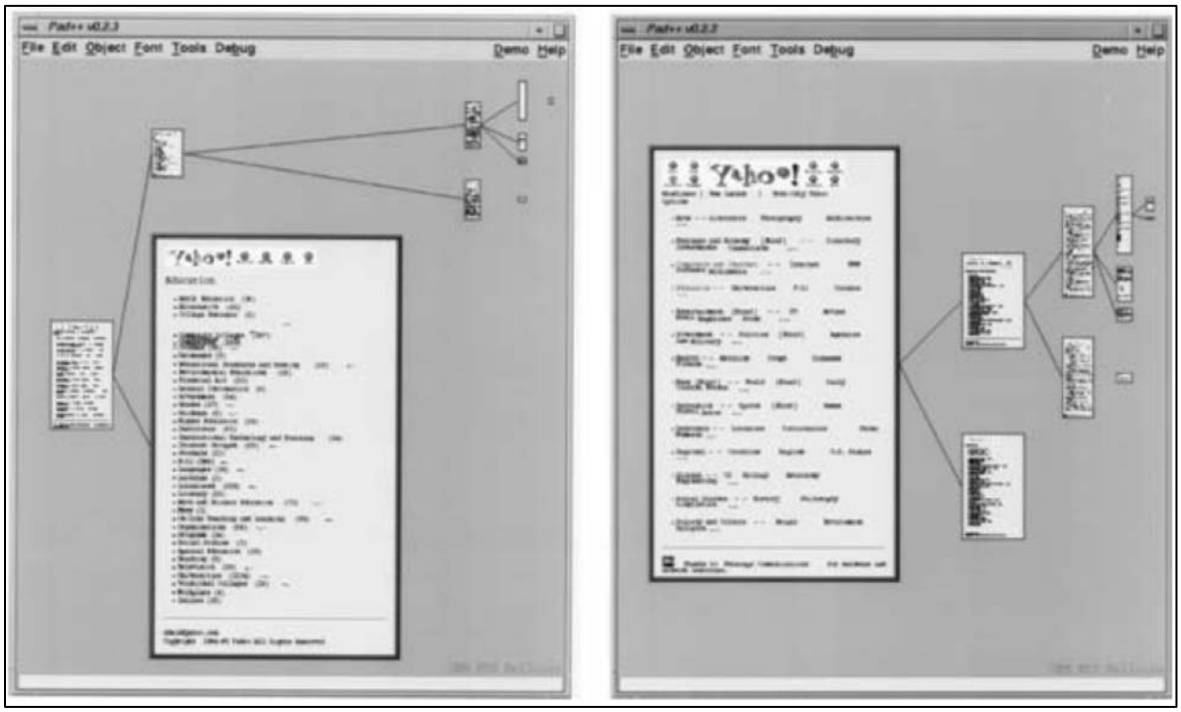


Figure 5 from "Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics"

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.

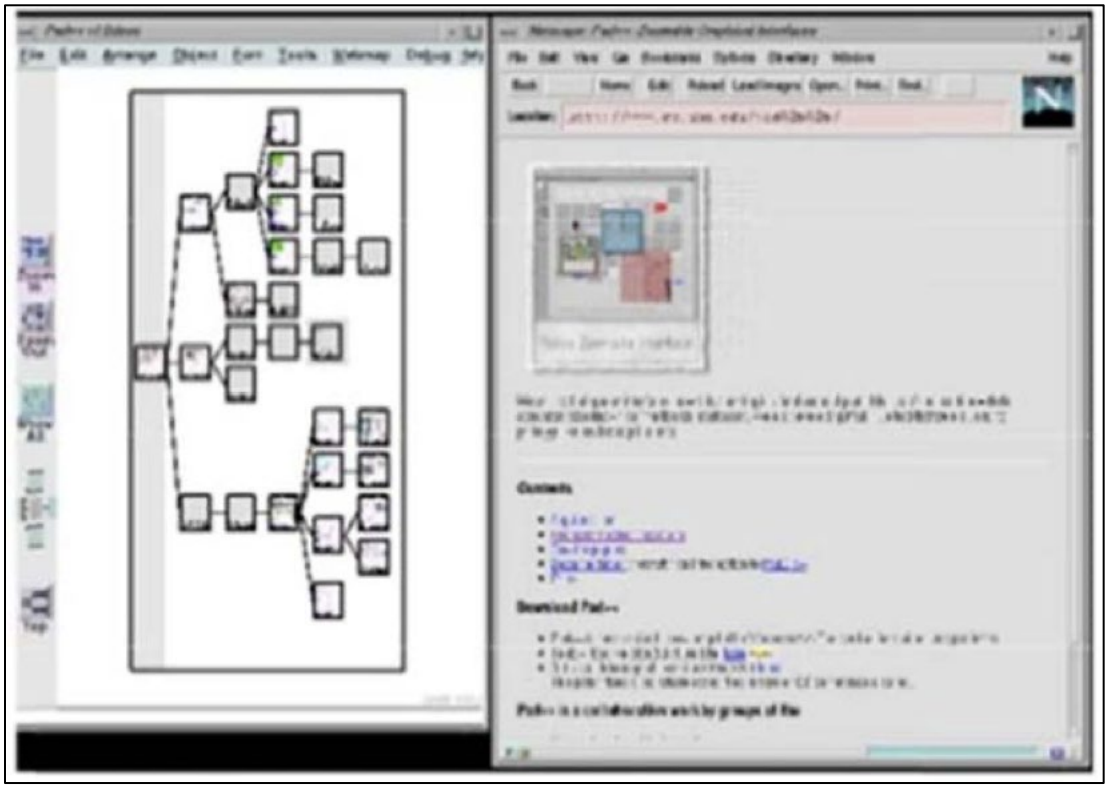


Figure 1 from PadPrints paper

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.

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

13. 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++/.

14. 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 paper[4] 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

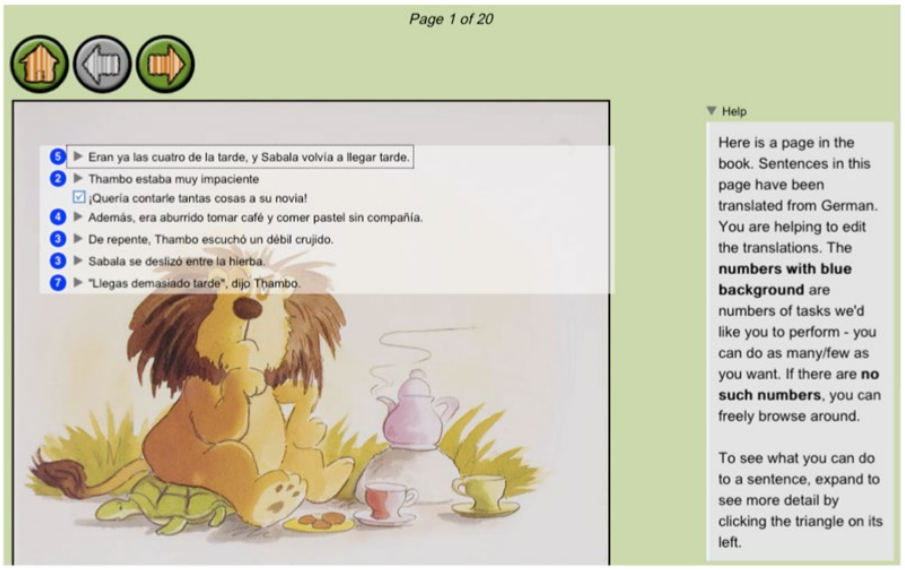


Figure 1. The MonoTrans2 UI for the target language speakers. Notice 1) the number in a blue circle indicating tasks before sentences; and 2) the contextual help on the right. The background image is a scan of the actual book (with original text removed) to provide context.

15. 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.

16. 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://ieeevis.org/year/2021/info/awards/test-of-time-awards; https://ieeevis.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.

17. 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

18. 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 '651 Patent, its prosecution history, as well as any other material referenced herein.

19. 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 '651 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

20. 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

21. 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)

22. I understand that my assessment of the claims of the '651 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 '651 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.

23. 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.

24. I understand that Foleon, Inc. proposed that a POSITA in the field of the '651 Patent as of July 6, 2020, would have been someone knowledgeable about and familiar with electronic document assembly systems and methods pertinent to the '651 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.

25. I agree with this level of ordinary skill for the ’651 Patent. Based on my experience, I understand and know of the capabilities of a person of this skill level as of the time the ’651 Patent was filed. I am familiar with how a POSITA would have understood and used the terminology found in the ’651 Patent at the time of its filing, and with the state of the art at that time.

26. My analysis and opinions presented herein regarding the claims of the ’651 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 ‘651 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

27. In making this Declaration, I have been asked to review the terms in the ’651 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 '651 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 '651 Patent.

D. Anticipation

28. 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

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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:

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

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

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

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

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

40. (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;

41. (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.

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

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

48. 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.

49. 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.

50. 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.

51. 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.

52. 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

53. 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

54. 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 "[i]t 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, 1:12-14. In particular, Minkler's system "provide[s] a pseudorandom computer printout of poetry in response to a plurality of input data items related to the intended recipient and/or sender of that poetry." 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.

55. 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." EX1005-Hailey, ¶3. Hailey's system, among other things, "creat[es] [] documents by assembling document components and applying precedence and rules to create a desired document." EX1005-Hailey, ¶2.

56. 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.

57. 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 was also well known. 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 V and VII below, Harrop and Hailey also describe using assembly rules to determine what content to include in a generated document.

VI. OVERVIEW OF THE '651 PATENT

58. The '651 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, 1:33-39; 2:5-15. 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.*, 4:40-45. The patent interchangeably refers to these assembled documents as "derivative" or "assembled" documents. EX1001, 6:2-4 ("In some cases, the personalized document assembly may also be referred to as derivative document which are utilized interchangeably throughout the specification.").

59. 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, 23:12-27 ("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), 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), 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, 25:10-14 ("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.

60. The patent's discussion of "interaction" is related to its described collection of "data indicative of" a user's interaction with a document. EX1001, 23:11-12 ("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, 23:11-26.

61. 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.

62. Even without describing the particularities of the collected data, the patent describes using the collected data to analyze "usage analytics", "performance metrics," "impact analytics," and "various other analytics" to "personalize further interactive content/document[s]." EX1001, 23:112-27.

63. Regarding the creation of personalized interactive documents, the '651 Patent describes using a "content creation and management system 121" which "permits users (*e.g.*, creators, personalizers, analyzers, administers, 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, 8:16-21. 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, 15:47-51 ("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.").

64. 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, 5:51-56. The patent refers to these "conditions" defined in the assembly rules as "personalization parameters." EX1001, 5:56-63, 20:1-6. 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, 16:44-47. 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, 8:28-38. Thus, the personalization parameters are variables which influence which content are included in the generated document.

65. 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, 3:47-49.

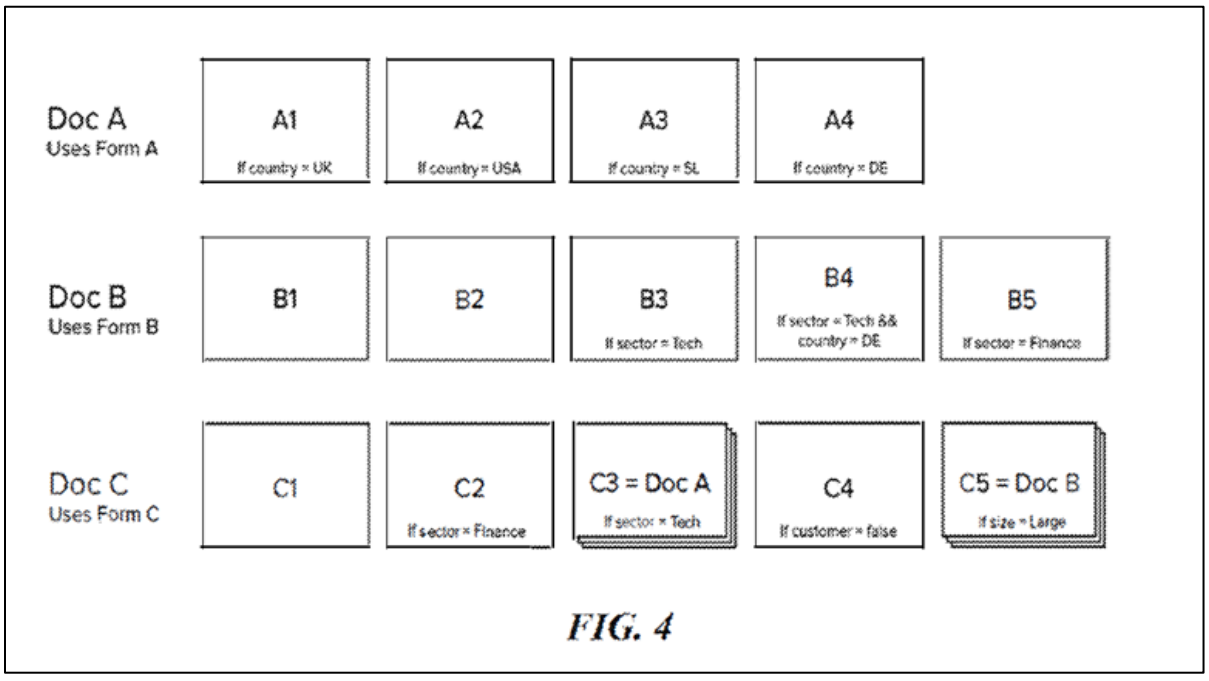


FIG. 4

66. 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, 19:1-4. 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 A**1** has embedded rule 'if country=UK' that when the condition is met (i.e., the personalization request indicates country=UK) page A**1** will be selected or displayed." EX1001, 19:1-4. 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 '651 patent includes claims which recite limitations directed to a derivative document which exclude embedded rules, the '651 patent provides no explanation, example, or discussion of a derivative document either including or excluding these assembly rules.

67. 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, 22:52-63. However, the patent does not provide any example of what a "content renderer" may be, other than describing the intended function.

68. 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."

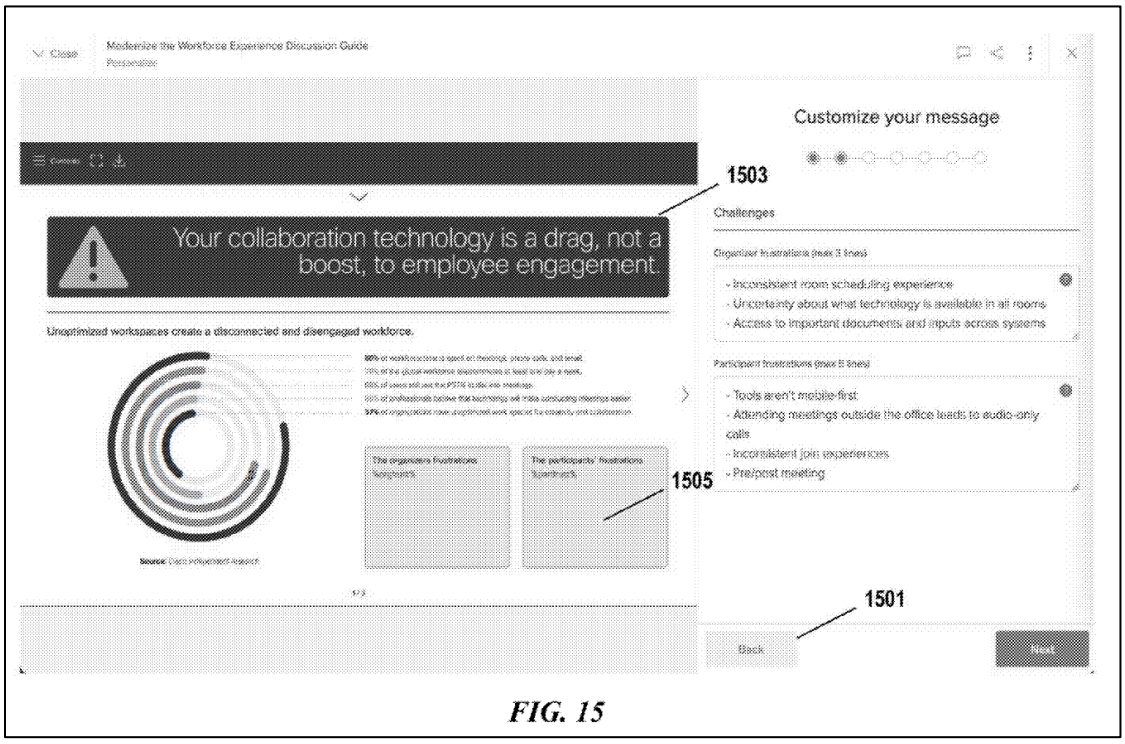


FIG. 15

69. EX1001, Fig. 15, 22:39-41. 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, 22:44-47. 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.

70. Another aspect of the ’651 Patent is the creation of a “link or connection between the source document and the derivative document.” EX1001, 6:7-9. 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, 19:33-38. 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, 19:38-41. 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.”

71. Lastly, the ’651 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, 20:12-27. The patent describes that an aspect of converting the derivative document

into a master document may involve "hardcoding" stored fields as assembly rules. EX1001, 20:23-27.

VII. SUMMARY OF THE PRIOR ART

A. Harrop

72. Just like the '651 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.

73. 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).

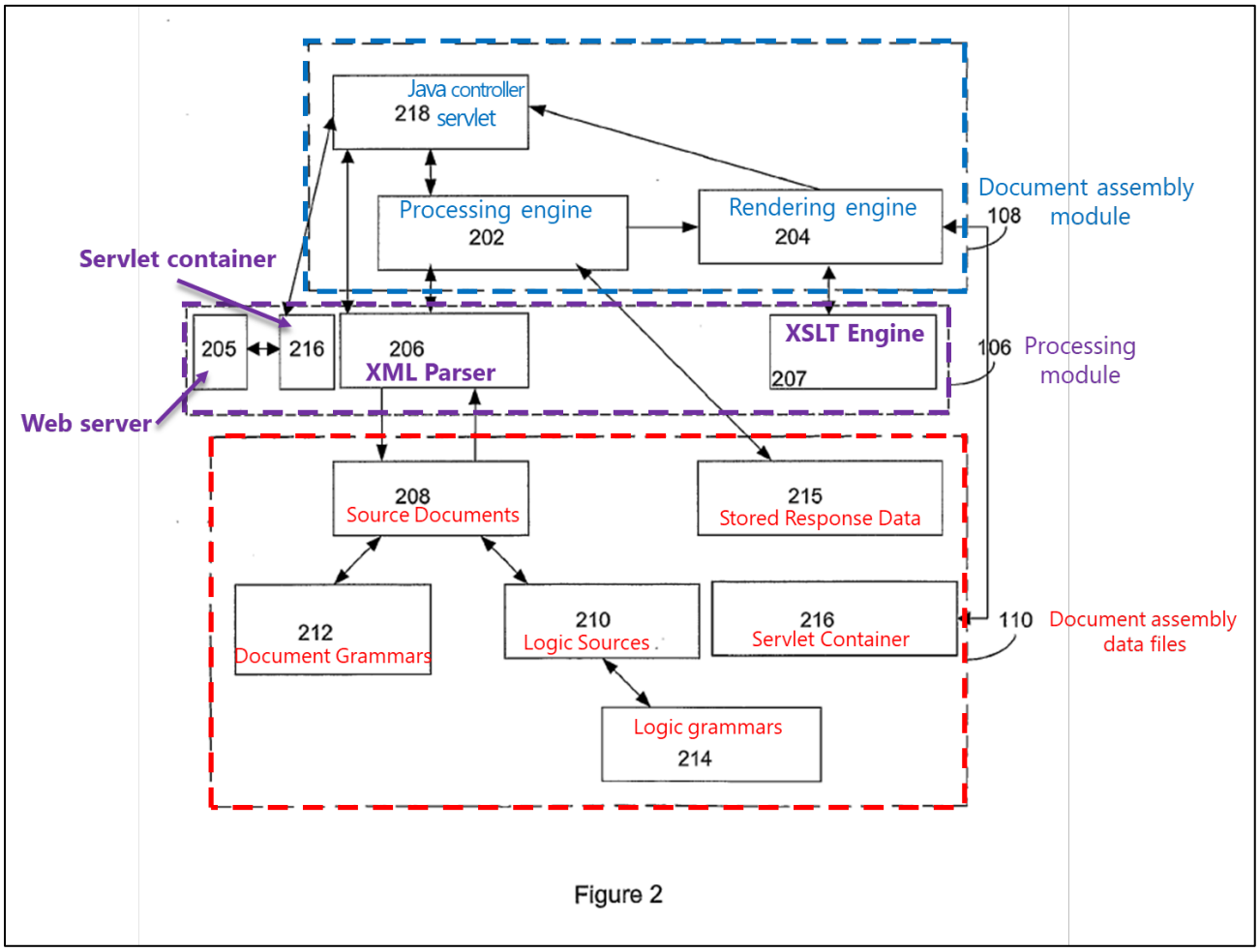
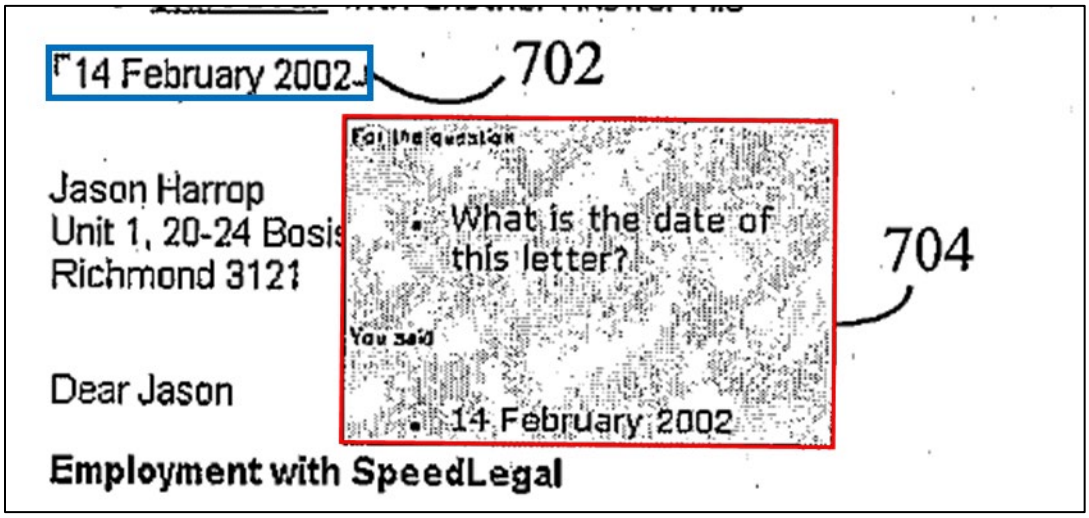


Figure 2

74. 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.

75. 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.

76. 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, 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

77. Just like the '651 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.

78. 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).

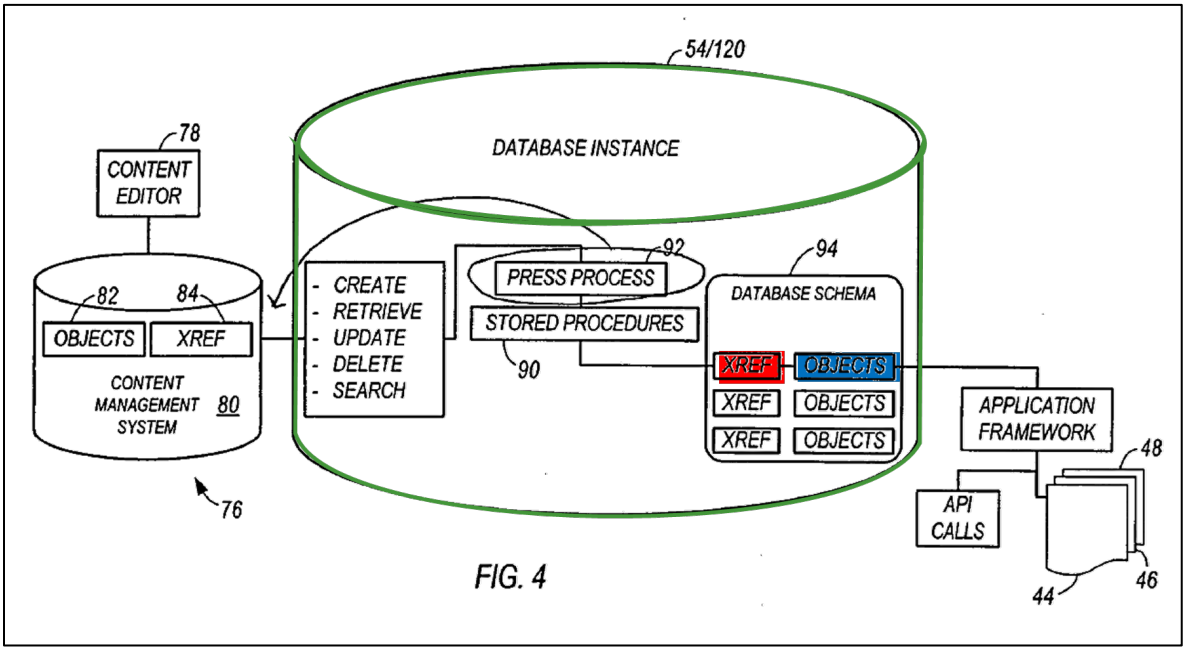


FIG. 4

79. 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.

80. 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, ¶9.

C. Davis

81. 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, 4:3-8. 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.

82. 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*., 8:12-16. 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. Davis teaches that the server comprises a memory and a processor. *See* EX1007-Davis, 7:30-37.

83. 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

84. 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.

85. 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.

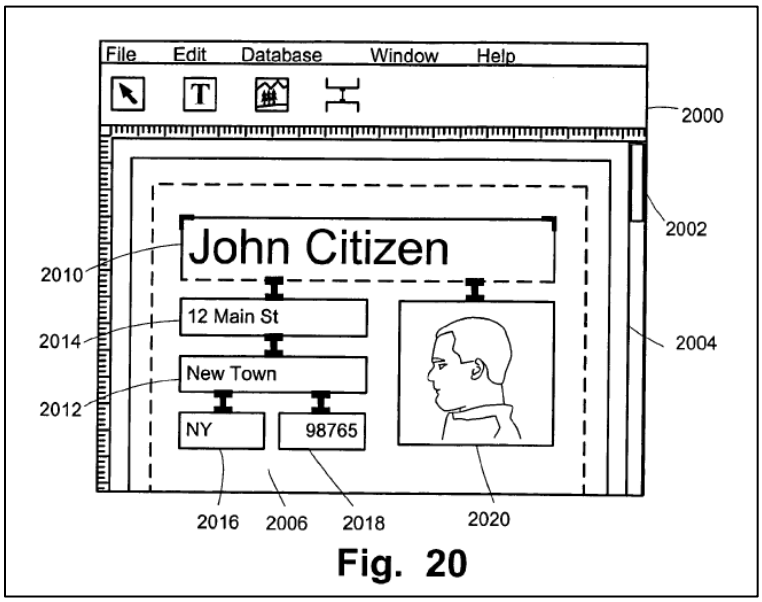


Fig. 20

86. 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.

87. 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.

VIII. GROUNDS FOR UNPATENTABILITY

A. Ground 1: Claims 1-6, 8-10, and 13-20 are rendered obvious by Harrop, and Hailey

88. As explained in greater detail in the following paragraphs (¶¶88-176), it is my opinion that document-assembly systems were well known to a POSITA at the time of filing of the '651 Patent, and any limitations recited in claims 1-6, 8-10, and 13-20 of the '651 Patent would have been obvious to a POSITA in view of Harrop and Hailey. The combined teachings of Hailey and Harrop are referred to below as the "Hailey-Harrop combination."

1. Reasons to combine

89. A POSITA would have been motivated to combine Harrop and Hailey to arrive at the features of 1-6, 8-10, and 13-20 for numerous reasons presented below.

a) Analogous Art

90. In my opinion, Harrop and Hailey are in the same field of endeavor as the '651 Patent, namely document assembly and creation. *See* EX1001, Abstract; 1:31-51. 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 at Abstract, 1:4-23. Further, the '651 Patent generally describes document creation and assembly systems and methods for use on the Internet. *See* EX1001, 4:40-58. 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

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

92. 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

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

94. 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 ’651 Patent.

2. Claim 1

a) Limitation 1[PRE]: A method for automatically assembling an interactive document comprising:

95. The Hailey-Harrop combination renders the elements of the preamble obvious by teaching methods for automatically assembling an interactive document.

96. For example, Hailey and Harrop each teach methods for creating interactive 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 document is then displayed by the web browser, as shown in FIG. 7.”). A POSITA would understand 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 the same interactions may be present in a PDF document as well. Accordingly, a POSITA would understand that Harrop and Hailey both teach methods of creating interactive documents. *Id.*

97. As discussed above, the ’651 Patent describes “interaction” as including user actions such as reading, editing, savings, browsing, filtering, and/or

selecting video or web page. EX1001, 6:21-36, 25:10-15. Accordingly, Harrop and Hailey teach this feature at least in as much detail as the ’651 Patent.

98. Further, the ’651 patent’s document assembly system explicitly contemplates that “the value of the one or more parameters is specified by a user via a user interface” during the document assembly process. EX1001, 2:5-6. Thus, a POSITA would understand that the ’651 patent does not use the term “automatically” to exclude user action or intervention such as providing answers, responses, or selections. Accordingly, a POSITA would have understood Hailey and Harrop as teaching “automatically assembling” interactive documents as claimed and as described by the ’651 patent.

99. Thus, Hailey and Harrop each teach methods of creating interactive documents and the Hailey-Harrop combination renders limitation 1[PRE] obvious.

b) Limitation 1[A]: embedding a first set of rules into a first modular item of a first document, wherein the first document comprises a first set of modular items:

100. The Hailey-Harrop combination renders this limitation obvious because each of Hailey and Harrop teaches this limitation.

101. 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. 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. Similarly, the '651 Patent describes that the first (or master) document may be "stored as structured data … in the document database." *See, e.g.*, EX1001, 1:66-67; 13:34-14:22; 15:33-46; 24:33-35. 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). 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.").

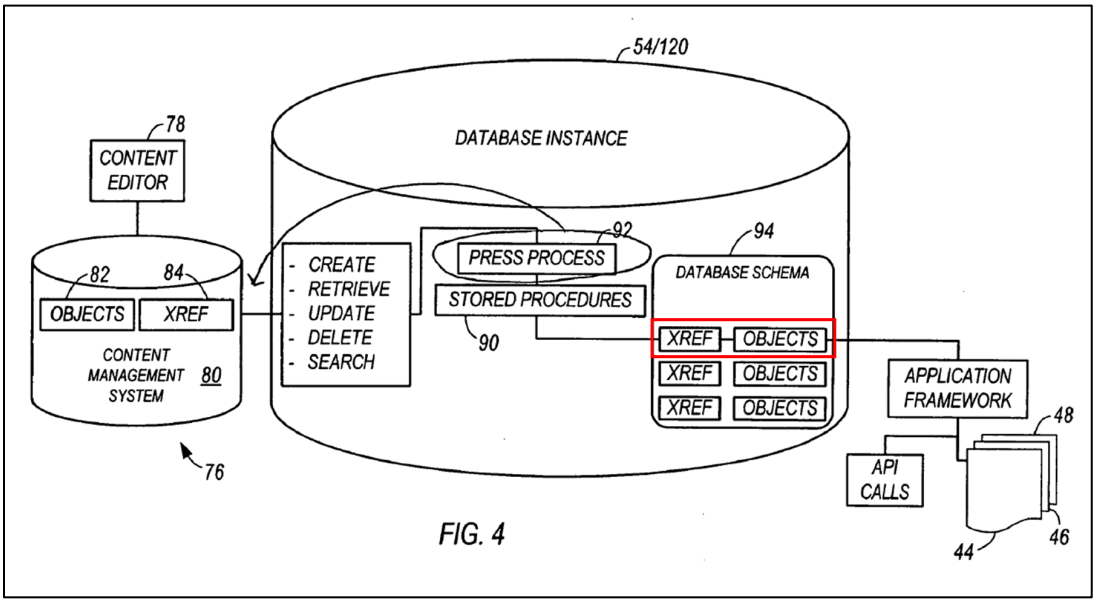


FIG. 4

102. Similarly, Harrop teaches generating an instance document (interactive document) based on logic sources (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. A POSITA would understand that these logic sources are modular items defining rules and various text options for a derivative document.

103. 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, 20:4-12. 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/xhtml-letter/dtd/xhtr
v2.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 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, Harrop teaches embedding an assembly rule (i.e., logic source) in a modular item (i.e., XML processing instruction module) in a master document (i.e., XML source document).

104. Thus, both Hailey and Harrop teach embedding a first set of rules (XREF rules / Logic Sources) into a first modular item (Xref-content groups / Logic Sources) of a first document (database / XML source documents), wherein the first document comprises a first set of modular items a first set of first document.

Accordingly, Hailey and Harrop each teach Limitation 1[A] and the Hailey-Harrop combination renders this limitation obvious.

c) Limitation 1[B]: receiving one or more parameters for personalizing the first document to generate a derivative document;

105. The Hailey-Harrop combination renders this limitation obvious because each of Hailey and Harrop teach this limitation. As explained in §VI, *supra* (incorporated here), the '651 patent describes parameters for personalization as variables or options which a user can select to personalize a generated document. EX1001, 8:28-33. A POSITA would understand that this description aligns with the plain and ordinary meaning of "parameters for personalization."

106. Hailey teaches that the content (*i.e.* objects and rules) are pulled from the database (first document) based on various data input (i.e. personalization parameters) input 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 (emphasis added). 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.

107. Similarly, Harrop teaches receiving values of variables (e.g., based on information specific to the parties involved and the circumstances of their relationships) to generate an instance document. EX1006-Harrop, 8:1-20, 8:22-9:23.

108. In one example, Harrop teaches receiving 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. The answers provided by a user are an example of receiving a "personalization parameter for generating a derivative document."

109. Thus, it is my opinion 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. *Id.*

d) Limitation 1[C1]: selecting to display or hide at least the first modular item from the first document based at least in part on a value of the one or more parameters and the first set of rules or

110. The Hailey-Harrop combination renders this limitation obvious because each of Hailey and Harrop teach this limitation.

111. 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) and displaying the selected modular item in the generated documents: 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***." (emphasis added)). As discussed above, 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 data (e.g., location, loan, amount) input into the origination platform are personalization parameters which influence whether certain content is assembled into the final output file. 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.

112. 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 variables (i.e., personalization parameters) to rules to

determine which content is to be displayed (included) or hidden (excluded). 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.

113. 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?"; where 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.

114. 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 and the appropriate portion of text is selected and displayed in the generated instance document (i.e., interactive document) based on how the values (i.e. answer) meet the logic rule.

115. Thus, it is my opinion 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. Thus, Hailey and Harrop each teach Limitation 1[C1] and the Hailey–Harrop combination render this limitation obvious. *Id*.

e) Limitation 1[C2.A] pull a second modular item from a second document based at least in part on the value of the one or more parameters and a second set of rules embedded with the second modular item,

116. The Hailey-Harrop combination renders this limitation obvious because each of Harrop teaches this limitation.

117. Harrop teaches using links to incorporate a module from a second source into a first XML document:

> 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.

EX1006-Harrop, 55:22-24. For each module link, Harrop's editor 112 includes, *inter alia*, the "logic sources required by the module to the target document LogicSources" and "party mapping (if possible) so that the module can be imported into the target document." EX1006-Harrop, 56:2-3. Thus, Harrop teaches that a second set of rules (the logic sources and party mapping) are embedded in the second modular item (the linked module). A POSITA would understand Harrop as teaching

that the linked modules are evaluated and included in the generated document according to a similar process as modules that are directly included in a source document—that is, input data is evaluated and a module's content is included if the proper conditions are met. *See* EX1006-Harrop, 8:1-20, 8:22-9:23. Accordingly, Harrop teaches pulling a second modular item (content in a referenced module) from a second document (the database or source document where the referenced module resides) depending on the value of a personalization parameter.

f) Limitation 1[C2.B] wherein the second document comprises a second set of modular items, and wherein the first document and the second document are master documents that one or more of the first set of modular items and one or more of the second set of modular items are embedded with rules; and

118. The Hailey-Harrop combination renders this limitation obvious because each of Harrop teaches this limitation.

119. Harrop teaches that the modular items reside in target documents directly when a user imports the text and all associated logic of the module into a target documents or modules reside elsewhere in data structures and are incorporated into a target document via reference. EX1006-Harrop, 55:6-20, 22-56:13; *see* §VIII.A.2.d, supra (incorporated here). As discussed above, Harrop teaches source XML documents 208 which contain logic sources (i.e., modules) having "questions, conditions, and reusable text objects." EX1006-Harrop, 8:22-30; *see* §VIII.A.2.c),

supra (incorporated here). Similarly, the '651 patent describes that the master document may be stored as "structured data" in a document database. *See, e.g.*, EX1001, 1:66-67; 13:34-14:22; 15:33-46; 24:33-35. A POSITA would understand that in Harrop's system, the modules, which include reusable text and logic sources, are contained in XML source documents.

120. Further, Harrop teaches that the modules comprise: "(i) the XML element or text the user has selected; (ii) external-items (that is, items in the document but outside the text selected by the user that are referred to (generally by ID) by that text or by a reference from that reference and so on); and ***(iii) a LogicSources element declaring logic required by (i) and (ii)***." EX1006-Harrop, 54:23-28 (emphasis added). A POSITA would understand that the included "LogicSources" are rules which are embedded in each module.

g) Limitation 1[D1] generating the derivative document by assembling the second modular item with the first document or by displaying or hiding the at least first modular item from the first document, wherein

121. The Hailey-Harrop combination renders this limitation obvious because each of Hailey and Harrop teach this limitation.

122. Hailey teaches generating a derivative document by assembling content (modular items) based on various input values (personalization parameters). EX1005-Hailey, ¶39. "The assembly module 30 retrieves data or content and rules

from the knowledge base 54 and assembles the content to create documents (such as documents 44, 46, and 48).” EX1005-Hailey, ¶66. A POSITA would understand that when a content is assembled in the generated document, the content is “displaying” as required by limitation 1[D1] since the generated document is ultimately displayed with included/assembled content. Where a second modular item is selected, as discussed regarding limitation 1[C2.A] above, the second modular item will be assembled and displayed as part of the generated document. EX1005-Hailey, ¶66; *see* §VIII.A.2.e), supra (incorporated here).

123. Similarly, Harrop teaches generating an instance document (interactive document) based on logic sources or modules (“reusable XML chunks”) 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. Harrop teaches the modular content included in the generated document is displayed to a user in HTML or PDF format. Thus, whether the first modular item or second modular item is assembled in the generated document, the included modular item will be displayed.

124. Thus, Hailey and Harrop each teach Limitation 1[D1] and the Haiely-Harrop combination renders this limitation obvious.

h) Limitation 1[D2]: the derivative document comprises a plurality of modular items that are not embedded with rules.

125. The Hailey-Harrop combination renders this limitation obvious because each of Hailey and Harrop teaches this limitation.

126. First, it is worth noting that the '651 Patent describes the embedding of rules in modular items in a master document, but the '651 Patent provides no description or example of an interactive document that ***does not*** include the assembly rule. *See* EX1001, 3:47-49. Notwithstanding a lack of supporting disclosure for this limitation, a POSITA would have been able to understand the plain and ordinary meaning of this limitation based on the claim language itself and would have found the claim to be obvious over the Hailey-Harrop combination.

127. 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, 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 a POSITA would have found it 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, 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, 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.

3. Claim 2: The method of claim 1, wherein the first modular item is stored in a structured data that defines the first set of rules.

128. In addition to all the limitations of claim 1, the Hailey-Harrop combination further renders claim 2 obvious. *See* §VIII.A.2, *supra* (incorporated here).

129. Hailey teaches storing content (*e.g.*, objects) in a knowledge base (shown as a runtime database 120 in FIGS. 8 and 4) from which the content (*e.g.* objects) for a particular document can be extracted. EX1005-Hailey, ¶95. Figure 4 (reproduced below) demonstrates that this data is stored in a structured manner.

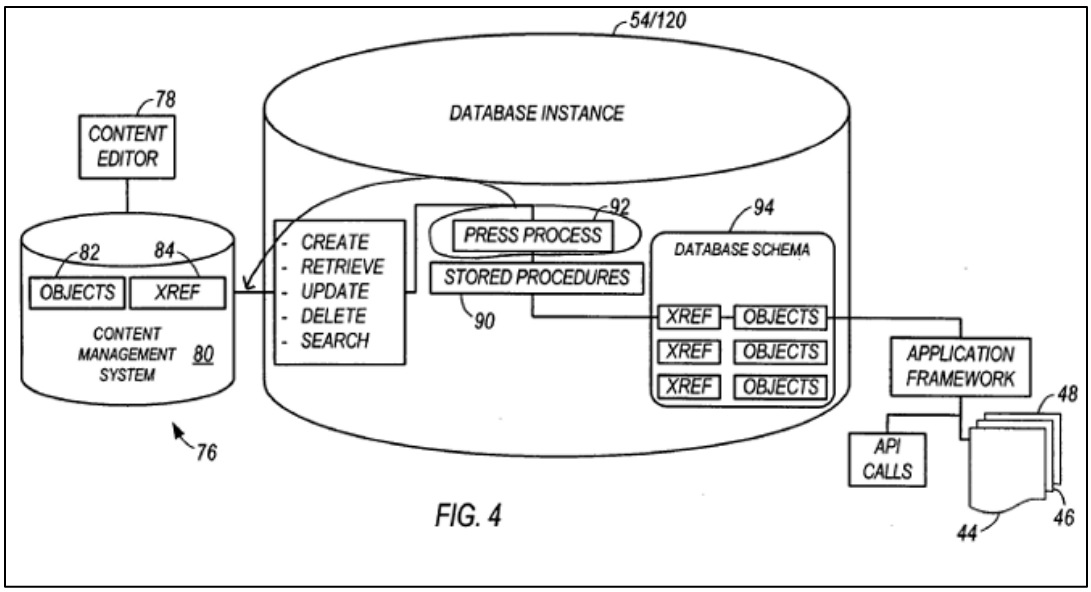


FIG. 4

130. Similarly, Harrop teaches that the source documents 208 XML documents (i.e., structured data files) in data files 110, which are stored in a hard drive (i.e., database). EX1006-Harrop, 7:21-31.

131. Accordingly, the Hailey-Harrop combination renders claim 2 obvious.

4. Claim 3: The method of claim 1, wherein the first set of rules define a link between the first modular item and another document or a rule for assembling the first modular item into a derivative document.

132. 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).

133. As discussed in §VIII.A.2.b), *supra* (incorporated here) regarding limitation 1[A], Hailey teaches that content (modular items) are stored in the database as groups of XREF (rules) and objects. *See* EX1005-Hailey, ¶¶39-41, 64; "The assembly module 30 retrieves data or content and rules from the knowledge base 54 and assembles the content to create documents (such as documents 44, 46, and 48)." EX1005-Hailey, ¶66. These XREF rules define a rule for assembling the modular items into a derivative document.

134. In addition, Harrop teaches generating an instance document (interactive document) based on logic sources or modules ("reusable XML chunks") 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. The logic sources define rules "used to determine values for variables required by a source document or other

logic source referring to that logic source." EX1006-Harrop, 9:11-12. The application of variables "determin[es] whether an associated portion of text is to be included in a generated instance document." *Id.*, 9:19-23.

135. Thus, both Hailey and Harrop teach a first set of rules which define a rule for assembling a first modular item into a derivative document.

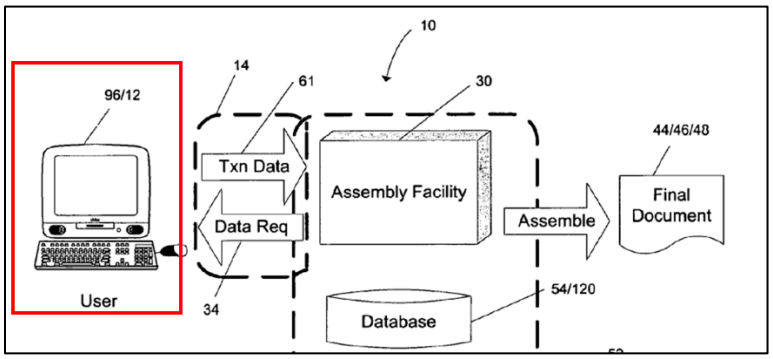
5. Claim 4: The method of claim 3, wherein the value of the one or more parameters is specified by a user via a user interface.

136. As discussed above, the Hailey-Harrop combination renders obvious all the limitations of claims 1 and 3, and the Hailey-Harrop combination further renders claim 4 obvious. §§VIII.A.2, VIII.A.4, *supra* (incorporated here).

137. Harrop teaches that a web browser 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) which allows a user to input or select answers and the user's answers to the displayed interview questions are parameters specified by a user.

138. 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. 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").



139. Accordingly, Hailey teaches that a user inputs "transaction data" into the computer via an included a user interface, which necessarily exists on a computer with a monitor which receives user input.

6. Claim 5: The method of claim 1, wherein the user interface displays one or more personalization fields that are automatically generated based on the first set of rules.

140. In addition to all the limitations of claim 1, the Hailey-Harrop combination further renders claim 5 obvious. *See* §VIII.A.2, *supra* (incorporated here).

141. Harrop teaches displaying interview questions to a user to solicit answers in order to resolve logic rules. EX1006-Harrop, 9:6-9 ("When a user of the system is used as a data source, the logic is resolved by performing one or more question and answer interview rounds that determine which components and text from the source documents 208, referenced logic sources 210 and user responses are

included in the resulting instance document."). The displayed interview questions are "personalization fields" which the user inputs or selects answers. *Cf.* EX1001, 8:30-33 ("a user may select a parameter from a plurality of options for a personalization field to customize a document that is to be delivered to a specific reader/customer."). Harrop further teaches generating the interview questions based on the variables that need to be resolved for the given XML source document:

> Variables that can be resolved without user input are resolved by the processing engine 202 from other data sources. ***The processing engine 202 provides the questions to which it requires responses to the Java controller servlet 218.*** It is the responsibility of that servlet 218 to return the responses to the processing engine 202. ***Where the questions are put to a user via their web browser, the controller servlet 218 can put all the questions to the user in a single HTML form***, or it can put the questions to the user one at a time, or it can group them by topic, asking the questions in a single topic in a given form.

EX1006-Harrop, 14:13-19 (emphasis added). Thus, Harrop teaches displaying questions (personalization fields) to a user via a GUI (web browser) which are automatically generated based on the modules / logic sources in the XML source document.

7. Claim 6: The method of claim 5, wherein at least one of the one or more personalization fields is generated based on the second set of rules associated with the second modular item.

142. In addition to all the limitations of claims 1 and 5, the Hailey-Harrop combination further renders claim 6 obvious. *See* §§VIII.A.2, VIII.A.6, *supra* (incorporated here).

143. As discussed above regarding limitation 1[C2.A], Harrop teaches that modules ("reusable XML chunks") may be incorporated by reference into a target document and that the referenced module includes logic sources for resolving the required text. EX1006-Harrop, 55:22-56:13; §VIII.A.2.e), supra (incorporated here). Harrop further teaches that logic sources define the question using "<Question>" tags and provides an example code:

```
<UserTextQuestion Columns="20" ID="LetterSubject" Rows="1" Type="Text">
  <Question>What is the subject of this letter?</Question>
    <Topic>Subject of Letter</Topic>
  </UserTextQuestion>
```

EX1006-Harrop, 35:32-36:3 (annotated). A POSITA would understand that when a reference to a module is included in a target document, Harrop's processing engine will include the question embedded in the referenced module as part of the interview questions presented to the user in order to resolve the logic of referenced module. Thus, the question (i.e., personalization field) presented to the user will be based on

the second set of rules (logic sources in the referenced module) associated with the second modular item (referenced module).

8. Claim 8: The method of claim 6, wherein the value of the one or more parameters is specified by a user in a text file.

144. In addition to all the limitations of claims 1, 5 and 6, the Hailey-Harrop combination further renders claim 8 obvious. *See* §§VIII.A.2, VIII.A.6, VIII.A.7, supra (incorporated here).

145. Harrop teaches that user response data may be stored in "HTML form" and re-applied to a source document in order to avoid unnecessary repetition of previous answers:

> It is useful to be able to store the response data, so that if a user wishes to modify some of the answers they have given in an earlier session, they can start by applying the stored response data to the source document, rather than having to answer all the questions again themselves. ***When an answer file containing stored response data is loaded, the questions and answers are displayed to the user in an HTML form.*** After the user has modified the responses as appropriate, and submitted the form (by selecting a "next" button), those interview items are provided to the evaluator module 402

EX1006-Harrop, 16:4-10; *see also* EX1006-Harrop, 7:22-23 ("The data files 110 may also include stored response data 215"), 9:18-19 (" A given set of responses to a set of questions can be stored as stored response data 215 on the hard disk 104 of the document assembly system."). A POSITA would understand that Harrop's

system allows for the user to edit answers via text input and the "HTML form" taught by Harrop is a text file that contains the user's answers.

146. Further, to the extent that there is any argument that an "HTML form" file is not a "text file," a POSITA would have found it obvious to store user response data in a text file since text files have been known and used to electronic store data since nearly the invention of the computer.

9. Claim 9: The method of claim 1, wherein the first set of rules includes a mapping relationship that maps the first modular item to a style for rendering the first modular item.

147. In addition to all the limitations of claim 1, the Hailey-Harrop combination further renders claim 9 obvious. See §VIII.A.2, supra (incorporated here).

148. Hailey teaches using document-specific style attributes to specify formatting styles of the generated derivative document:

> The document element includes a number of attributes 153. In the example shown, the attributes include a source attribute, a product family attribute, a document class attribute, a recordable attribute, an applicable law state attribute, a margins state attribute, ***a document-specific style attribute***, and a source name attribute. The attributes **153** may be used for a variety of purposes including… to indicate specific margin requirements of the state or other government entity; to specify formatting styles….

EX1005-Hailey, ¶79 (emphasis added). Hailey further teaches that the rules 56 are used to select and generate a document, a question set, o***r a style sheet***." EX1005-

Hailey, ¶43 (emphasis added). Hailey teaches that these "style sheets or other formatting mechanisms are applied" to the output document. EX1005-Hailey, ¶42. Accordingly, Hailey teaches rules that include a mapping relationship to a style (i.e., a style sheet) that is used to render the generated document in a certain format or style.

10. Claim 10: The method of claim 1, wherein the derivative document is displayed to a reader using the mapped style.[6]

149. In addition to all the limitations of claim 1, the Hailey-Harrop combination further renders claim 10 obvious. See §VIII.A.2, supra (incorporated here). As discussed regarding claim 9 above, Hailey teaches that the XREF rules may include formatting or a style sheet which are applied to the output/generated document. *See* §VIII.A.9, supra (incorporated here). It is known that style sheets are a known method for applying formatting, color, and other aesthetic choices to a generated document. Style sheets (such as those described using CSS) are commonly used to control the visual presentation of XML and HTML data-based files. CSS can be used to control colors, fonts, sizes as well as images. It is the standard way to control visual formatting and branding of XML and HTML files. As such, it teaches the use of mapped styles, as a POSITA would understand the switching style sheets

[6] I am analyzing Claim 10 as if it properly depends from claim 9 which provides antecedent basis for "the mapped style" recited in Claim 10. Without dependency to claim 9, claim 10 appears to lack antecedent basis for "the mapped style."

would provide a "mapping" to a different style. Hailey also teaches that the output document is displayed to the user as an HTML, pdf, or other file. EX1005-Hailey, ¶42. Accordingly, the output file will be rendered using the style sheets or formatting rules defined by the XREF rules.

11. Claim 13: The method of claim 1, further comprising storing the one or more parameters and a reference to the first document for updating the derivative document.

150. In addition to all the limitations of claim 1, the Hailey-Harrop combination further renders claim 13 obvious. See §VIII.A.2, supra (incorporated here). The Hailey-Harrop combination teaches this limitation.

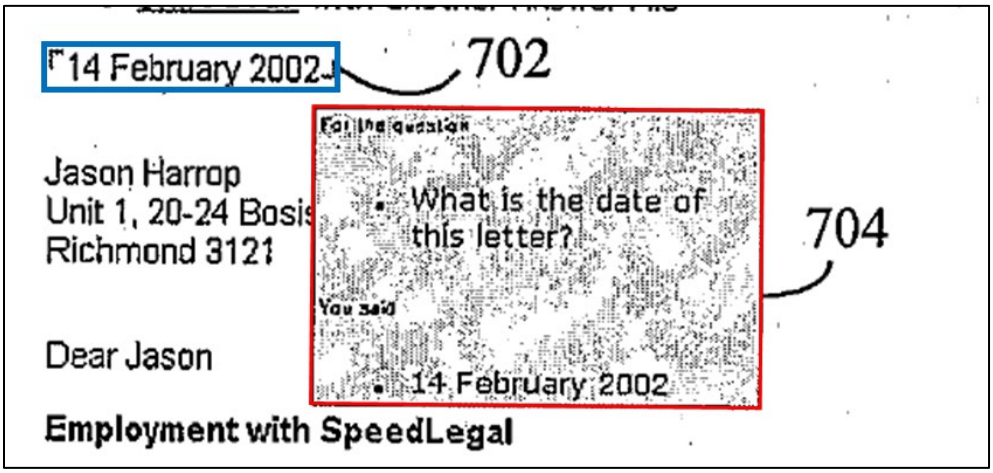
151. Claim 13 recites storing "one or more parameters and a reference to the first document for updating the derivative document." The '651 Patent describes that a "the personalization parameters (e.g., fields in the personalization form) that were provided during the personalization process and a reference 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, 19:33-38. The '651 Patent does not describe any form or data structure for the "link or connection."

152. First, Harrop expressly teaches storing a user's answers for reference in creating derivative documents based on the same XML source document or based on other source documents:

> Controls 712 are also provided for storing responses on the system as stored response data 215 and for applying stored responses to the source document they came from, or to another source document that uses some of the same logic sources, and for processing groups of related documents.
>
> It is useful to be able to store the response data, so that if a user wishes to modify some of the answers they have given in an earlier session, they can start by applying the stored response data to the source document, rather than having to answer all the questions again themselves. When an answer file containing stored response data is loaded, the questions and answers are displayed to the user in an HTML form.

EX1006-Harrop, 15:30-16:8. Thus, Harrop teaches storing the one or more parameters (user's answers).

153. Further, Harrop teaches creating a hypertext anchor (a reference) that links the one or more parameters (i.e., user responses) to the derivative 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 (i.e., the parameters) that resulted in that particular text being included in the rendered derivative 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).

154. 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 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. 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 XML source document (i.e., the first document) located on server.

155. 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 Harrop teaches updating the instance document using first a link from the derivative document to the XML source document and then using a link back to the derivative document to include updated content based on a user's changed responses.

156. Accordingly, the Hailey-Harrop combination teaches a processor storing the one or more parameters and a reference to the first document for updating the derivative document as claimed.

12. Claim 14: The method of claim 1, further comprising converting the derivative document into a master document by storing the one or more parameters as a set of rules associated with the derivative document.

157. In addition to all the limitations of claim 1, the Hailey-Harrop combination renders claim 14 obvious. *See* §§VIII.A.2, *supra* (incorporated here).

158. Claim 14 requires a derivative document to be converted to a master document by "storing the one or more ***parameters*** as a set of rules associated with the derivative document." However, aside from parroting the claim language, the '651 patent provides no description or example where a set of parameters are stored "as a set of rules" nor does the patent provide any example where storing parameters "as a set of rules" results in the conversion of a derivative document to a master document. *See generally*, EX1001. Regardless, to the extent that the claim requires

converting the derivative document by storing the personalization parameters as a set of rules, it is my opinion that the Hailey-Harrop combination renders this claim obvious because Harrop teaches this limitation.

159. Harrop teaches that the user can select options to decide how the instance document will be further converted to a output format. Specifically, Harrop teaches that the user can select to generate an output document "in HTML (without controls), XML, or in rich text format" and that the user can select whether the output document includes "the instance document text, editorial notes… and/or the interview questions and answers on which the document was based." EX1006-Harrop, 15:18-28. Further, Harrop describes that the "controls" mentioned above "are provided for storing responses on the system as stored response data [] and for applying stored responses to the source document they came from, or to another source document that uses some of the same logic sources, and for processing groups of related documents." EX1006-Harrop, 15:30-16:2. Harrop thus teaches converting a derivative document into an XML format that includes controls, the document content, user notes, and the interview questions and answers (parameters) associated with the derivative document. A POSITA would understand that Harrop's outputting a derivative document to XML format with controls, content, notes, and questions/answers would result in a master document that can be applied to another source document. *See* EX1006-Harrop, 15:30-16:2.

13. Claim 15

160. Claim 15 is directed to a computer-implemented method, reciting limitations that are substantially identical to the recitations in claim 1. Thus, claim 15, for at least the same reasons as claim 1, is unpatentable over Hailey and Harrop. VIII.A.ii, *supra* (incorporated here).

a) Limitation 15[PRE]: 15. A method for personalizing an interactive document comprising:

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

b) Limitation 15[A1] (a) creating one or more modular items of a first document, wherein at least one modular item is embedded with a first set of rules defining a link between the at least one modular item and a second document or a first set of rules for assembling the at least one modular item into a derivative document,

162. As discussed regarding limitations 1[A], 1[C1], and 1[C2.A] the Hailey-Harrop combination renders this limitation obvious. *See* §§VIII.A.2.b), VIII.A.2.d), VIII.A.2.e), *supra* (incorporated here).

c) Limitation 15[B]: (b) receiving an instruction for personalizing the first document;

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

d) Limitation 15[C1]: (c) assembling a plurality of modular items to generate the derivative document of the first document;

164. As discussed regarding limitations 1[C1] and 1[D1], the Hailey-Harrop combination renders this limitation obvious. *See* §§VIII.A.2.d), VIII.A.2.g), *supra* (incorporated here).

e) Limitation 15[C2]: wherein the plurality of modular items comprise the at least one item displayed or hidden from the given document based on the first set of rules and the instruction or

165. As discussed regarding limitations 1[C1], 1[C2.A], and 1[C2.B], the Hailey-Harrop combination renders this limitation obvious. See §§VIII.A.2.d)-f), supra (incorporated here).

f) Limitation 15[C3]: the plurality of modular items are pulled from the first document and the second document based at least in part on the first set of rules embedded with the at least one modular item of the first document, wherein the first document and the second document are master documents that one or more modular items of the second document are embedded with rules; and

166. As discussed regarding limitations 1[C2.A] and 1[C2.B], the Hailey-Harrop combination renders this limitation obvious. *See* §§VIII.A.2.e)-f, *supra* (incorporated here).

g) Limitation 15[D]: (iii) rendering the derivative document using a style associated with the first document and

167. As discussed regarding claims 9 and 10, the Hailey-Harrop combination renders this limitation obvious. *See* §VIII.A.9, VIII.A.10, supra (incorporated here).

h) Limitation 15[E]: wherein the plurality of modular items of the derivative document are not embedded with rules.

168. As discussed regarding limitations 1[D2], the Hailey-Harrop combination renders this limitation obvious. *See* §VIII.A.2.h), *supra* (incorporated here).

14. Claim 16: The method of claim 15, further comprising providing a first user interface for receiving a user input to generate the first set of rules.

169. In addition to all the limitations of claim 15, the Hailey-Harrop combination further renders claim 16 obvious. *See* §VIII.A.13, *supra* (incorporated here).

170. Hailey teaches that a GUI is provided to the user which allows the user to modify documents via the editing module 20. *See* EX1005-Hailey, ¶39 ("If desired, the document Selection engine 16 may return the listing of documents 18 to the origination platform for display in a graphical user interface ("GUI") (not shown) presented to a user Such that the user may choose to modify the list of documents in the editing module 20."). Hailey describes the editing module as "editing module or

interface 20." Thus, Hailey teaches a user interface for receiving user input to generate the first set of rules.

171. Harrop also teaches "an editor 112 for creating and editing the data files 110." EX1006-Harrop, 7:3-4. "The editor 112 includes a user interface module 302…. the user interface 302 and XML editor 304 modules allow a user of the document assembly system to add, modify, or delete XML elements, attributes, comments, processing instructions, and text." EX1006-Harrop, 10:30-11:3. Thus, Harrop teaches a user interface for receiving a user input to generate the first set of rules.

15. Claim 17: The method of claim 16, wherein the first user interface displays a plurality of fields for defining the first set of rules.

172. In addition to all the limitations of claim 16, the Hailey-Harrop combination further renders claim 17 obvious. *See* §VIII.A.14, *supra* (incorporated here).

173. As discussed above, Harrop teaches an editor 112 which includes a user interface which "allow a user of the document assembly system to add, modify, or delete XML elements, attributes, comments, processing instructions, and text." EX1006-Harrop, 11:1-3. Further, the XML editor module 304, which is part of the editor 112, "provides standard XML editor functions pl***us the ability to add and manipulate condition references*** reusable text, user text, notes, and meta-

information processing instructions around and within document components." EX1005-Hailey, 11:3-6. The editing of "condition references" means that fields are present in the editor 112 which allow the user to define the "first set of rules" (the condition references in the XML elements).

16. Claim 18: The method of claim 15, further comprising providing a second user interface for receiving user input to customize the one or more modular items.

174. As discussed regarding claim 4, the Hailey-Harrop combination renders this claim obvious. *See* §VIII.A.5, *supra* (incorporated here).

17. Claim 19: The method of claim 15, wherein the first set of rules embedded with the at least one [modular] item of the first document includes a mapping relationship that maps the at least one item to the style.[7]

175. As discussed regarding claims 9 and 10, the Hailey-Harrop combination renders this claim obvious. *See* §VIII.A.9, VIII.A.10, *supra* (incorporated here).

18. Claim 20: The method of claim 15, further comprising providing a user interface for receiving the instruction for personalizing the first document.

176. In addition to all the limitations of claim 15 and as discussed regarding limitation 1[B] and claim 16, the Hailey-Harrop combination renders this claim obvious. *See* §§VIII.2.c), VIII.13, VIII.14, *supra* (all incorporated here).

[7] The word "modular" is not present in the claim but it appears that the claim may have intended to refer to the "at least one modular item."

B. Ground 2: Claims 11 and 12 are rendered obvious by Hailey and Harrop in further view of Davis

1. Technology Background – User Tracking

177. Since (practically) the beginning of the web, user tracking has been known and implemented ubiquitously. 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.

178. 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.

179. As discussed below, 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. 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

180. A POSITA would consider Harrop, Hailey, Davis and the '651 Patent to be analogous art. As discussed above, Harrop, Hailey, and the '651 Patent are analogous art. *See* §VIII.1.a), *supra*. Davis is also analogous art to Hailey, Harrop, and the '651 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 '651 Patent. *See* EX1005-Hailey, ¶42 (output file in HTML format; 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

181. Davis teaches embedding a tracking features in an derivative document to collect data indicative of user interaction with the document. EX1007-Davis,

5:14-18; *see* §VII.B, 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.

182. Accordingly, 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, a POSITA would have known that user tracking information can be used to understand what works well and what does not for their users, thereby improving system designs and ensuring the system works as intended.

c) Reasonable Expectation of Success

183. The references include similar arrangements of well-understood components and use predictable technology in routine and conventional ways.

184. 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"). 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 11: The method of claim 10, further comprising collecting data about the reader's interaction with the derivative document to extract insight data.

185. In addition to all the limitations of claims 1 and 10, the Hailey/Harrop/Davis combination further renders claim 11 obvious.

186. Davis teaches a tracking feature that collects data about the reader's (user's) interaction with the derivative document to extract insight data. 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."). Davis further teaches that the acquired data is "used to build historical profiles of individual users, to serve out highly targeted information based upon user profiles, as well as to extract information about how much data was

downloaded by a respective client, and how long or how often specific files were displayed or in use by the client."

187. As discussed above, 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 11.

4. Claim 12: The method of claim 11, further comprising receiving an instruction for personalizing the first modular item or the second modular item.

188. In addition to all the limitations of claim 11, the Hailey-Harrop combination further renders claim 12 obvious. *See* §VIII.B.3, *supra* (incorporated here). Further, as discussed regarding claim 17 above, Harrop teaches an editor 112 which receives instructions (input) from a user to personalize the modular items in the source documents. *See* §VIII.A.15, supra (incorporated here). Therefore, the Hailey/Harrop/Davis combination renders claim 12 obvious.

C. Ground 3: Claim 7 is rendered obvious by Harrop and Hailey in further view of Lo

189. The features of claim 7 are taught by Lo and rendered obvious when combined the teachings of Hailey and Harrop.

1. Reasons to Combine

a) Analogous Art

190. Lo is analogous art to the Harrop, Hailey, and Davis, as well as the '651 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, Harrop, Davis, and the '651 Patent.

b) Motivation to combine

191. A POSITA would have been motivated to combine the teachings of Lo with the teachings of Haiely 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. 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.

192. A POSITA would have been motivated to combine Lo's teaching of live-proofing previews with the teachings of Hailey and/or Harrop to enable a user to find errors in document design at an earlier stage, as taught by Lo.

c) Reasonable Expectation of Success

193. A POSITA would have had a reasonable expectation of success in modifying the systems of Hailey and/or 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.5, supra (incorporated here). A POSITA would have found it routine to modify the already-existing user interfaces of Hailey and/or Harrop to include Lo's teachings regarding the live-proofing preview feature.

2. Claim 7: The method of claim 6, further comprising: upon receiving an input in a given personalization field, displaying at least a portion of the content from the first document that is related to the given personalization field.

194. As discussed above, the Hailey-Harrop combination renders claim 6 obvious. Claim 7 is rendered obvious over Hailey and Harrop in further view of Lo. *See* §VIII.A.7, *supra* (incorporated here);

195. 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.

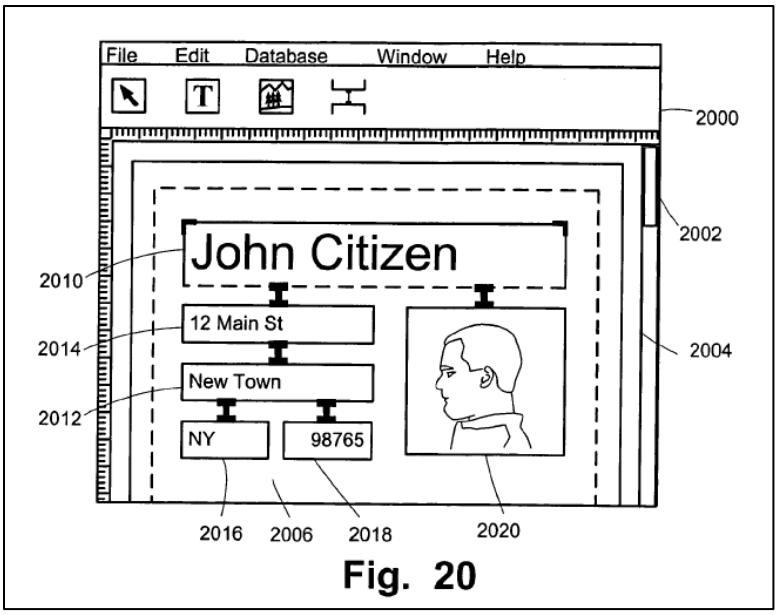


Fig. 20

196. Thus, Lo teaches displaying a portion of the content from the first document that is related to a personalization field (Lo's "variable data") in real-time as shown in figure 20.

197. For the reasons discussed above, 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 preview of how the generated interactive document would

change depending on changes to input variables. *See* §VIII.C.1, *supra* (incorporated here); Accordingly, the Hailey-Harrop combination as modified by Lo renders claim 7 obvious.

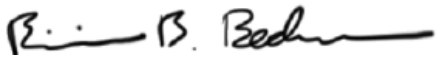
IX. CONCLUSION

198. For these reasons, it is my opinion that claims 1-20 of the ’651 Patent are not novel and, within my understanding as an external technical expert, appear to be unpatentable.

199. 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.

200. 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.