

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,
Petitioner,

v.

GOTV STREAMING, LLC
Patent Owner.

Case No. IPR2023-00759

U.S. Patent No. 8,103,865

**DECLARATION OF BENJAMIN B. BEDERSON
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,103,865**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS	2
III.	SCOPE OF OPINIONS	16
IV.	MATERIALS REVIEWED AND CONSIDERED	16
V.	LEVEL OF ORDINARY SKILL IN THE ART	17
VI.	STATEMENT OF LEGAL PRINCIPLES.....	18
A.	Claim Construction.....	18
B.	Anticipation	18
C.	Obviousness.....	19
VII.	TECHNOLOGY BACKGROUND AND DISCUSSION OF THE STATE OF THE ART	20
A.	Graphics, Rendering, and Graphical User Interfaces.....	20
B.	Wireless Devices	24
C.	Client-Server Architectures.....	25
D.	Overview of CSS.....	31
E.	Overview of JavaScript and the Document Object Model.	32
F.	Overview of U.S. Patent Application Publication No. 2007/0150617 to Hariki, filed on July 25, 2006 (“ <i>Hariki</i> ”).	35
G.	Overview of U.S. Patent Application Publication No. 2003/0023755 to Harris et al., filed on December 18, 2001 (“ <i>Harris</i> ”).	38
H.	Overview of U.S. Patent Application Publication No. 2003/0137539 to Dees et al., filed on October 1, 2002 (“ <i>Dees</i> ”).	42
I.	Other Evidence Regarding the State of the Art.....	44

VIII. SUMMARY OF THE '865 PATENT	44
A. The '865 Patent Specification	44
B. The '865 Patent File History (Ex. 1004)	48
C. Challenged Claims	50
D. Claim Construction (37 C.F.R. § 42.104(b)(3))	55
IX. PRIORITY DATE OF THE '865 PATENT	55
X. APPLICATION OF THE PRIOR ART TO CLAIMS 1–20, 31, 34	56
A. Ground 1: <i>Hariki</i> in view of <i>Harris</i> renders obvious Claims 1- 20, 31, 34	56
B. Ground 2: <i>Hariki</i> in view of <i>Harris</i> and <i>Dees</i> renders obvious Claims 1-20, 31, 34	108
XI. Secondary Considerations	122
XII. CONCLUSION	122

EXHIBIT LIST

Exhibit Number	Description
1001	U.S. Patent No. 8,103,865 (“’865Pat”)
1002	Expert Declaration of Ben Bederson (“ <i>Bederson</i> ”)
1003	Curriculum Vitae of Benjamin B. Bederson, Ph.D.
1004	File History of U.S. Patent No. 8,103,865 (“’865FH”)
1005	U.S. Patent Application Publication No. 2007/0150617 (“ <i>Hariki</i> ”)
1006	U.S. Patent Application Publication No. 2003/0023755 (“ <i>Harris</i> ”)
1007	U.S. Patent Application Publication No. 2003/0137539 (“ <i>Dees</i> ”)
1008	RESERVED
1009	U.S. Patent No. 7,447,486 (“ <i>Tamura</i> ”)
1010	U.S. Patent No. 6,996,627 (“ <i>Carden</i> ”)
1011	U.S. Patent No. 6,669,564 (“ <i>Young</i> ”)
1012	U.S. Patent No. 6,732,183 (“ <i>Graham</i> ”)
1013	U.S. Patent No. 6,507,727 (“ <i>Henrick</i> ”)
1014	Webpage excerpt from Computer History Museum at https://www.computerhistory.org/tdih/april/6/ (last accessed March 29, 2023)
1015	https://www.webdesignerdepot.com/2009/05/the-evolution-of-cell-phone-design-between-1983-2009/
1016	Excerpts from Dan R. Olsen, Jr., Developing user interfaces (1998)
1017	Alok Sinha, Client-server computing, in Communications of the ACM, 35, 7 (1992)

Exhibit Number	Description
1018	Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret, The World-Wide Web. Communications of the ACM 37, 8 76-82 (August 1994), available at http://dx.doi.org/10.1145/179606.179671
1019	Eric Kasten, HTML: A Gentle Introduction, Linux Journal (July 1995), available at https://www.linuxjournal.com/article/1081
1020	U.S. Patent Application Publication No. 2013/0124961 (“Linburn”)
1021	U.S. Patent No. 8,111,326 (“Talwar”)
1022	Docket Control Order, <i>GovTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556-RGK-SHK, Dkt. 61 (C.D. Cal. Feb. 13, 2023)
1023	Complaint in <i>GovTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556-RGK-SHK (C.D. Cal. October 17, 2022)
1024	https://web.archive.org/web/20070406171650/http://www.zenzui.com/products.html
1025	Bert Boss, <i>A brief history of CSS until 2016</i> , W3 Blog (Dec. 17, 2016), https://www.w3.org/Style/CSS20/history.html
1026	Eric A. Meyer, <i>Cascading Style Sheets – The Definitive Guide</i> (O’Reilly, 2d ed. 2004)
1027	<i>JavaScript</i> , Wikipedia (Aug. 12, 2004), https://web.archive.org/web/20040812085853/http://en.wikipedia.org/wiki/JavaScript
1028	<i>Document Object Model (DOM) Level 1 Specification Version 1.0</i> , W3C Recommendation (Oct. 1, 1998), https://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/DOM.pdf
1029	<i>ECMAScript: A general purpose, cross-platform programming language</i> , Standard ECMA-262 (June, 1997), https://www.ecma-international.org/wp-content/uploads/ECMA-262_1st_edition_june_1997.pdf

Exhibit Number	Description
1030	Peter-Paul Koch, <i>The Document Object Model: an Introduction</i> , Digital Web Magazine (May 14, 2001), https://www.digital-web.com/articles/the_document_object_model/

I. INTRODUCTION

1. My name is Benjamin B. Bederson, and I have been retained by counsel for Netflix Inc. (“Netflix” or “Petitioner”) to submit this Declaration in support of Netflix’s petition for *Inter Partes* Review (“IPR”) and cancellation of claims 1–20, 31, and 34 (the “Challenged Claims”) of U.S. Patent No. 8,103,865 (“the ’865 patent” or “’865*Pat*”) (Ex. 1001), which I understand has been assigned to GoTV Streaming, LLC (“GoTV” or “Patent Owner”).

2. I have been asked to opine on whether the ’865 patent is anticipated and/or rendered obvious by the prior art. My opinions are based on my years of education, research and experience, as well as my investigation and study of relevant materials. The materials that I considered for this Declaration include all exhibits cited in this Declaration.

3. I may rely upon these materials, my knowledge and experience, and/or additional materials to rebut arguments raised by Patent Owner. Further, I may also consider additional documents and information in forming any necessary opinions, including documents that may not yet have been provided to me.

4. My analysis of the materials produced in this investigation is ongoing and I will continue to review any new material as it is provided. This Declaration represents only those opinions I have formed to date. I reserve the right to revise,

supplement, and/or amend my opinions stated herein based on new information and on my continuing analysis of the materials already provided.

5. I am being compensated at my standard consulting rate of \$600/hour for my time spent working on issues in this matter. I am also being reimbursed for any reasonable and customary expenses associated with my work and testimony. I have no financial interest in, or affiliation with, Petitioner or Patent Owner. My compensation is not dependent upon the outcome of, or my testimony in, the present IPR or any litigation proceedings.

II. QUALIFICATIONS

6. I am currently Professor Emeritus of Computer Science at the University of Maryland (“UMD”). From 2014 to 2018, I was the Associate Provost of Learning Initiatives and Executive Director of the Teaching and Learning Transformation Center at the UMD. 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. Last year, I co-founded the J.S. Bryant School, a therapeutic high school to launch in 2025. 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 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 prior Chief Technology Officer of Hazel Analytics from 2014 to 2023, a data analytics company to improve food safety and better public health whose product sends alerts in warranted circumstances.

8. In addition, I have for more than 25 years consulted for numerous companies in the area of user interfaces, including EPAM, Hillcrest Labs, Lockheed Martin, Logitech, Microsoft, NASA Goddard Space Flight Center, the Palo Alto Research Center, and Sony.

9. The devices and methods claimed in the '865 Patent generally relate to human-computer interaction, graphics and rendering on wireless devices. For more than 30 years, I have studied, designed, and worked in the field of computer science and HCI. My experience includes 30 years of teaching and research, with research interests in HCI and the software and technology underlying today's interactive computing systems with a focus on wireless devices, graphics and rendering.

10. At UMD, I have been focused primarily on the area of HCI, a field that relates to the development and understanding of computing systems to serve users' needs. Researchers and practitioners 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.

11. Example systems following this approach that I have built include Cortex-I (1992 embedded computer vision system that sensed licensed plates with custom motor, camera and controller), Audio Augmented Reality (1995 embedded system for sensing a user's location and playing audio suited to that location), Fisheye Menus (2000 software for sensing movement within and selection of linear list of items in a menu), PhotoMesa (2001 software for end users to browse personal photos), DateLens (2002 software for end users to use their mobile devices to efficiently access their calendar information), SlideBar (2005 linear sensor to control scrolling), LaunchTile (2005 "home screen" software for mobile devices to allow users to navigate apps in a zoomable environment), SpaceTree (2002 software for end users to efficiently browse very large hierarchies), ICDL (2001 International Children's Digital Library), and StoryKit (a 2009 iPhone app for children to create stories).

12. LaunchTile led to my co-founding of Zumobi Inc. in 2006, where I was responsible for investigating new software platforms and developing new user interface designs that provided efficient and engaging interfaces to permit end users to access a wide range of content on mobile platforms (including the iPhone and Android-based devices). For example, I designed and implemented software called

“Ziibii,” a “river” of news for iPhone that used a capacitive sensor for controlling linear movement through news, software called “ZoomCanvas,” a zoomable user interface for several iPhone apps, and iPhone apps including “Inside Xbox” for Microsoft and Snow Report for REI.

13. Starting in the mid-1990s and going through the 2000s, I worked on a range of “zoomable user interfaces” (ZUIs) that supported users in navigating large information spaces by zooming in and out of the information on the screen. I built several different ZUI systems over the years, including Pad++, Jazz, and Piccolo. I wrote Pad++ in C++ using the X Windows System including a number of techniques to support high performance interactive visualization.¹

14. Pad++ included a custom web browser which I built from scratch in 1994 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”².

¹ Bederson, B.B., & Meyer, J. (1998). Implementing a Zooming User Interface: Experience Building Pad++. *Software: Practice and Experience*, 28(10), 1101-1135.

² Bederson, B. B. and Hollan, J. D. (1994). “Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics,” *ACM Conference on User Interface Software and Technology*, 17-26.

15. It supported the version of HTML available at the time (HTML 1.0) as well as an embedded scripting language. The following images are from a custom web browser that I built that used a “zoomable user interface” to show a visual representation of which pages a user visited.³

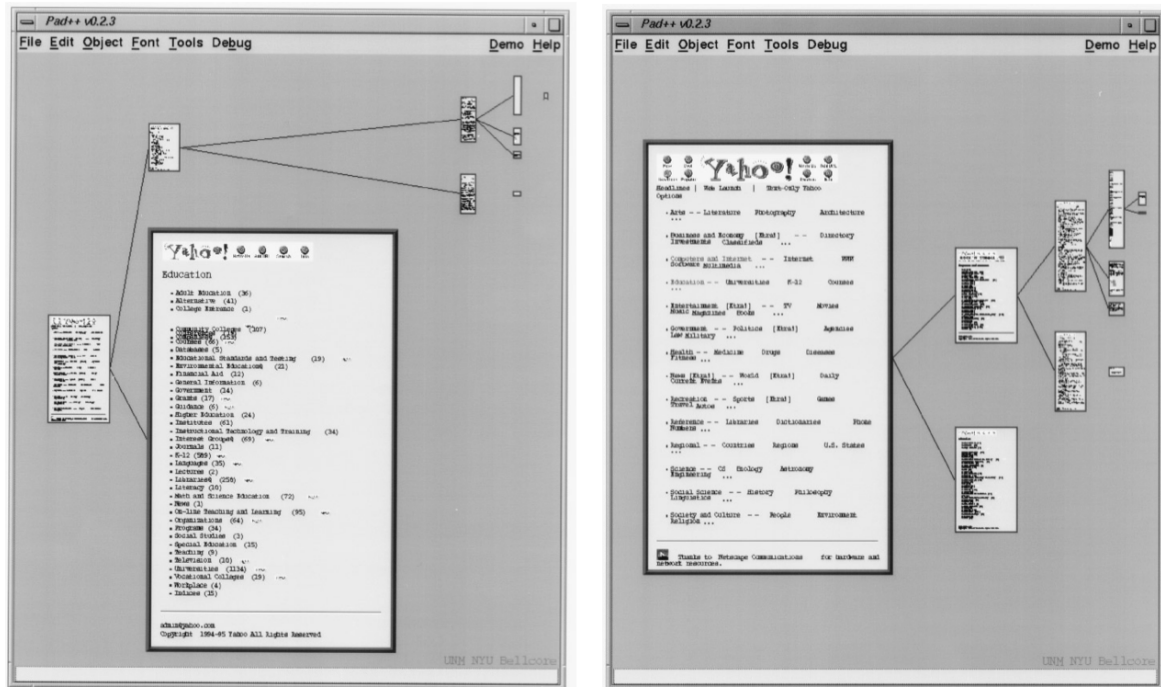


Figure 5. Many different HTML documents loaded in Pad++. Their layout implicitly shows the history of the user's interaction. The two views show the same tree focused on different nodes

16. By 1998, my team had combined our web browser 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

³ Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. (1996). Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *Journal of Visual Languages & Computing*, 7(1), 3-32.

“PadPrints”.⁴ 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 passing the traffic through and monitoring it, it can modify the data itself, or hold that connection and open 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.

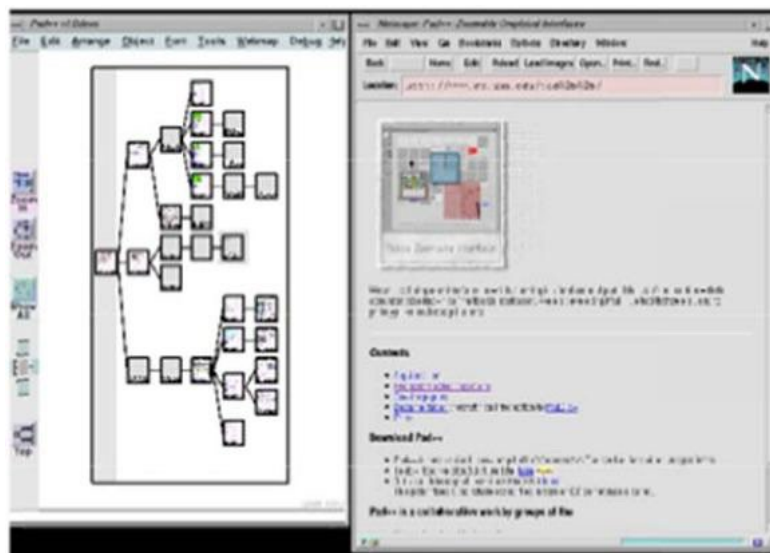


Figure 1 from PadPrints paper

⁴ Hightower, R.R., et. al., “Graphical Multiscale Web Histories: A Study of PadPrints,” Proceedings of the ninth ACM conference on hypertext and hypermedia: links, objects, time and space---structure in hypermedia systems, 58-65, 59 (1998).

17. Starting in 2001 and continuing for several years, I worked on photo management systems. I wrote a paper⁵ in 2001 describing PhotoMesa, a zoomable image browser using a novel 2D layout algorithm I called “quantum treemaps” to position the images on the screen in a way that attempted to fill the screen while keeping groups of related images together. As indicated in the figure below, PhotoMesa could display many hundreds of images at a time by showing small thumbnails, by showing larger thumbnails when the mouse hovered over a small thumbnail, and by allowing a user to zoom in to see a high resolution version of the images.

⁵ Benjamin B. Bederson. 2001. PhotoMesa: a zoomable image browser using quantum treemaps and bubblemaps. In Proceedings of the 14th annual ACM symposium on User interface software and technology (UIST '01). Association for Computing Machinery, New York, NY, USA, 71–80. DOI: <https://doi.org/10.1145/502348.502359>

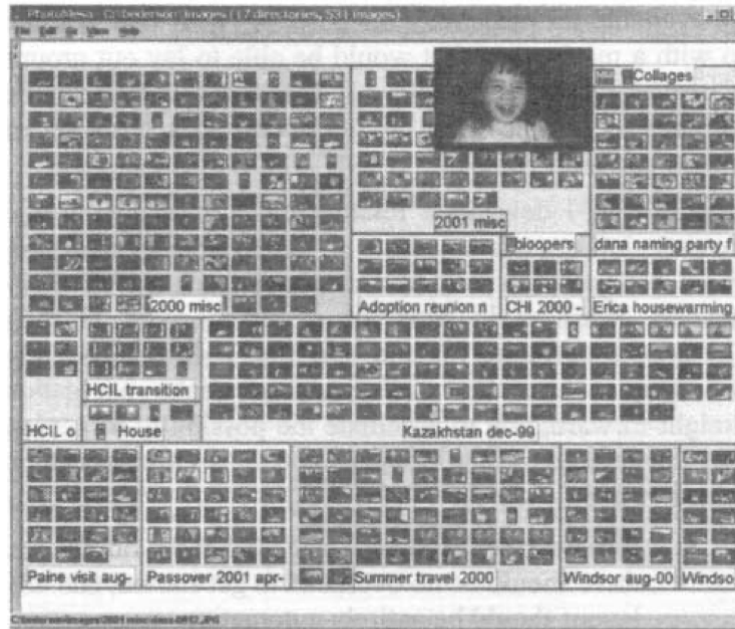


Figure 1: Screen snapshot of PhotoMesa with over 500 images in 17 groups.

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



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

19. I was heavily involved in the research and development of mobile device applications during the early 2000's. For example, I envisioned, designed and built DateLens in 2002-2004 to create a richer and more usable calendar for the Microsoft PocketPC platform as well as desktops. One of its innovations was its design that enabled it to scale between small (mobile) and large (desktop) computers. It also had a special double-headed scroll bar that allowed a user to touch scroll and also control how much information was presented on the screen at a time. I started

DateLens as a research project and eventually sold it commercially. DateLens displayed calendar data that came straight from the Microsoft PocketOutlook database and allowed modification and viewing of appointments. The research website from that time period that describes DateLens is still available at <http://www.cs.umd.edu/hcil/datelens/> which shows screenshots and a picture of me demonstrating DateLens to Bill Gates. The commercial website from that time period is also available at <http://www.windsorinterfaces.com/datelens.shtml>, which shows a number of news articles reviewing DateLens at that time.

20. In approximately 2004-2005, in collaboration with graduate students Amy Karlson and John SanGiovanni of Microsoft, I worked on a prototype software called LaunchTile. LaunchTile allowed zooming and scrolling of multiple applications on a handheld touchscreen PDA. The interface offered a grid of applications that could be zoomed and scrolled. It included several different methods of scrolling, including gestures where the scrolling continued after a finger was lifted from the touch screen, and then slowed to a stop.⁶

⁶Amy K. Karlson, Benjamin B. Bederson, and John SanGiovanni. AppLens and LaunchTile: two designs for one-handed thumb use on small devices. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05) (2005). ACM, New York, NY, USA, 201-210. DOI: <https://doi.org/10.1145/1054972.105500>.

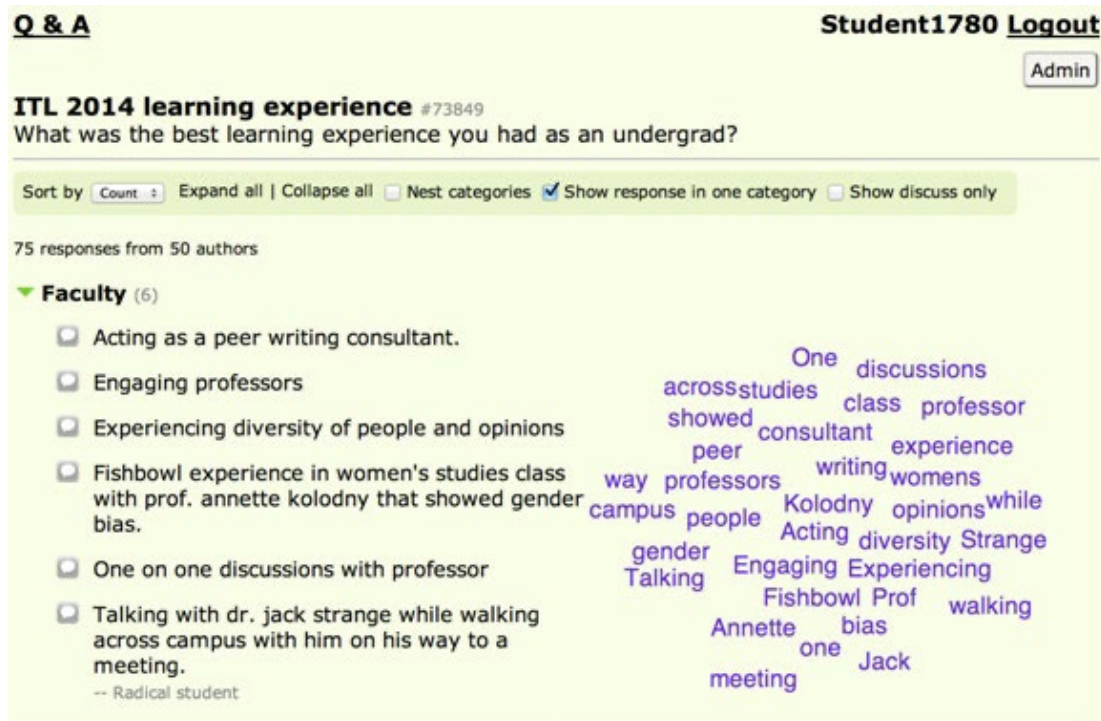
21. In 2007 and 2008, I wrote an app for Apple iOS called “ICDL on iPhone” which provided a scrollable and zoomable interface for reading children's picture books from my International Children's Digital Library on iPhones. At the time, this was the only effort of its kind to provide the ability read children's picture books on phones.

22. That led to my design and development of StoryKit starting in 2009. StoryKit is an iOS app that supports children to write their own stories, including text, images, drawings and sound. It has become quite popular among school teachers and for many years has gotten about 10,000 hours a month of use of children writing stories.

23. 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 UMD 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++/>.

24. More recently, I have worked on complex web “apps,” some of which include search results. These include a system called “Q&A” which used AJAX to support real-time support for classroom interactive response by students, with

collaborative annotation of each other's responses. The following shows the "Q&A" interactive student response web site. This screenshot shows the results of 50 students simultaneously answering a question and then annotating each other's annotations.



Screen capture taken July 18, 2014 by Ben Bederson from
<http://qaumd.appspot.com/results/73849>

25. I received the Janet Fabri Memorial Award for Outstanding Doctoral Dissertation for my Ph.D. work in robotics and computer vision. 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.

26. 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 the 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.

27. 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.”

28. 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 HCI. 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.⁷

29. 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 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.

30. I have designed, programmed, and publicly deployed dozens of user-facing software products that have cumulatively had millions of users. My work is cited by several major companies, including Amazon, Apple, Facebook, Google, and Microsoft. I am a named inventor on 14 U.S. patents and 20 U.S. patent applications. These patents are generally directed to user interfaces and experience.

31. I received a B.S. degree in Computer Science with a minor in Electrical Engineering in 1986 from the Rensselaer Polytechnic Institute. I received M.S. and

⁷<https://ieevis.org/year/2021/info/awards/test-of-time-awards>;
<https://ieevis.org/year/2022/info/awards/test-of-time-awards>

Ph.D. degrees in Computer Science in 1989 and 1992, both from New York University.

III. SCOPE OF OPINIONS

32. I have been asked to provide my opinions regarding whether:

- Claims 1-20, 31 and 34 of the '865 patent would have been obvious to a person of ordinary skill in the art ("POSITA") at the time of the alleged invention in view of *Hariki* (Ex. 1005) and *Harris* (Ex. 1006) (Ground I);
- Claims 1-20, 31 and 34 of the '865 patent would have been obvious to a POSITA at the time of the alleged invention in view of *Hariki* (Ex. 1005), *Harris* (Ex. 1006), and *Dees* (Ex. 1007) (Ground II);

33. This Declaration, including the exhibits cited herein, sets forth my opinions on these topics. In short, it is my opinion that *Hariki* and *Harris* render obvious claims 1-20, 31 and 34 of the '865 patent, and that *Hariki*, *Harris* and *Dees* also render obvious claims 1-20, 31 and 34 of the '865 patent.

IV. MATERIALS REVIEWED AND CONSIDERED

34. The materials I considered and relied upon in preparing my Declaration and forming my opinions include all exhibits cited to in this Declaration, including the '865 patent, the '865 file history, and the relevant prior art, as I discuss below.

35. I also have relied on my academic and professional experience in reaching the opinions expressed in this Declaration.

V. LEVEL OF ORDINARY SKILL IN THE ART

36. In rendering the opinions set forth in this Declaration, I was asked to consider the patent claims and the prior art through the eyes of a POSITA. The “art” is the field of technology to which a patent is related. I understand that the purpose of using the viewpoint of a POSITA is for objectivity.

37. I considered factors such as the educational level and years of experience of those working in the pertinent art; the types of problems encountered in the art; the teachings of the prior art; patents and publications of other persons or companies; and the sophistication of the technology. I understand that a POSITA is not a specific real individual, but rather a hypothetical individual having the qualities reflected by the factors discussed above.

38. Taking these factors into consideration, it is my opinion that a POSITA as of the time of the alleged invention of the '865 patent would have had a bachelor's degree in electrical or computer engineering, or a closely related scientific field such as computer science, and two to three years of work experience with remote page display rendering and/or remote server applications. Alternatively, any lack of work experience could be remedied with additional education (e.g., a Ph.D.) concentrating on multimedia content transmission and/or remote server applications, and likewise, a lack of education can be remedied with additional work experience in multimedia content transmission and/or remote server applications (e.g., 4-5 years).

VI. STATEMENT OF LEGAL PRINCIPLES**A. Claim Construction**

39. I understand that terms appearing in the patent claims are to be interpreted according to their “ordinary and customary meaning” in an IPR proceeding. In determining the ordinary and customary meaning, the words of a claim are first given their plain meaning as they would have been understood by a POSITA at the time of the alleged invention, in light of the specification and file history. I understand that treatises and dictionaries may be consulted, albeit under limited circumstances, to determine the meaning attributed by a POSITA at the time of the alleged invention. I have followed this approach in my analysis and have applied the ordinary and customary meaning of those terms throughout my analysis in this Declaration.

40. I understand that the words of the claims should be interpreted as they would have been understood by a person of ordinary skill in the art at the time the alleged invention was made (not today).

B. Anticipation

41. I understand that if each and every element of a claim is disclosed in a single prior art reference, then the claimed invention is anticipated and not patentable under pre-AIA 35 U.S.C. § 102. In order for the invention to be anticipated, each element of the claimed invention must be described or embodied, either expressly or

inherently, in a single prior art reference. I also understand that a reference inherently discloses a claim limitation when that claim limitation is necessarily present in the reference.

C. Obviousness

42. I have been informed that a patent claim is invalid as “obvious” under pre-AIA 35 U.S.C. § 103 in light of one or more prior art references if it would have been obvious to one of ordinary skill in the art, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so-called “secondary considerations” of non-obviousness, which include: (i) “long felt need” for the claimed invention, (ii) commercial success attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) “copying” of the claimed invention by others.

43. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason to modify the single prior art reference, or combine two or more references, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference or references themselves, the knowledge of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole. I have been

informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

VII. TECHNOLOGY BACKGROUND AND DISCUSSION OF THE STATE OF THE ART

44. It is my opinion that all of the elements of claims 1-20, 31, 34 were already known in the prior art before the priority date of the '865 patent. Accordingly, I conclude that there is nothing novel or non-obvious about the alleged invention disclosed in the '865 patent.

45. The '865 Patent is directed to well-known methods and systems for “rendering applications on a wireless device.” Ex-1001, 1:15-17.

46. To provide background for the element-by-element analysis of the claims to follow, I present an overview of some of the relevant state of the art existing as of the time of the alleged invention relating to the use of rendering technologies, wireless devices, and client/server architectures. As I will describe below, all of these technologies and techniques were well-known to those of ordinary skill in the art at the time of the alleged invention claimed in the '865 patent.

A. Graphics, Rendering, and Graphical User Interfaces

47. In computing, whether mobile or desktop, a user interface is the boundary where interactions between humans and computers occur. An early type

of user interface is the command-line interface, where the user enters text commands to the computer via the keyboard and the computer executes those commands. The command-line interface was simple and efficient, but it suffered from several drawbacks. For instance, the interface was difficult for users to learn, because to operate the computer, they first had to learn all the commands. Generally, commands in command-line interfaces were not expressed in natural language, but rather arbitrary abbreviations and acronyms that users had to memorize. For example, in the early Microsoft DOS operation system, text commands included “cd,” “dir,” and “chkdsk.” This resulted in a user interface that was not very intuitive – a novice user could not learn the user interface by simply experimenting with the interface.

48. An advancement to the command-line interface was the Graphical User Interface (“GUI”). GUIs offer visual interfaces on display screens that are designed for simplicity and ease of learning. A common and well-known example of a GUI uses the “desktop metaphor,” where files and applications of the computer system are represented as graphical icons on a virtual desktop shown on the computer screen, and where the user interacts with those icons using a pointing device such as a mouse cursor. Windows 3.1, released by Microsoft in 1992⁸, was one example of such a GUI:

⁸ <https://www.computerhistory.org/tdih/april/6/>

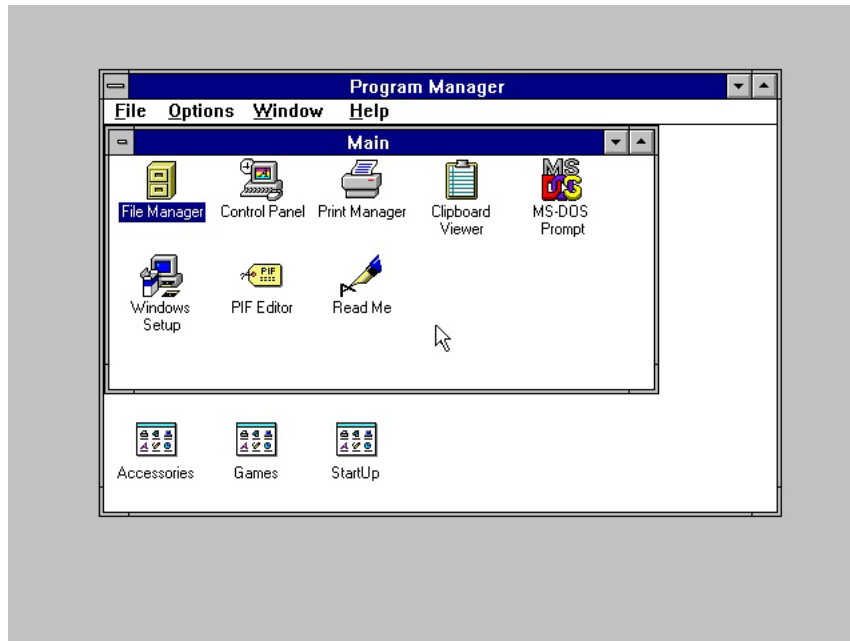
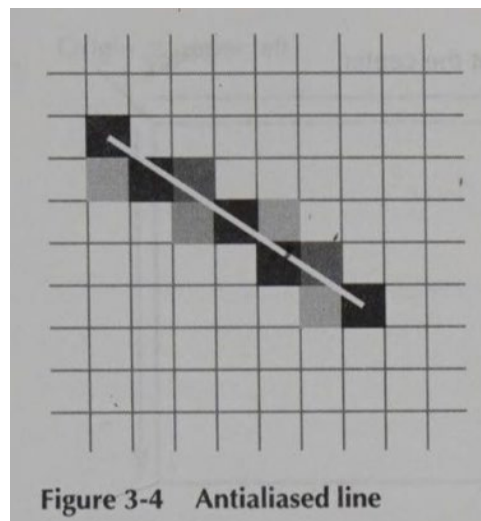


Figure A. Screenshot of Microsoft Windows 3.1 taken by Benjamin B. Bederson on March 28, 2023 using an emulator running in VMWare Fusion.

49. One main goal of GUIs was to make the user interface more intuitive and easy to learn by implementing concepts that are analogues to the real world – dragging an icon to move across the computer screen “desktop,” for example, resembles the act of moving an object in the real world that a user is already familiar with.

50. Part of the process for generating GUIs involves “rendering” the graphics – that is, deciding which specific color each pixel on the screen should be so as to form the visual elements of the interface. This rendering technology has been well-known for decades. For example, a 1998 book titled “Developing User

Interfaces”⁹ describes many of the technical elements of this rendering. For example, on page 51, it discusses “the basic 2D primitives that are required to present information to the users so they can interactively manipulate it.” *Id.*, p. 51. It covers topics such as the coordinate systems of 2D computer graphics, how to represent shapes such as lines with pixels, how to manage the drawing of text at different sizes with different fonts, etc. For example, the figure below shows how to represent a line segment with pixels on a display using a technique called “antialiasing.” Here, every pixel that overlaps the line segment is filled in with some level of gray. The more of the pixel that overlaps the line segment is filled in with some level of gray. The more of the pixel that overlaps the line, the darker the pixels.



Olsen, FIG. 3-8

⁹ Dan R. Olsen. 1998. Developing user interfaces. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

51. I taught an undergraduate course on the development of GUIs myself in 2005 and used *Olsen* as the textbook for the course. And I implemented most of these rendering techniques myself in the development of the Zoomable User Interface systems Pad++, Jazz, and Piccolo that I described in the Qualifications Section II.

52. By 2007, such rendering technologies were fairly standardized and built in to essentially every computer operating system that supported GUIs, including desktops and mobile devices. While many systems rendered the application GUIs on the same device that executed the application, there were also other well-known architectures that rendered GUIs on remote devices (e.g., “clients”). In this case, the application ran on other computers and sent the graphics commands to a different computer to be rendered. I describe some of these architectures in the section below on Client Server Architectures.

B. Wireless Devices

53. Mobile phones, a common type of wireless device, have been around since the 1980’s. Generally recognized as the first commercial mobile phone, the Motorola DynaTAC 8000X, using what is referred to as First Generation technology or 1G, was introduced by Motorola in 1983. These phones were the size of a briefcase and offered a talk time of about 30 minutes.

54. During the 1990s, great improvements were made in mobile phone technology and phones using Second Generation, or 2G technology, were introduced. The new technology made mobile phones smaller and more energy-efficient. Their small size and relatively long battery life contributed significantly to their popularity with consumers. In the 2000's, mobile phones using Third Generation technology, or 3G, were introduced. This technology was not only capable of transferring voice data (such as a phone call) but was also capable of transferring other types of data, including emails and instant messages. By 2000, wireless phones included touch screens and increased graphics capacity. In 2003, the PalmOne Treo 600 and BlackBerry 7210 included color screens.¹⁰ PDAs (Personal Digital Assistants) were another common kind of wireless device available in the early 2000s.

C. Client-Server Architectures

55. Computer systems have been architected for decades with a wide range of structures. Historically, computers were big and expensive and they provided “time shared” access to multiple terminals. Sometimes those terminals (a.k.a. “clients”) were “dumb” or “thin”, and provided primarily input and output capabilities while most of the computation was provided at the “server”. Other times,

¹⁰<https://www.webdesignerdepot.com/2009/05/the-evolution-of-cell-phone-design-between-1983-2009/>

those clients had greater computation capability, and in the early 1980s with the introduction of “personal computers”, those clients (while still often connected to servers) had sufficient computational capabilities to run independently.

56. The relationship between “clients” and “servers” is so foundational to the history of computers that it is often used to define the nature of the computing architecture as well as the user experience. The state of the client-side computing was described in a 1992 article¹¹ in Communications of the ACM – a broad interest magazine of the main professional international computer society. *Sinha* characterized some of the key elements of client-server computing, including that the client typically communicates with the server, transmitting queries or commands, performs data analysis on query or command results from the server and presents the GUI to the user. On the other side, servers typically provide a service to the client, responding to the queries or commands – and do so in a way that hides much of the details of its implementation so that clients with various kinds of software and hardware can work with it.

57. *Sinha* also explains that as far back as 1992, it was common to have the GUI of an application displayed on a computer remote from the application. It describes the X Window System that commonly ran on Unix environments which

¹¹ Alok Sinha. 1992. Client-server computing. Communications of the ACM, 35, 7 (July 1992), 77–98. DOI:<https://doi.org/10.1145/129902.129908> (“*Sinha*”)

“provides network transparency so that an application displaying a chart on a screen could be running on a local machine or a remote machine thousands of miles away.”

Id., p. 91.

58. One particularly important set of clients and servers communicate over the Internet in what has come to be called the World Wide Web (WWW). Tim Berners-Lee initially developed the WWW in 1990. While Internet-based communications had been around for some time at that point, the WWW simplified things by combining a few basic technologies including web servers, web browsers, HTTP, and HTML. The Internet was an enabling technology – allowing computers to communicate arbitrary information between any location in the world.¹²

59. Web servers are computers on the Internet that run software to listen to web “requests” and respond to those requests – typically by returning (i.e., serving) a document that was requested. Subject to hardware resource limitations, a web server may serve any number of web pages. Static web pages may be stored on a server’s hard drives and processed in memory. Server-generated web pages, such as those generated in response to a user’s search request, are generated and processed in memory. *See Berners-Lee* at 76-78. Web servers used the Hypertext Transfer

¹² Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret. 1994. The World-Wide Web. Communications of the ACM 37, 8 (August 1994), 76-82. DOI=<http://dx.doi.org/10.1145/179606.179671> (“*Berners-Lee*”)

Protocol (HTTP), a simple request/response network protocol, which allows an HTTP connection to be made from a client to a host.

60. Web browsers are client applications that provide a user interface for displaying data fetched from web servers. Somewhat like a word processing program that can open a document and render it, a web browser can open a special kind of document written using Hypertext Markup Language (HTML) and render it. However, web browsers have additional features including that they can fetch those documents across the Internet from web servers using the HTTP protocol.

61. Web browser requests are made to specific servers that are identified by Uniform Resource Locators (“URLs”). HTML documents can include references to images or other data that are specified via URLs and are fetched from specific web servers via those URLs. The browser then displays those separately retrieved images integrated with the textual document. Web browsers also can interpret the HTML documents they display to recognize hyperlinks that, when clicked on by a user, fetch the document or data referenced by another URL that is specified by the HTML’s hyperlink. If the link points to a document, the newly fetched HTML document is then displayed, typically replacing the previously displayed document. As *Berners-Lee* described in 1994, these URLs identify a single specific server,

offering a “Universal” way of locating and accessing for any network communications protocol, including HTTP.¹³

62. HTML is a textual document format for describing the visual presentation of documents, including embedded objects such as images and forms as well as links to other documents. It consists of plain text that will be displayed as such along with “tags” which surround portions of text and are used to modify how that text is displayed. Tags are formed by a word surrounded by a pair of angle brackets with a matching closing tag that is prefixed with a forward slash. For example, the following is a simple HTML snippet: “Here is some bold text.”¹⁴ When this HTML is interpreted by a web browser, it would be displayed as following:

Here is some **bold text**.

63. HTML can include links to other documents. It specifies those links with an “anchor” tag (i.e., <A ...> and a closing tag) along with URLs to indicate the location of the other document. URLs are created in text with a specific structured format including several elements such as the address (often represented by a domain name) of a web server the other document is on, and the location (or

¹³ See *Berners-Lee* at 76-78.

¹⁴ Eric Kasten, HTML: A Gentle Introduction, *Linux Journal* (July 1995), available at <https://www.linuxjournal.com/article/1081>

“path”) of that document on the web server. An example of an anchor tag follows where the “HREF” parameter specifies the URL of the document to be fetched and displayed when the link is clicked:¹⁵

```
<A HREF=“http://remote.host.name/docs/foo.html”>Foodocument</A>
```

64. As shown in these examples, HTML provides a syntax that is generic and can be used to define applications in a way that can work on client computer. In general, web sites are collections of HTML-based web pages that are stored or generated on a web server that is typically accessed by a single domain address. In order for a user to view those web sites, they would direct their web browser to load the web page located at a specific URL. At that point, the web browser would communicate with the web server specified by the URL via the World Wide Web to fetch the desired web page and would then render the page for display.¹⁶

65. Web pages supported multimedia and rich interactive content well before 2007. A key part of HTML is that it allowed the ability to download separately defined images or icons from web servers and embed them in a web page. I did this myself in the web browser I implemented in 1994 that I described in the Qualifications Section II.

¹⁵ See *Kasten* at 3

¹⁶ See *Kasten* at 1-3.

D. Overview of CSS

66. From the beginning of HTML in the 1990s, it was always intended to separate the document structure from its layout and the details of its visual presentation. Although HTML did include various ways to control that presentation, there were early efforts to provide ways to separate that control. The primary technology developed – during the same period that HTML was being developed in the 1990s – was Cascading Style Sheets (CSS)¹⁷. As described in a 2004 book¹⁸, CSS could be embedded into HTML or put in separate files and linked to from an HTML file. Colors, fonts, sizes, etc. could be applied to specific content, or entire categories such as headers – making it efficient to control the overall style of a document. For example, in HTML, the content of a header could be defined with something like the following. *Id.* at chapter 1. `<h1>Leaping Above The Water</h1>`.

67. And then in CSS, you could define the font and color of all `<h1>` headers – which would apply to every such header in the document. For example, the CSS for this might look like the following. *Id.*

68. `h1 {color: maroon; font: italic 2em Times, serif; text-decoration: underline; background: yellow;}`

¹⁷ <https://www.w3.org/Style/CSS20/history.html>

¹⁸ Eric A. Meyer (2004). Cascading Style Sheets: The Definitive Guide, Second Edition. O'Reilly Media, Inc. <https://learning.oreilly.com/library/view/cascading-style-sheets/0596005253/>

69. In addition to controlling the visual style of HTML, CSS could also be used to control the layout of the HTML elements. There were many ways to do this including tables and floating elements (that hugged an edge of the window) as well as absolute positioning. In regular HTML layout, the content generally flows from left to right and top to bottom. But CSS also gives the ability to break that flow and place elements in a particular position using the “absolute” “position” command. *Id.* at chapter 10.

E. Overview of JavaScript and the Document Object Model.

70. While HTML was originally created to allow the viewing and navigation of documents, it quickly became apparent that HTML could be used for additional purposes, including modification of and interaction with content. JavaScript was designed at Netscape in 1995 as one way to provide that functionality¹⁹. And over the next few years, a competitive marketplace largely driven by Netscape and Microsoft resulted in multiple variations of JavaScript and the Document Object Model (“DOM”) encouraged this interaction. And as described above, I created my own browser using the TCL scripting language to provide for interaction within a web page.

¹⁹ W3C Document Object Model (DOM) Level 1 Specification Version 1.0 (1998) at 2. <https://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/DOM.pdf>

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

72. This specification defines the Document Object Model Level 1, a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents. The Document Object Model provides a standard set of objects for representing HTML and XML documents, a standard model of how these objects can be combined, and a standard interface for accessing and manipulating them²⁰.

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

With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model, with a few exceptions . . .²¹

²⁰ W3C Document Object Model (DOM) Level 1 Specification Version 1.0 (1998) at 2. <https://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/DOM.pdf>

²¹ W3C Document Object Model (DOM) Level 1 Specification Version 1.0 (1998) at 10.

74. In this paradigm, the combination of an appropriately written HTML document and appropriately programmed JavaScript (which could be included within the HTML document or separately downloaded as a result of certain content in the HTML document) allowed users to interact with the HTML document itself. The code in JavaScript files (or files of any other similarly capable programming language) was “interpreted” by the web browser. This was described in 1997 by ECMA International, a European focused international computing standards body. At that point, JavaScript had been developed by competing organizations including Netscape and Microsoft, and ECMA moved to standardize it in what they termed “ECMAScript.” In the 1997 first edition of this standard,²² ECMA explained that ECMAScript was “not intended to be computationally self-sufficient,” but rather was intended to be run within browsers, which “provide a host environment of objects and facilities which completes the capabilities of the scripting language.” ECMAScript Standard at 1-2.

75. Despite the W3C’s effort to standardize the DOM, Netscape’s “Navigator” and Microsoft’s “Internet Explorer” web browsers provided competing versions of the DOM that were similar, but not identical. This resulted in web developers having to make their web pages work with multiple versions of web

²²https://www.ecma-international.org/wp-content/uploads/ECMA262_1st_edition_june_1997.pdf (“ECMAScript Standard”)

browsers and DOMs, as described in a 2001 article (which also demonstrated that the DOM and JavaScript were very well used by that point).²³

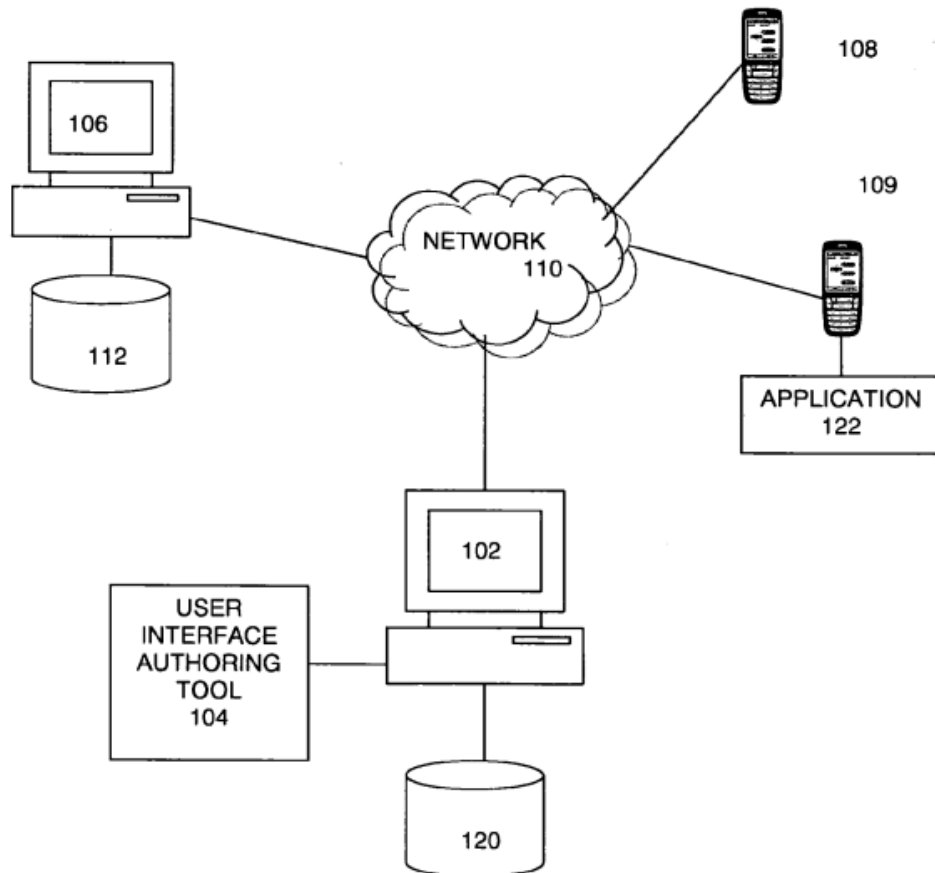
F. Overview of U.S. Patent Application Publication No. 2007/0150617 to Hariki, filed on July 25, 2006 (“*Hariki*”).

76. The inventors of *Hariki* recognized that “[c]ustomization features, referred to as “UI skin packages,” are often developed and provided by third party vendors.” *Hariki*, ¶7. Recognizing this, the inventors realized the need for “a mobile device configuration system that allows modification of mobile device user interfaces or application programs without modification of the application programs themselves.” *Hariki*, ¶9.

77. *Hariki* is thus directed to “a mobile device incorporating a resource application program interface for downloading user interface components for a resident application.” *Hariki*, Abstract. *Hariki* provides that the “[e]mbodiments of the inventions related generally to communication networks and more specifically, to a user interface generation system for mobile communication devices.” *Id.*, ¶3. These various embodiments may be implemented in a client-server arrangement in a network “that incorporate[s] a telecommunications or similar wireless network system.” *Id.*, ¶18. Figure 1 illustrates this network system. The system comprises

²³ Peter-Paul Koch, “Digital Web Magazine - The Document Object Model: an Introduction”, May 14, 2001, https://www.digitalweb.com/articles/the_document_object_model/

mobile devices 108 and 109, such as cellular phones “configured to communicate with one or more client and/or server computers that are directly or indirectly coupled to network 110.” *Id.*, ¶20. *Hariki* explains that “a server computer 102 operated by the service provider or a third party vendor is a content provider server computer that provides content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile device 108.” *Id.*, ¶21.

**FIGURE 1**

Hariki, FIG. 1

78. *Hariki* discloses a wide range of embodiments of the network system 110. The network 110 “may also include one or more Wide Area Networks (WAN). Local Area Networks (LAN), or any combination thereof. The server computer can be a wireless service provider or Internet Service Provider or combination of both.” *Id* at 20. Similarly, *Hariki* discloses a large range of mobile devices, “computers that are directly or indirectly coupled to network 110. The mobile devices 108 and 109 can represent devices made by different manufacturers, different models of devices made by the same manufacturer or any type of devices that have different user interface elements from one another.” *Id*.

79. The server can execute programs, such as a web server process, to send HTML pages to the mobile devices. *Id.*, ¶21. The server can also “generate a customized user interface for a plurality of different makes and types of mobile devices” that “allow a user to customize the ‘look and feel’ or application program environment of a device by altering display and/or sound output aspects of the device, such as backgrounds, title bars, buttons, alert sounds, and so on.” *Id.*, ¶24. This customized “look and feel” is delivered in a content package and is “specific to the type of device, manufacturer of the device, operating system, application programs, and other relevant information regarding the mobile device.” *Id.*, ¶28.

G. Overview of U.S. Patent Application Publication No. 2003/0023755 to Harris et al., filed on December 18, 2001 (“Harris”).

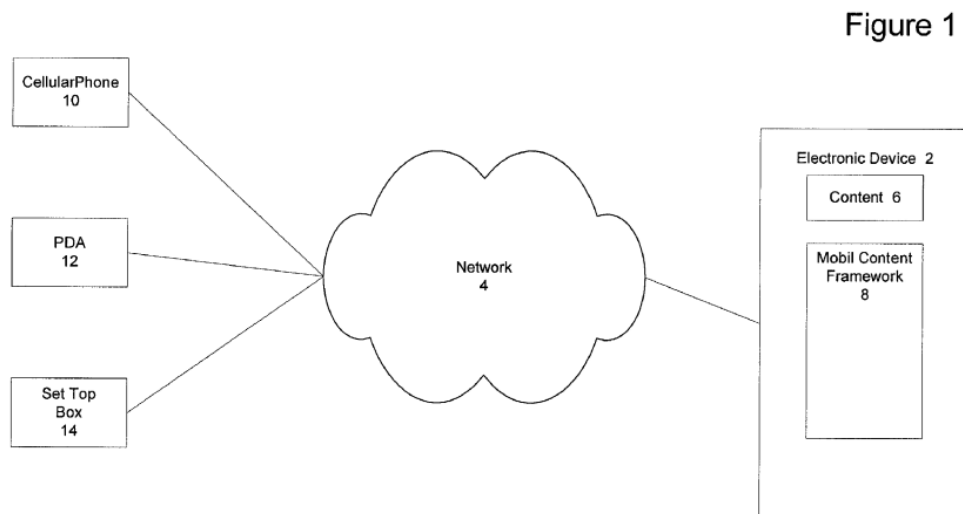
80. *Harris* is directed at “the delivery of content to mobile devices, and more particularly the delivery of the same content to multiple mobile devices using different device protocols. *Harris*, ¶2.

81. *Harris* discloses “a platform that enables a content developer to distribute uniform content to multiple types of requesting mobile devices.” *Harris*, Abstract. *Harris* explains that “[b]y authoring content in a generic markup language and using the technologies included in the Mobile Content Framework (MCF), the content developer is able to provide content to multiple types devices without providing different versions of the content.” *Id.* *Harris* further explains that “[c]ontent is abstracted in a manner specifically tailored to take into account the limited resources of certain devices such as mobile devices” and that “[c]ontent is generated specifically for each device, both from a display standpoint and a content navigation standpoint.” *Id.*, ¶7. Using the *Harris* platform, the “interface may be dynamically personalized to the taste of the individual.” *Id.* The content is first generated in a generic markup language and then translated “into a device appropriate language for a requesting mobile device.” *Id.*

82. The Mobile Content Framework (MCF) in *Harris* contains the generic markup language, “referred to as Wireless Abstract XML (hereafter WAX)” that enables the content developer to “to describe content at a more abstract level... the

greater level of abstraction enables the subsequent conversion of WAX into the languages used by the requesting mobile devices such as WML, HDML and HTML.” *Id.*

83. *Harris* discloses its environment with reference to Figure 1, reproduced below. The environment includes a server (electronic device 2) with content and an MCF. *Id.*, ¶15. The server is interfaced to a plurality of wireless devices through a network. *Id.*



Harris, FIG. 1

84. *Harris* further explains, with reference to Figure 3, below, that the server receives a request from a wireless device (step 60) and the device type is determined from the request (step 62). *Id.*, ¶24. The server determines device capabilities (step 64) and a session with an application is started to store application data (step 66). *Id.* *Harris* elaborates that the data can come from applications, such

as “web site, a static file, a Servlet or a Java Server Page.” *Id.*, ¶26. A wireless device independent template is generated in the WAX (XML) language (step 68) and subsequently translated to a device specific language in step 70. *Id.*, ¶24. After further processing, “customized content is then returned to requesting device (step 76).” *Id.*

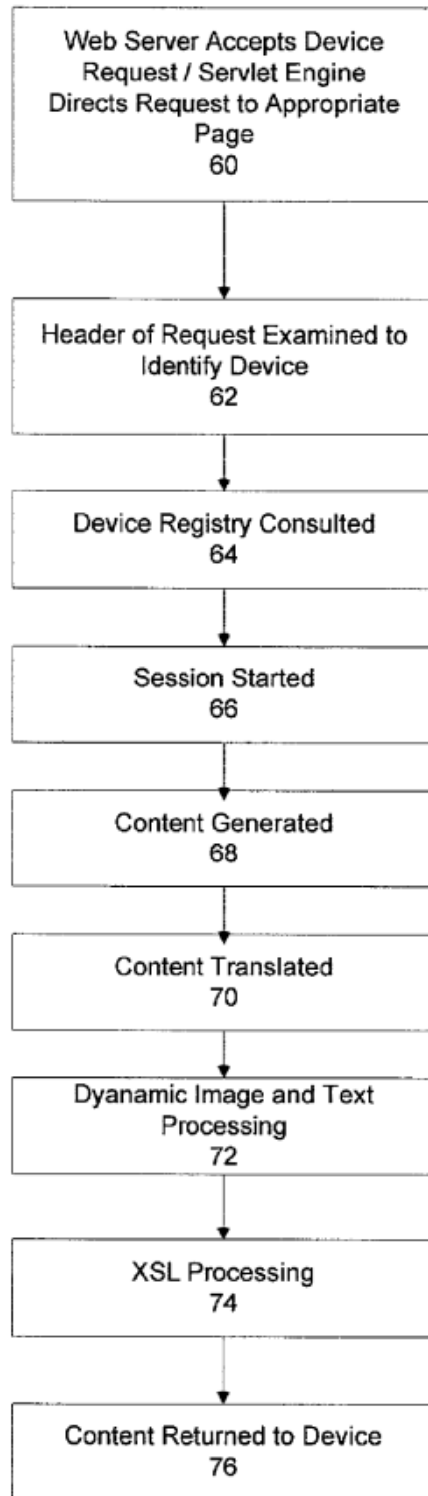


Figure 3

H. Overview of U.S. Patent Application Publication No. 2003/0137539 to Dees et al., filed on October 1, 2002 (“Dees”).

85. Prior art *Dees* is directed to “a method of styling a user interface for a device” (*Dees*, ¶1) in a client-server system (*id.*, ¶33).

86. *Dees* explains that “[t]he user interface module 420 [of a client device] activates the networking module 410 to cause it to download the first style sheet document 211 from the server 101.” *Dees*, ¶51. *Dees* further explains that “[h]aving downloaded the document 211, the networking module 410 feeds the document 211 to the user interface module 420. The user interface module 420 then uses the style information contained in the document 211 in generating the user interface.” *Id.*

87. Furthermore, *Dees* discloses that the “style sheet document [] can comprise style information in any language suitable for marking up style information, such as CSS or DSSSL. It is also possible to use XML to mark up style information.” *Id.*, ¶53.

88. Additionally, *Dees* discloses that “HyperText Markup Language HTML also provides [] ways to provide a user interface definition.” *Id.*, ¶37. *Dees* further states that “[u]sing a language such as CSS or DSSSL, style properties such as the font or the color of a button or the position of a checkbox on the screen can easily be defined.” *Id.*, ¶3.

89. Figure 1 of *Dees*, shown below, “schematically shows a system 100 according to the invention.” *Id.*, ¶33. The system comprises server 101 connected to

a network 110, along with laptop computer 120, a desktop computer 125, a mobile phone 130 and a handheld computer 135. *Id. Dees* explains that the “[t]he mobile phone 130 and the handheld computer 135 have a wireless connection to the network 110.” *Id.*, ¶34.

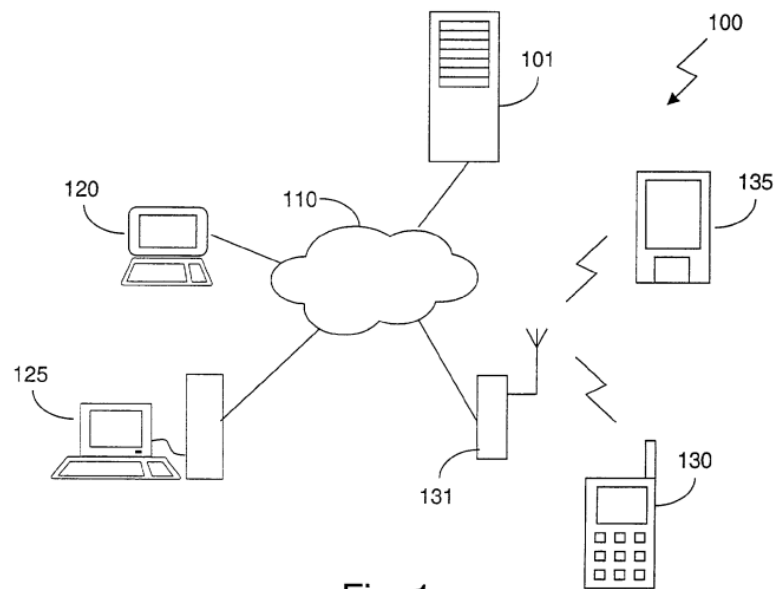


Fig. 1

90. Figure 4 of *Dees*, shown below, “schematically shows a client device 400 in more detail,” including “a networking module 410 which obtains the user interface definition document 210 from the server 101, via the network 110. The document 210 is then fed to a user interface module 420, which processes the document 210 in order to generate a user interface. A rendering module 450 then presents the generated user interface to a user of the client device 400 using output

means such as a display 452 or a speaker 451. Style information is applied to this user interface, as will become apparent below.” *Id.*, ¶¶48-49.

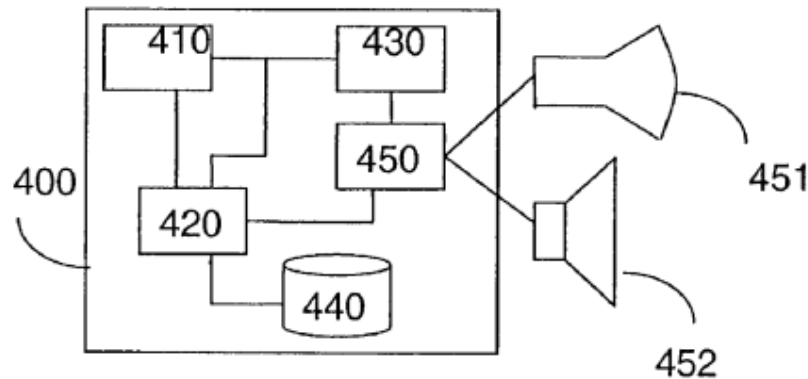


Fig. 4

I. Other Evidence Regarding the State of the Art

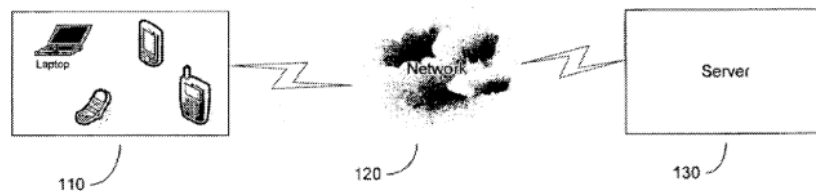
91. Exhibits 1009-1030 are additional prior art references that provide more information regarding the state of the art, level of ordinary skill in the art, common knowledge in the art, and/or common sense in the art. These prior art references are pertinent to understanding the background of the '865 patent and my invalidity analysis.

VIII. SUMMARY OF THE '865 PATENT

A. The '865 Patent Specification

92. The '865 patent is directed to “server method and system for rendering content on a wireless device” ’865*Pat*, Abstract. The specification explains that “a wireless device . . . is operable to communicate with a server and receive a series of low level basic commands that layout content, position, etc. for rendering application

content on the wireless device.” *Id.*, 2:46-50. Figure 1A is described as “an exemplary communication system 100A in accordance with one embodiment of the present invention including wireless clients and a remote server.” *Id.*, 5:65-6:1. The system includes “one or more wireless devices 110 are coupled to a server 130 through a network 120.” *Id.*, 6:1-3.

100A*'865Pat*, FIG. 1.

93. Figure 2A and 2B show “communication sequence[s] 200A and 200B between a wireless device 210 and a remote server 230.” *Id.*, 11:9-12. “At step 212 the client 210 sends a message to the server 230 identifying the wireless device type and its capabilities along with a request to access an application on the server.” *Id.*, 11:12-15. The specification explains that “the application is generic and independent of the wireless device type” and “the wireless device type and its capabilities may include” a multitude of things, such as “the brand, the model and the version number” or device capabilities. *Id.*, 11:15-16, 22-31.

94. The specification makes clear that enumerated device types, however, “should not be construed as limiting the scope of the present invention.” *Id.*, 11:31-33. “In response to the message 212, the server 230 identifies the requested application and its corresponding custom configuration to be used by the client 210” by “send[ing] a message to the client 210 identifying the corresponding custom configuration.” *Id.*, 11:59-63. If the wireless device does not have the identified custom configuration, it may request “to download the identified information (e.g., custom configuration) from the server 230” at step 216. *Id.*, 12:17-21. After receiving the request, the server sends the custom configuration to the wireless device. *Id.*, 12:62-64.

200A

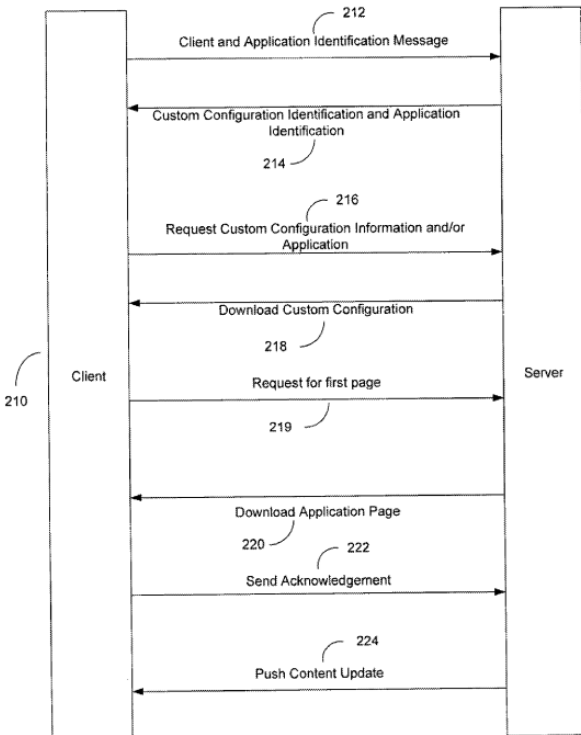


FIGURE 2A

200B

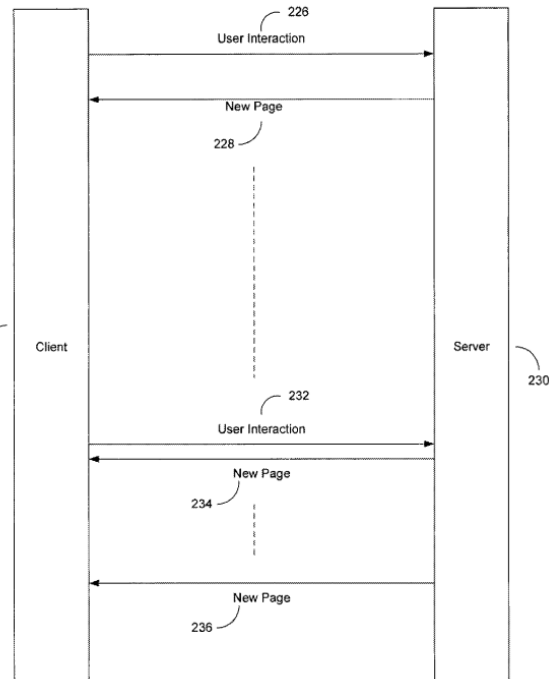
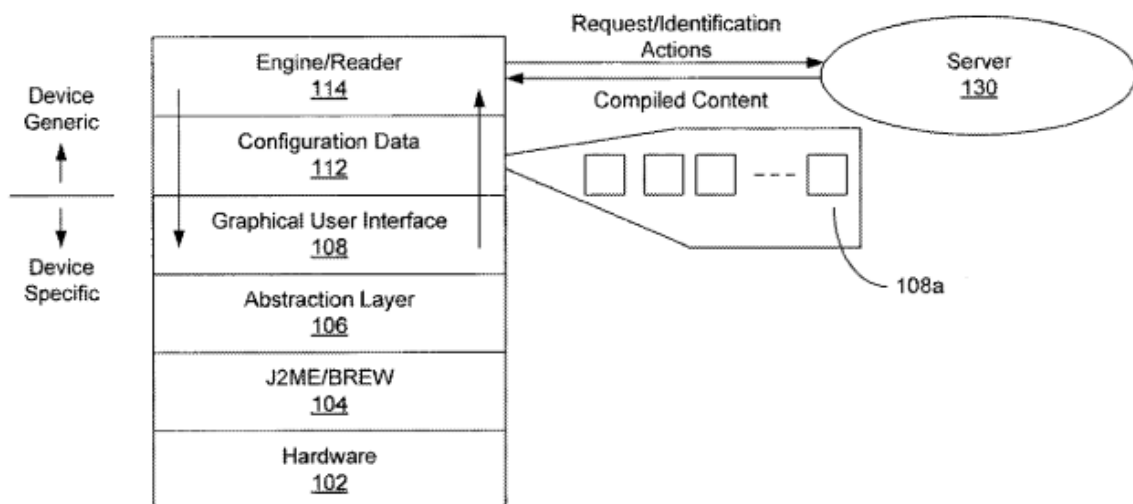


FIGURE 2B

'865Pat, FIGs. 2A & 2B

95. This custom configuration “customizes the look and feel of the content displayed on the wireless device from execution of the requested application.” *Id.*, 18:41-43. The specification explains that a graphical user interface (“GUI” shown in Figure 1B below) in the wireless device includes rendering blocks (*id.*, 7:21-34) that use the custom configuration “to customize the appearance of the requested application and represents programming and stored content of the graphical user interface 108” (*id.*, 7:59-62). “In other words, configuration data may be a set of low level instructions for preprogramming a plurality of rendering blocks 108a of the graphical user interface to operate and render data (e.g., “look”) a certain way.” *Id.*, 7:62-66.

100B



'865Pat, FIG. 1B

96. The specification further explains (and is shown as step 220 in Figure 2A above) that the wireless device “may receive a compiled content from the server 130 that includes a series of basic commands for rendering the requested application.” *Id.*, 7:42-45, 13:11-25. The GUI in the wireless device receives this compiled content and “render the page of the application based on the received basic commands and the customized preprogrammed plurality of rendering blocks.” *Id.*, 10:64-11:5. The user may interact with the page, causing a new (second) page to be sent from the server, as shown above at steps 226 and 228 of Figure 2B. *Id.*, 14:17-31, FIG. 2B.

B. The '865 Patent File History (Ex. 1004)

97. It is my understanding that the '865 patent issued on January 24, 2012 from U.S. Patent Application 11/888,799, filed August 1, 2007. The application that led to the '865 patent was prosecuted under the pre-AIA statutory regime and included 41 claims. *'865PH*, 231.

98. The August 1, 2007 filing was accompanied with a non-publication request. *Id.*, 81. Applicant later rescinded the non-publication request on July 31, 2008 without providing any reasons for the rescission. *Id.*, 110. The initial information disclosure statement (IDS) filed on August 1, 2007 only included one patent. *Id.*, 255. A subsequent IDS filed, almost two years later, on June 2, 2009

included only three patents and patent applications, and five non-patent literature documents. *Id.*, 249. A non-final rejection was mailed on June 25, 2010, in which the Examiner rejected all 41 claims using a single published patent application. *Id.*, 231. Additionally, the Examiner listed just four references in the notice of references cited. *Id.*, 242. In response, applicant heavily amended some claims, while canceling others on December 21, 2010. *Id.*, 264. The amended claims were again rejected in a final rejection by the Examiner on lack of written description, enablement, and prior art grounds (maintained from the earlier non-final rejection) on February 7, 2011. *Id.*, 532. In the final rejection, the Examiner noted that they were unable to find support for multiple limitations, including the newly added limitation of “said plurality of content items using both said screen description and said custom configuration.” *Id.*, 533.

99. To overcome the rejection, applicant amended the claims to simply replace “screen description” with “page description” in a request for continued examination on August 8, 2011. *Id.*, 549. Without providing any reasons for allowance, the Examiner allowed all outstanding claims on September 7, 2011. *Id.*, 568.

C. Challenged Claims

100. The '865 patent has three independent claims, claims 1, 12, and 21, and thirty one dependent claims, claims 2-11, 13-20, and 22-34. I have been asked to opine on the validity of claims 1-20, 31, and 34, which are listed below:

Claim 1	
[1pre]	A server implemented method for processing data for a wireless device, comprising:
1[a]	receiving from the wireless device a request for an application program, said request including an indication of a type of the wireless device;
1[b]	executing, in response to receiving said request, said application program to generate a wireless device generic template including a plurality of content items;
[1c]	sending a custom configuration to the wireless device, said custom configuration being specific to said application program;
[1d]	generating a page description based on said wireless device generic template and a capability of the wireless device,
[1e]	said page description having at least one discrete low level rendering command that is within said rendering capability of said wireless device but that is of a syntax that is wireless device generic; and
[1f]	sending said page description to the wireless device such that the wireless device is capable of presenting at least one content item from said plurality of content items using both said page description and said custom configuration.
Claim 2	
[2]	The method of claim 1, wherein said wireless device generic template includes relative screen location and display object size

	information and is independent of screen dimensions of the wireless device.
Claim 3	
[3]	The method of claim 1, wherein said wireless device generic template is substantially compliant with extensible markup language (XML) syntax.
Claim 4	
[4]	The method of claim 1, wherein said at least one discrete low level rendering command of said page description is operable to be executed by a plurality of rendering blocks of a graphical user interface of the wireless device.
Claim 5	
[5]	The method of claim 4, wherein said at least one discrete low level rendering command includes a physical screen position of a display element of said at least one content item.
Claim 6	
[6]	The method of claim 4, wherein said at least one discrete low level rendering command includes: a specification of a location and a size of a screen area; and an identification of a rendering block from said plurality of rendering blocks of said graphical user interface to operate within said screen area.
Claim 7	
[7]	The method of claim 6, wherein said at least one discrete low level rendering command includes parameters associated with said rendering block from said plurality of rendering blocks.
Claim 8	
[8]	The method of claim 1, wherein said page description includes at least one display rendering command and at least one audio rendering command.

Claim 9	
[9]	The method of claim 1, wherein said application program is wireless device generic and said wireless device generic template is a wireless device generic page description template.
Claim 10	
[10]	The method of claim 1, wherein the wireless device is operable to configure a plurality of rendering blocks of a graphical user interface based on said custom configuration.
Claim 11	
[11a]	The method of claim 1, wherein said server stores a plurality of device profiles, said method further comprises:
[11b]	identifying a device profile from said plurality of device profiles associated with the wireless device based on said type of said wireless device.
Claim 12	
[12pre]	A non-transitory processor-readable medium storing code representing instructions to cause a processor to:
[12a]	receive a request for an application program from a wireless device, said request including an indication of a type of the wireless device;
[12b]	execute said application program to generate a wireless device generic template including a plurality of content items;
[12c]	request, using said indication of said type of the wireless device, a device profile from a library of device profiles,
[12d]	said device profile providing an indication of a rendering capability of the wireless device;
[12e]	eliminate, using said device profile, at least one content item from said plurality of content items from said wireless device generic template to define a device specific template associated with the wireless device,

[12f]	said at least one content item being unsupported by the wireless device;
[12g]	translate said device specific template into a page description having at least one discrete low level rendering command that is within said rendering capability of the wireless device but that is of a syntax that is wireless device generic; and
[12h]	send said page description to the wireless device.
Claim 13	
[13]	The non-transitory processor-readable medium of claim 12, wherein said wireless device generic template contains relative screen location and display object size information independent of screen dimensions of the wireless device.
Claim 14	
[14]	The non-transitory processor-readable medium of claim 12, wherein said wireless device generic template is substantially compliant with extensible markup language (XML) syntax.
Claim 15	
[15]	The non-transitory processor-readable medium of claim 12, wherein said at least one discrete low level rendering command of said page description is operable to be executed by a plurality of rendering blocks of a graphical user interface of the wireless device.
Claim 16	
[16]	The non-transitory processor-readable medium of claim 15, wherein said at least one discrete low level rendering command includes a physical screen position of a display element of a content item from the plurality of content items.
Claim 17	
[17a]	The non-transitory processor-readable medium of claim 15, wherein said at least one discrete low level rendering command includes:

	a specification of a location and a size of a screen area; and
[17b]	an identification of a rendering block from said plurality of rendering blocks of said graphical user interface.
Claim 18	
[18]	The non-transitory processor-readable medium of claim 17, wherein said at least one discrete low level rendering command includes at least one parameter associated with said rendering block from said plurality of rendering blocks.
Claim 19	
[19]	The non-transitory processor-readable medium of claim 12, wherein said page description includes at least one display rendering command and at least one audio rendering command.
Claim 20	
[20a]	The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause said processor to: send, in response to said request, a custom configuration to said wireless device,
[20b]	the wireless device is operable to configure a plurality of rendering blocks of graphical user interface based on said custom configuration.
Claim 31	
[31]	The method of claim 1, wherein the sending the custom configuration to the wireless device is in response to receiving an indication from the wireless device that the custom configuration is not stored at the wireless device.
Claim 34	
[34a]	The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause the processor to: send a custom configuration to the wireless device, said custom configuration being specific to said application program,

[34b]	the wireless device operable to present a content item from said plurality of content items using said page description and said custom configuration.
-------	--

D. Claim Construction (37 C.F.R. § 42.104(b)(3))

101. I understand that in the litigation, *GoTV, Streaming, LLC, v. Netflix Inc.*, Case 2:22-cv-7556 (C.D. Cal.), neither Petitioner nor Patent Owner have proposed any constructions for any terms in the claims of the '865 patent. For the purposes of this proceeding, I have been asked to analyze the claims assuming the ordinary and customary meanings of the claim terms apply, as they would have been understood by a POSITA at the time of the alleged invention of the '865 patent.

102. I understand that there are no other litigations that involve or had involved the '865 patent, and none of the terms in the '865 patent have been construed.

IX. PRIORITY DATE OF THE '865 PATENT

103. I have been informed that a patent is entitled to the benefit of the filing date of an earlier filed application only if the disclosure of the earlier application provides support for the claims of the latter application as required by the “written description” requirement of 35 U.S.C. §112. I have been informed that for an earlier patent application to have an adequate written description of the invention it must show, within the four corners of the earlier application, that the inventors were in

possession of the invention at the time the earlier application was filed. I have also been informed that in order to satisfy the written description requirement, a POSITA, reading the original disclosure, must immediately discern the limitation at issue in the claims. I have been further informed that it is insufficient if the limitations are obvious variants of the disclosure in the specification.

104. I understand that the '865 patent was filed on August 1, 2007. The references I have used in Grounds 1 and 2 are prior art to the August 1st filing date.

X. APPLICATION OF THE PRIOR ART TO CLAIMS 1–20, 31, 34

A. Ground 1: *Hariki* in view of *Harris* renders obvious Claims 1-20, 31, 34

- 1. Claim 1 “A server implemented method for processing data for a wireless device, comprising: receiving from the wireless device a request for an application program, said request including an indication of a type of the wireless device; executing, in response to receiving said request, said application program to generate a wireless device generic template including a plurality of content items; sending a custom configuration to the wireless device, said custom configuration being specific to said application program; generating a page description based on said wireless device generic template and a capability of the wireless device, said page description having at least one discrete low level rendering command that is within said rendering capability of said wireless device but that is of a syntax that is wireless device generic; and sending said page description to the wireless device such that the wireless device is capable of presenting at least one content item from said plurality of content items using both said page description and said custom configuration.”**

- a. [1pre] A server implemented method for processing data for a wireless device, comprising:**

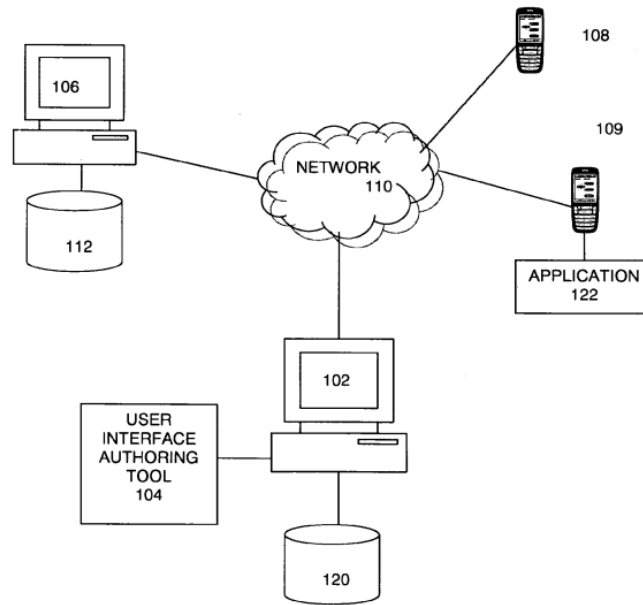
105. It is my opinion that *Hariki* in combination with Harris discloses claim 1.

106. Furthermore, it is my opinion that to the extent the preamble is limiting, *Hariki* discloses this feature.

107. For example, *Hariki* discloses, with reference to Figure 1 below, “server computer 102 . . . provides content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile device 108. . . . For example, server 102 can execute[] a web server process to serve web pages,”²⁴ such as HTML files over a network to the mobile device. *Hariki*, ¶21.

108. Furthermore, the server “can generate a customized user interface for a plurality of different makes and types of mobile devices.” *Hariki*, ¶24. *Hariki* explains that “the customized user interface is referred to as a ‘UI skin’” to “allow a user to customize the ‘look and feel’ or application program environment of a device by altering display and/or sound output aspects of the device.” *Id.*

²⁴ Unless otherwise stated, all emphases are added.

**FIGURE 1**

Hariki, FIG. 1.

109. Furthermore, *Hariki* discloses that mobile devices 108 and 109 may be cellular phones. *Id.*, ¶20. With reference to Figure 1 above, *Hariki* explains that “mobile devices 108 and 109 communicate with each other and other mobile devices over the **wireless portion of network 110.**” *Id.*; see also ¶22 (“The mobile devices 108 and 109 may be coupled to the network 110 over a wired connection, a **wireless connection** or any combination thereof.”).

110. It is therefore my opinion that the mobile devices are wireless devices.

111. *Hariki* further discloses “a **method** of generating and downloading UI skins.” *Hariki*, ¶36. For example, as shown in Figure 4 below, “the method begins in step 401 with selection and retrieval of the appropriate resource profile for the

device from the resource profiles” and concludes with “[t]he UI content package . . . downloaded to the target mobile device” in step 410. *Id.*, ¶36.

112. It is my opinion that *Hariki* explains that the UI content package (in other words, the UI skin) alters the display of the mobile device. *Hariki*, ¶24.

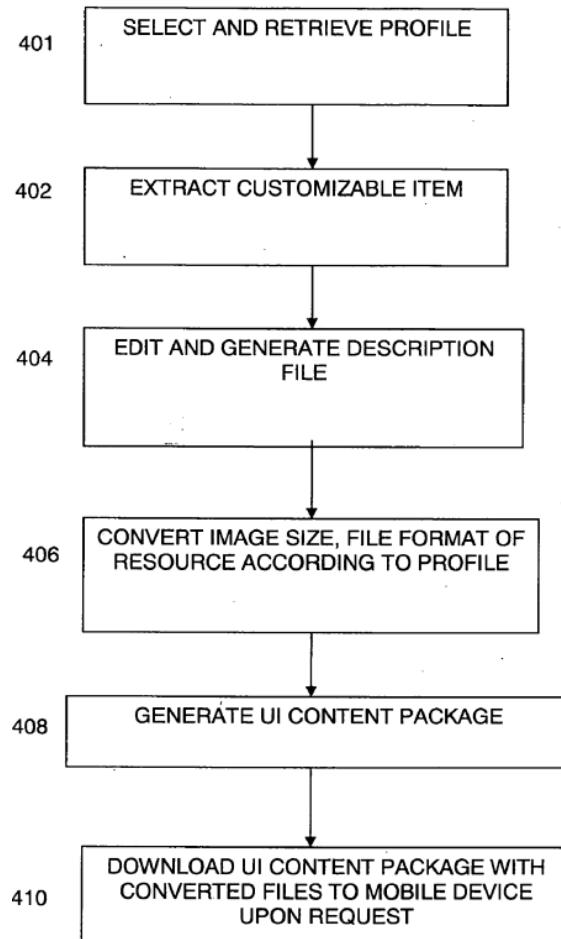


FIGURE 4

Hariki, FIG. 4.

113. It is, therefore, my opinion that *Hariki* discloses a method of generating UI skins and providing content data by a server (corresponding to the claimed

“server implemented method for processing data”) to a mobile device (corresponding to the claimed “for a wireless device”).

b. 1[a] receiving from the wireless device a request for an application program, said request including an indication of a type of the wireless device;

114. It is my opinion that *Hariki* discloses this feature.

115. *Hariki*, for example, discloses that a “server computer 102 operated by the service provider or a third party vendor is a content provider server computer that provides content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile device 108.” *Hariki*, ¶21.

116. Furthermore, *Hariki*’s server “can be a World-Wide Web (WWW) server that stores data in the form of web pages and transmits these pages as Hypertext Markup Language (HTML) files over the Internet 110 to the mobile devices 108 and 109.” *Hariki*, ¶21. The WWW server “executes a web server process to serve web pages.” *Id.* The mobile device “runs a web browser program to access the web pages served by server computer 102 and any other available content provider or supplemental server, such as computer 106.” *Id.*

117. It is my opinion that the web browser running on the mobile device sends a request to the web server process, which receives the request, to access the

corresponding HTML files (corresponding to the claimed “receiving from the wireless device a request for an application program”).

118. It is also my opinion that there is no distinction between “accessing” and “requesting” the web server process to receive the HTML files.

119. *Hariki* further discloses that “mobile devices 108 and 109 can represent devices made by different manufacturers, different models of devices made by the same manufacturer or any **type of devices** that have different user interface elements from one another.” *Hariki*, ¶20. *Hariki* explains “server computer 102 is operated by a content provider that can generate a customized user interface for a plurality of different makes and types of mobile devices” (*id.*, ¶24), and “[v]arious resource profiles and resource files for the different mobile devices can be stored in a database” (*id.*, ¶25). *Hariki*’s server “downloads the user interface to the appropriate mobile device upon request of the mobile device user.” *Id.*, ¶25.

120. Moreover, in order to download the appropriate interface, *Hariki* discloses “method begins in step 401 with selection and retrieval of the appropriate resource profile for the device.”²⁵ *Id.*, ¶36. The resource profile is selected “depending upon the device model selected by the content provider.” *Hariki*, ¶32.

²⁵ “Resource profiles are defined by device manufacturers for each model of mobile device.” *Hariki*, ¶19.

121. It is my opinion that to provide requested content to a mobile device, after checking stored device profiles, the server would have determined what type of mobile device is making the request from the information received from the mobile device.

122. It is also my opinion that in order for the server to determine what type of mobile device is accessing it, the device would have sent its model type or other identifying information to identify the type of mobile device. For example, the manufacturer model number would identify the type of device. Also, the size of the screen or capability to display color would have also identified the type of device.

123. It is my opinion, therefore, that *Hariki* discloses the mobile device identifying what type of device it is (corresponding to the claimed “including an indication of a type of the wireless device”).

124. *Harris*, like *Hariki*, discloses “a platform that enables a content developer to distribute uniform content to multiple types of requesting mobile devices.” *Harris*, Abstract.

125. As explained above in Section VII.B, *Harris* describes that a wireless device template independent of mobile device type is generated in the WAX (XML) language and subsequently translated to a device specific language. *Harris*, ¶24. After further processing, “customized content is then returned to requesting device (step 76).” *Id.* *Harris* explains that the customized content is in “a language the

requesting device can understand” (*id.*, ¶24), “such as WML, HDML, HTML, compact-HTML and Palm webClippings” (*id.*, ¶3). It is therefore my opinion that Harris discloses a detailed process of sending HTML files to a client mobile device where the HTML files are tailored to the particular mobile device.

126. It is also my opinion that the methods disclosed by *Harris* were compatible with and could be adapted and applied to *Hariki* because Harris’ platform is of the same general architecture as *Hariki*’s client-server system.

127. Therefore, it is my opinion that to the extent that Patent Owner contends that *Hariki* does not disclose that the mobile device sends a request to the server that indicated its device type, there is motivation to combine the client-server system of *Hariki* with the teachings of *Harris* directed to serving HTML pages tailored to a particular mobile device. In particular, it is my opinion that, to the extent that Patent Owner contends that while *Hariki* discloses sending HTML files to a mobile device, it does not expressly disclose how the HTML files are generated, a POSITA would have been motivated to find ways of generating and sending information from applications such as a web server process customized to a particular mobile device and looked to *Harris*.

128. It is my opinion that there is a reasonable expectation of success in implementing Harris’ content generation scheme with *Hariki*’s customization scheme. For example, Patent Owner could contend that while *Harris* discusses that

an “interface may be dynamically personalized to the taste of the individual” (*Harris*, ¶7), it does not explain how this is accomplished other than “a client API to interact directly with a personalization service” or “personalization engines” (*id.*, ¶24). *Hariki* fills this gap in *Harris* by explaining how “[r]esource profiles are defined by device manufacturers for each model of mobile device. The resource profiles specify the composition, format, and appearance of one or more user interface elements displayed or executed by the mobile device.” *Hariki*, ¶19.

129. It is also my opinion that *Hariki* further explains how to customize the user interface (“UI skin”) by “altering display and/or sound output aspects of the device, such as backgrounds, title bars, buttons, alert sounds, and so on.” (*Hariki*, ¶24), and “UI skins can be downloaded by the user and installed on the mobile device to alter the default UI or permanently change the UI of the device” (*id.*).

130. It is my opinion that there is motivation to combine *Hariki* and *Harris* because, while each discloses sending content specific to a mobile device, each describes in detail different aspects of the customization. *Id.*, ¶135.

131. *Harris* discloses that “when an electronic device, such as a web server accepts a connection and an HTTP request from a mobile device... [t]he headers of the HTTP request are examined by the MCF 8 in order to determine the type of device.” *Harris*, ¶24.

132. It is my opinion that the *Hariki-Harris* combination discloses receiving by a web server process a content request from a mobile device (corresponding to “receiving from the wireless device a request for an application program”), where the request includes a header containing the mobile device type (corresponding to “said request including an indication of a type of the wireless device”).

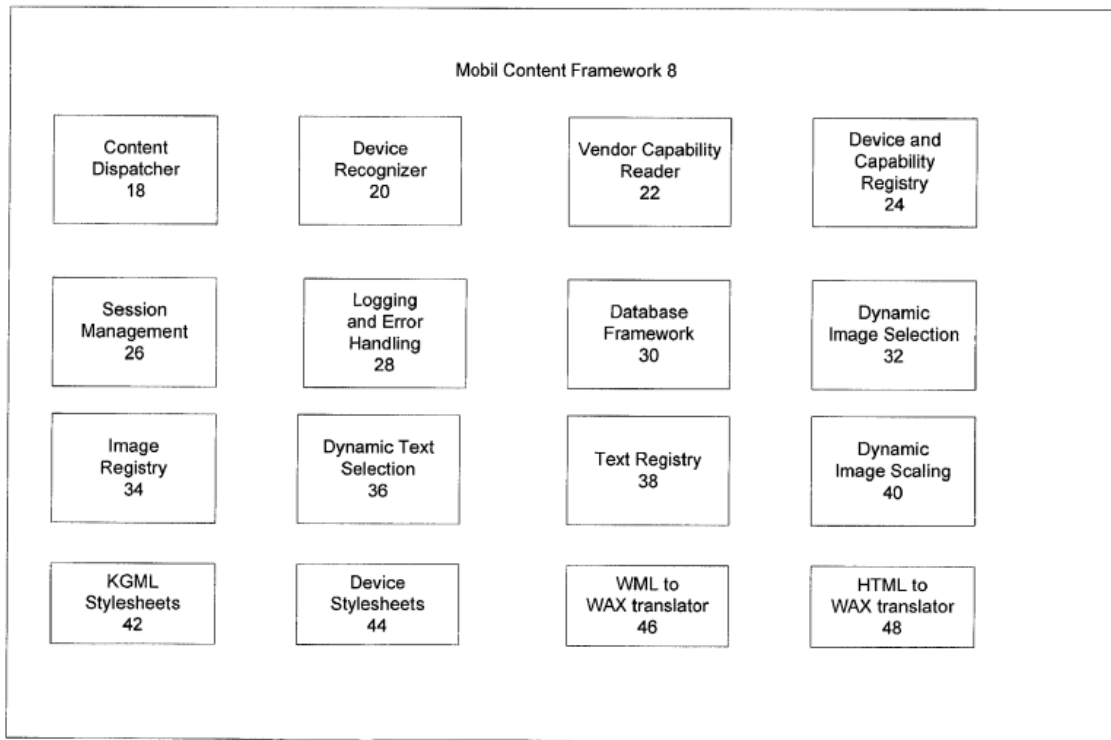
c. **1[b] executing, in response to receiving said request, said application program to generate a wireless device generic template including a plurality of content items;**

133. It is my opinion that *Hariki* in combination with *Harris* discloses this feature.

134. The '865 patent discloses that “the template engine provides a high level template (e.g., extensible markup language (XML) format) that includes the merged screens of the requested application and dynamic data provided by the business logic.” '865Pat, 3:38-42.

135. In the *Hariki-Harris* combination, *Harris* discloses, referring to Fig. 2 below, that the MCF’s “dynamic image selection component 32 uses the image registry 34, and the dynamic text selection component 36 uses the text registry 38 to determine the best content to deliver to a requesting device at any given time.” *Harris*, ¶22. Additionally, “MCF 8 includes a WAX stylesheets component 42 [and] a device stylesheets component 44.” *Id.*, ¶22.

Figure 2

*Harris*, FIG. 2.

136. *Harris* explains that “[t]he WAX stylesheets component 42 dictates the presentation of the WAX content, and the device stylesheets component 40 tailors the presentation of the content to the requesting device based on the attributes possessed by the requesting device.” *Id.*, ¶22. Moreover, *Harris* explains that “WAX is designed to enable the content developer to describe content at a more abstract level than that used in individual protocols. The greater level of abstraction enables the subsequent conversion of WAX into the languages used by the requesting mobile devices such as WML, HDML and HTML.” *Id.*, ¶7.

137. Accordingly, it is my opinion that *Harris* discloses generating WAX stylesheets, which are abstract and not specific to a mobile device (i.e., “wireless device generic template”) that include content (i.e., “plurality of content items”) tailored by the device stylesheets.

138. In *Harris*, the template generation process begins when “a web server accepts a connection and **an HTTP request from a mobile device** . . . [where] headers of the HTTP request are examined by the MCF 8 in order to determine the type of [requesting] device,” in response to which, a “registry of devices is consulted to determine device capabilities (screen size, browser version, etc.).” *Harris*, ¶24. WAX is then translated “into a language the requesting device can understand” and customized content is returned to the requesting device. *Id.*, ¶24.

139. It is my opinion that *Harris* discloses generating WAX stylesheets with content in response to receiving a request from a mobile device (i.e., “in response to receiving said request”).

140. Additionally, in the *Hariki-Harris* combination *Hariki* discloses that “server computer 102 . . . provides content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile device 108. . . . For example, server 102 can execute[] a web server process to serve

web pages,”²⁶ such as HTML files over a network to the mobile devices. *Hariki*, ¶21. Accordingly, it is my opinion that the server in the *Hariki-Harris* combination executes an application, i.e., web server process, to generate content for the mobile device.

141. It is also my opinion that *Harris* discloses a server running a web server process receiving a HTTP request from a mobile device (corresponding to “receiving said request”) and in response, executing the web server process (corresponding to “executing said application program to generate”), to generate a WAX stylesheet (corresponding to “to generate a wireless device generic template”) that includes content for presentation on the mobile device (corresponding to “including a plurality of content items”).

d. 1[c] sending a custom configuration to the wireless device, said custom configuration being specific to said application program;

142. It is my opinion that *Hariki* in combination with *Harris* discloses this feature.

143. *Hariki* explains that “[m]any different aspects of a mobile device may be customizable however, such as the format and display of menus, commands, subwindows, and so on,” and that the server can provide the customized UI to the

²⁶ Unless otherwise stated, all emphases are added.

mobile device. *Hariki*, ¶26. *Hariki* further explains that to alter the UI skin, a “UI package generator produces a downloadable UI content package consisting of [a] description file and [] converted resource files.” *Hariki*, ¶19. It is my opinion that the converted resource files and the description file in *Hariki*’s UI content package corresponded to “custom configuration” and “identification of [the] custom configuration,” respectively.

144. The UI content package includes a converted resource such as a Flash, PNG, or JPEG object used by engines that “process the applicable resource using the appropriate playback format,” such as the Flash engine for an applet or JPEG engine for a photograph. *Id.*, ¶43 The engine “converts each resource file to a format or embodiment that is compatible with the application 506,” which is “in terms of parameters such as image size, color, position, and so on.” *Id.*, ¶47. Accordingly, it is my opinion that *Hariki* discloses a server “sending a custom configuration to the wireless device.”

145. Furthermore, *Hariki* discloses “server computer 102 . . . is a content provider server computer that provides content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile device 108.” *Hariki*, ¶21. For example, the server “can be a World-Wide Web (WWW) server that stores data in the form of web pages and transmits these pages as Hypertext Markup Language (HTML) files over the Internet 110 to the

mobile devices 108 and 109.” *Id.* It is my opinion that in the context of accessing a WWW server to receive HTML files, the *Hariki-Harris* combination would have sent the customized UI when executing the web server process.

146. Furthermore, it is my opinion that the configurable aspects of the user interface may be configurable buttons and sounds. *Id.*, ¶24

147. *Hariki* explains that the converted resources in the UI content package are “files, links to files, and/or data or program objects associated with the configurable aspect of the user interface for each mobile device.” *Id.*, ¶31. In my opinion the converted resources in the UI content package configure the configurable aspects of the UI, by for example, changing a display image or color. Because the converted resources in the UI content package are generated and sent when the mobile device access the web server process, it is my opinion that *Hariki* discloses “said custom configuration being specific to said application program.”

148. It is also my opinion that *Hariki* discloses a server running a web server process sending converted resource files as part of a UI content package to a mobile device (corresponding to “sending a custom configuration to the wireless device”), where the converted resources are specific to an application program on the server (“said custom configuration being specific to said application program”).

- e. **1[d] generating a page description based on said wireless device generic template and a capability of the wireless device,**

149. It is my opinion that *Hariki* in combination with *Harris* discloses this feature.

150. The '865 patent explains that “wireless device . . . communicates with the server 130 via a device generic syntax to read the basic commands of a page description.” ’865*Pat*, 7:35-38. Furthermore, “the requested application content is received as compiled content that includes a series of basic commands representing a page description. Basic commands are precompiled by the server 230 and ready for rendering by the client.” *Id.*, 13:15-19. In my opinion, device specific HTML files are written in a generic syntax but include tags that are ready for rendering.

151. *Harris* discloses that “[t]he MCF 8 dynamically translates WAX into other markup languages using XML-based technologies. Content is translated from WAX to languages like WML, HDML and HTML. The content is not only translated into a specific language appropriate for the device, but also (unlike traditional XSL stylesheets) is tailored based upon the attributes of the requesting device.” *Harris*, ¶18.

152. *Harris* further explains that in the translation process, elements (or tags) in the WAX file are translated to be rendered in different ways, specific to the mobile device. For example, “[t]he <wax:button> element is displayed as a ‘soft-key’ for WAP devices, but as a ‘link’ for devices which understand only HTML.” *Harris*, ¶30.

153. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses generating HTML files (corresponding to “generating a page description”) from WAX content from WAX and device stylesheets (corresponding to “based on said wireless device generic template”) that is tailored based upon attributes of the requesting device (corresponding to “based on capability of the wireless device”).

f. **[1e] said page description having at least one discrete low level rendering command that is within said rendering capability of said wireless device but that is of a syntax that is wireless device generic; and**

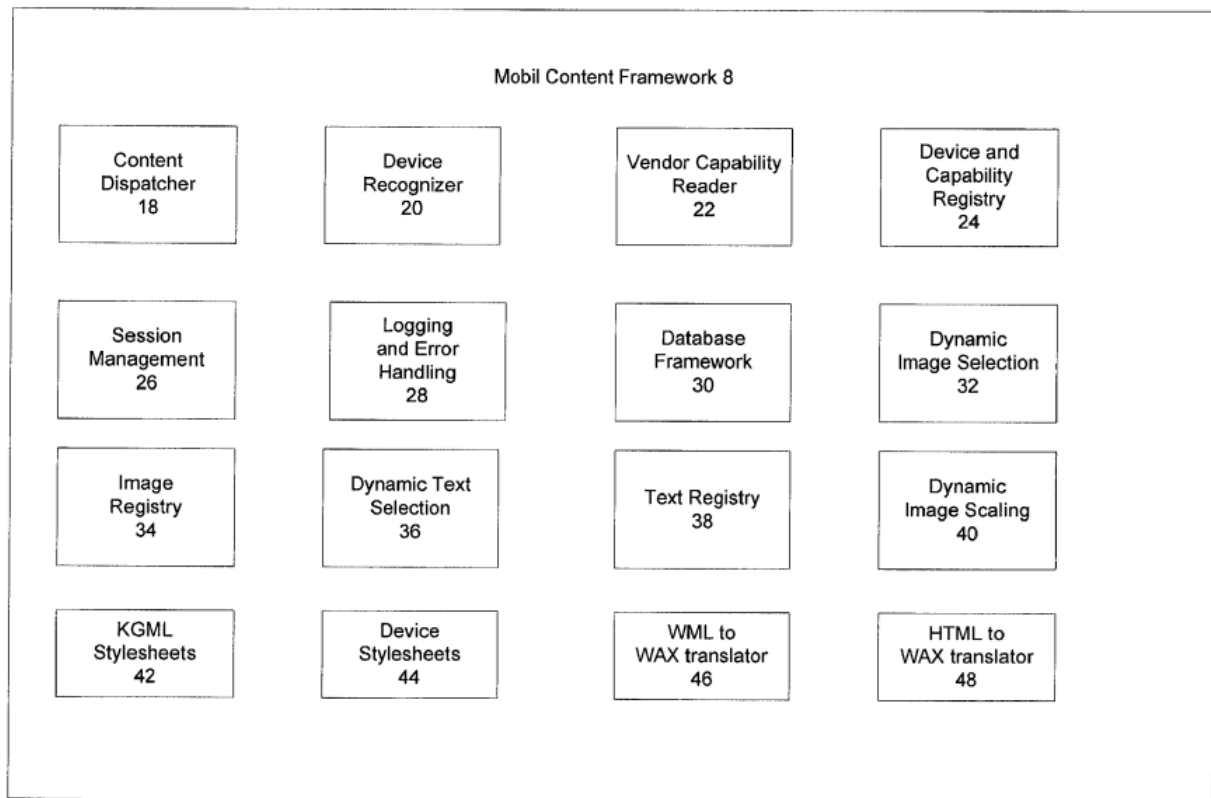
154. It is my opinion that *Hariki* in combination with *Harris* discloses this feature.

155. For example, in the *Hariki-Harris* combination, *Harris* explains that a generic WAX template “is easily extended and can be translated into a variety of different mobile device markup languages.” *Harris*, ¶7. *Harris* explains that WAX “is ‘device-smart’ in that it detects the idiosyncrasies of each device,” by “adjust[ing] to the limitations and features of each device.” *Id.*, ¶17.

156. Before sending content to a mobile device, “[t]he WAX language may be dynamically translated to a requesting device’s native language” (*id.*, ¶18), thereby “tailor[ing] content to specific requesting devices” (*id.*, ¶19). To do this, a Mobile Content Framework (“MCF”) “includes registries listing device capabilities which are utilized to provide appropriate amounts and types of content to the requesting device.” *Id.* For example, with reference to Figure 2, below, “[a]ll mobile

devices and their capabilities are kept in a device and capability registry 24.” *Id.*, ¶21. The device and capability registry 24 provides “attributes such as screen size, color depth, browser version and type, and translation rules” based on device type and application. *Id.*

Figure 2

*Harris, FIG. 2.*

157. Accordingly, it is my opinion that the *Hariki-Harris* combination tailors the HTML pages to the specific attributes (corresponding to the claimed “rendering capabilities”) of the mobile device.

158. The '865 patent explains that the page description, i.e., render commands²⁷, are “written in a syntax that is device generic regardless of the device type but whose parameters are tailored based on the wireless device capabilities.”

'865*Pat*, 13:19-22. The “syntax” may include:

a description of the scrolling area (e.g., starting and ending vertical positions), the horizontal and vertical coordinates, the width, the height, the type of component to be displayed (e.g., text, image, video, audio and the like), the unique identification of the rendering block to be used to render the component, related parameters for the rendering block and for display components (e.g., version number of the image) and the like.

Id., 13:27-34.

159. The '865 patent shows an example of basic commands in a “compiled page description” in Figure 4, reproduced below. The '865 patent explains:

Each basic command may describe a given component on the page of the requested application to be rendered and includes a rendering block to be used. For example, a basic command 410 may be a description for rendering an image whereas a basic command 430 may be the description for rendering a video clip. Accordingly, a collection of basic commands 410, 430 and 490 forms a single unified page to be rendered by the wireless device.”

²⁷ It is my opinion that the '865 patent does not disclose how to distinguish a “low level rendering command” from any other type of rendering command. Thus, to the extent that “low level rendering command” is disclosed, the rendering commands in HTML files disclose this feature.

Id., 15:59-67.

400

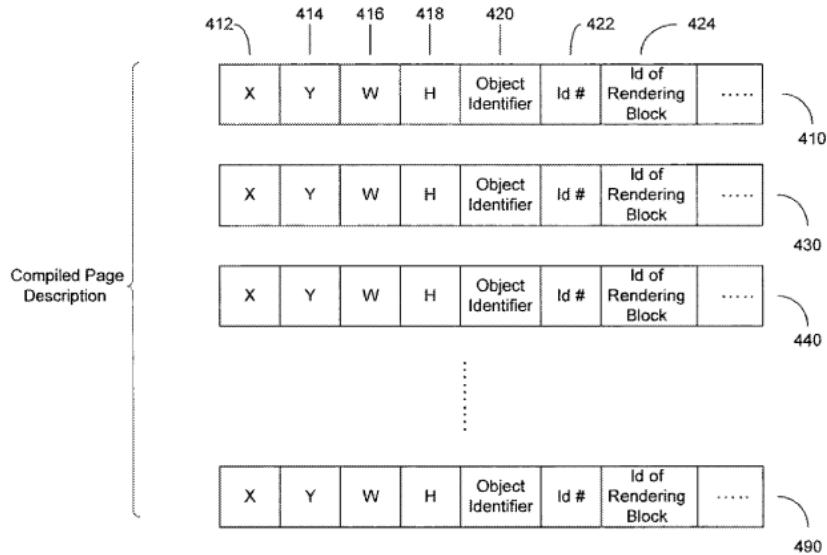


FIGURE 4

'865Pat, FIG. 4.

160. In the *Harris-Hariki* combination, it is my opinion that *Harris* discloses such basic rendering commands. For example, with respect to WAX, the generic high-level syntax used to make the HTML pages, *Harris* proves an example in the Table below:

-continued

```

        <wax:br/>
        <!--EXAMPLE 3 -->
        <wax: text id="welcome" />
        <!--EXAMPLE 4 -->
        <% if ((String)session.getAttribute("lang") ==
            null)
        out.println ("<wax:a
        href=\ "index?a=chlang\ ">Choose
        Language</wax:a>");
        %>
        <wax:br/>
        <wax:br/>
    </wax:p>
</wax:block>
</wax:doc>
</wax:wax>

```

Harris, ¶29.

161. In my opinion, the table above shows that the WAX language used a syntax that was device generic. For example, *Harris* explains that commands such “<wax:button>” are rendered differently in different devices, and text may be displayed in different languages. *Harris*, ¶30

162. In my opinion the HTML itself is in a syntax that is generic to the mobile device. For example, HTML is a standardized language that has the same syntax no matter what device is receiving it. It is also my opinion that there was a reasonable expectation of success of displaying content on different mobile devices using a generic syntax because different client devices could receive HTML-based files from a server through the World Wide Web and render them for display. Accordingly, in my opinion, in the *Hariki-Harris* combination, content data received

by the mobile device for rendering is expressed in WAX or HTML, both using a syntax that is device independent (i.e., “that is of a syntax that is wireless device generic”).

163. Furthermore, in my opinion, HTML files include tags that include low level commands to render content on a client device. In my opinion, HTML files sent by a web server are rendered into pages in a web browser on the mobile device.

164. Moreover, in my opinion HTML files are a compilation of tags (e.g., the tags are aggregated into a file) that direct a web browser to display content on device. Indeed, a compilation of commands is how the '865 patent describes the page description. '865*Pat*, 17:66-18:1 (“It is appreciated that basic commands may be low level compilation operable to render application content on the client 510 using the rendering blocks of the wireless device.”).

165. Accordingly, in my opinion, *Hariki* discloses a server that sends HTML files that include low level commands to a mobile device.

166. Therefore, in my opinion, the *Hariki-Harris* combination discloses that tailoring HTML pages to specific attributes of mobile devices, which include providing rendering commands, including low level rendering commands, (corresponding to the claimed “said page description having at least one discrete low level rendering command that is within said rendering capability of said wireless device”), where content data sent to the mobile device is expressed in WAX or

HTML using a syntax that is device independent (corresponding to the claimed “but that is of a syntax that is wireless device generic”).

g. [1f] sending said page description to the wireless device such that the wireless device is capable of presenting at least one content item from said plurality of content items using both said page description and said custom configuration.

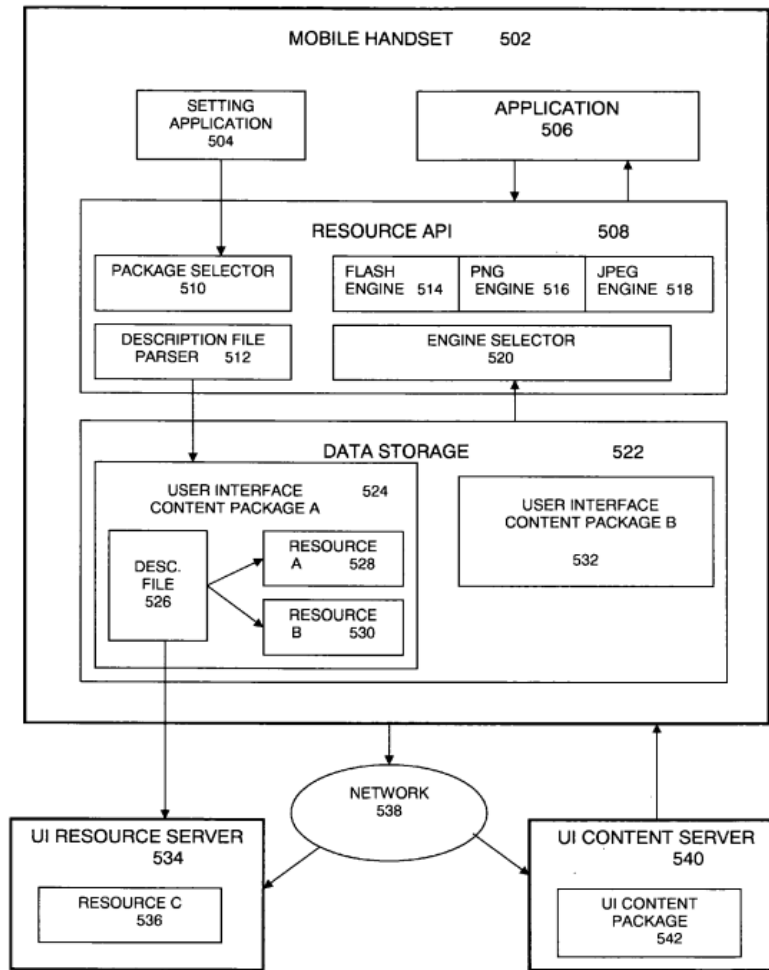
167. In my opinion, *Hariki* in combination with *Harris* discloses this feature.

168. As explained above, the *Hariki-Harris* combination sends HTML files (i.e., “page description”) to the mobile device.

169. The '865 patent explains that a “requested application content is received as compiled content that includes a series of basic commands representing a page description.” '865*Pat*, 13:15-17. Furthermore, the '865 patent functionally describes that a “graphical user interface [‘GUI’] is operable for rendering basic commands . . . and for customizing the operation and appearance of the requested application based on the custom configuration.” '865*Pat*, 2:60-63. The '865 patent further explains that the graphical user interface “render[s] content and enable[s] a user to interact” with an application, for example by allowing a user to click on a “submit” button. *Id.*, 8:2-25. With respect to Figure 6 of the '865 patent, the '865 patent paraphrases the claim language stating that “the wireless device uses a graphical user interface to generate renderable content based on the received compiled content and the custom configuration.” *Id.*, 18:63-65. The patent uses

“page description” interchangeably with “compiled content” and states “the requested application content is received as compiled content that includes a series of basic commands representing a page description.” *Id.*, 13:15-18.

170. In the *Hariki-Harris* combination, *Hariki* discloses engines in a resource API that “process the applicable resource using the appropriate playback format depending upon the type of data or program elements in the resource.” *Hariki*, ¶43. For example, with reference to Figure 5, below, *Hariki* discloses a general block diagram of the mobile device. *Hariki*, ¶41.

**FIGURE 5***Hariki*, FIG. 5.

171. *Hariki* explains that the mobile device includes application 506 that “may be a software program or utility that alters the appearance or functionality of the mobile device, or it may be a program that, when executed, provides a service to the user,” such as the web browser discussed above. *Id.*

172. As explained above, the application, e.g., a web browser, that receives both HTML files (i.e., “page description”) and converted resources in the UI

package (i.e., “custom configuration”), interacts with a “resource API 508, which contains a number of functional components such as package selector 510, description file parser 512, an engine selector 520, and one or more engines.” *Hariki*, ¶43. *Hariki* explains that the engines “process the applicable resource using the appropriate playback format,” such as the Flash engine for an applet or JPEG engine for a photograph. *Id.* The engine “converts each resource file to a format or embodiment that is compatible with the application 506,” which is “in terms of parameters such as image size, color, position, and so on.” *Id.*, ¶47.

173. It is my opinion that *Hariki*’s resource API and engines use the converted resources in the UI package and HTML file resources and format them into renderable content for the application for display as “the ‘look and feel’ or application program environment of a device” (*id.*, ¶24).

174. Additionally, it is my opinion that the client device rendering could have used JavaScript™ for displaying content received from the server. *See Linburn*, ¶50.

175. Accordingly, it is my opinion that *Hariki* discloses a server that sends HTML files that include low level commands to a mobile device (corresponding to “sending said page description to the wireless device”) and the mobile device that uses the received converted resources in the UI package and HTML file resources and formats them into renderable content for the application for display

(corresponding to “such that the wireless device is capable of presenting at least one content item from said plurality of content items using both said page description and said custom configuration”).

2. Claim 2 “The method of claim 1, wherein said wireless device generic template includes relative screen location and display object size information and is independent of screen dimensions of the wireless device.”

176. It is my opinion that *Hariki* in combination with *Harris* discloses claim 2.

177. For example, in the *Hariki-Harris* combination, *Harris* discloses WAX stylesheets to be generic templates that include content as explained in claim [1b]. See Section VIII.A.1.c. Additionally, *Harris* discloses that “[t]he WAX stylesheets component 42 dictates the presentation of the WAX content.” *Harris*, ¶22. *Harris* further explains that presentation of WAX content includes content alignment commands, such as “<wax:p align= “center”>” for center alignment. *Harris*, ¶29.

```
<?xml version=" 1. 0" encoding="utf-8"?>
<wax:wax xmlns:wax="http://www.kargo.com/wax" version="0.9">
  <wax:doc version="1.0">
    <wax:title>NY Nightlife</wax:title>
    <wax:block id="splash">
      <!--EXAMPLE 1 -->
      <wax : button href="index?a=main"
        keytype="accept" type="go" labelid="enter" />
      <wax:p align="center">
        <!--EXAMPLE 2 -->
        <wax: img srcid="logo" alt="My Nightlife"
          border="0" />
```

Harris’ system “ensures the best type and length of text is used, the best type and size of image is used, and that the content is well suited and customized for the

device attributes.” *Harris*, ¶7. Further, *Harris* discloses that the “WAX language may be dynamically translated to a requesting device's native language... For example, a WAX ‘Date-Chooser’ renders one way on the limited screen sizes of mobile phones (possibly through a series of WML decks and another way on a PDA (all on one Screen).” *Harris*, ¶18.

178. It is my opinion that in order to optimally present content to various types of mobile devices and PDAs, *Harris*’ generic WAX stylesheet includes specific location and size information for its content that are implemented through alignment commands such as “center,” which is a physical location relative to the edges of the containing window.

179. It is my opinion, accordingly, that the *Hariki-Harris* combination discloses the WAX stylesheet including content location and size information (corresponding to “wireless device generic template includes relative screen location and display object size information”) and that the WAX stylesheet and content can be optimally displayed across various types of devices (corresponding to “independent of screen dimensions of the wireless device”).

3. Claim 3 “The method of claim 1, wherein said wireless device generic template is substantially compliant with extensible markup language (XML) syntax.”

180. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, in the *Hariki-Harris* combination, *Harris* discloses WAX

stylesheets to be generic templates that include content as explained in claim [1b]. See Section VIII.A.1.c. Additionally, *Harris* discloses that the WAX is “Wireless Abstract XML.” *Harris*, ¶7.

4. Claim 4 “The method of claim 1, wherein said at least one discrete low level rendering command of said page description is operable to be executed by a plurality of rendering blocks of a graphical user interface of the wireless device.”

181. It is my opinion that *Hariki* in combination with *Harris* discloses this feature.

182. In the *Hariki-Harris* combination, as explained in claim [1d], *Harris* discloses generating HTML files to deliver content for rendering on mobile devices. See Section VIII.A.1.e.

183. Additionally, *Hariki* explains that the UI skin may consist “of a particular screen configuration, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection.” *Hariki*, ¶26. *Hariki*, explains that “[m]any different aspects of a mobile device may be customizable however, such as the format and display of menus, commands, subwindows, and so on,” and that the server can provide the customized UI. *Id.* *Hariki* explains that the UI skin allows configuration of “backgrounds, title bars, buttons, alert sounds, and so on.” *Id.*, ¶24. It is my opinion that these backgrounds, title bars, buttons, and alert sounds comprise “rendering blocks” within the context of the ’865 patent. It is also my opinion that a

HTML file can reference multiple rendering blocks, which use the received HTML files to render content at a mobile device.

184. Furthermore, it is my opinion that rendering a command is to execute that command to cause a UI to display content, because the tags and commands in HTML files are interpreted by a web browser to display content. *Id.*,

185. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses a server sending HTML files (corresponding to the claimed “said at least one discrete low level rendering command of said page description”) to a mobile device, where rendering blocks use the HTML files to render content (corresponding to “is operable to be executed by the plurality of rendering blocks of a graphical user interface of the wireless device”).

5. Claim 5 “The method of claim 4, wherein said at least one discrete low level rendering command includes a physical screen position of a display element of said at least one content item.”

186. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, the *Hariki-Harris* combination discloses low level rendering commands in HTML files as explained in claim [1e]. *See* Section VIII.A.1.f.

187. Furthermore, *Harris* discloses that the MCF “generate[s] images that are displayed correctly on each device. The image-scaling framework scales and crops images to the right size and translates between image formats.” *Harris*, ¶22. Moreover, it is my opinion that *Harris* discloses aligning content, such as a button,

to the “center” of the screen. *See, e.g., id.*, ¶29 (showing that the “wax : button” is center aligned (i.e., “<wax:p align=‘center’”)).

188. It is my opinion that the center of the screen is a “physical screen position,” as claimed. It is also my opinion that to correctly display images on a device, for example in the center of the screen, *Harris*’ HTML file would include comparable alignment commands that specify content position and location on the device. Additionally, as I described in the background section on CSS (*see* Section VII.D), it was well-known that HTML files were commonly used alongside of CSS files, and CSS files had express “absolute” positioning commands used for positioning HTML content. It was also well-known that on mobile devices, HTML applications often ran in full-screen mode, meaning that the window would fill the screen. Consequently, any positioning that was set for the window would be positioned for the screen in the same way.

189. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses a server transmitting HTML files to a client device (corresponding to the claimed “at least one discrete low level rendering command”), which include commands for content position and location on the client device’s display (corresponding to the claimed “includes a physical screen position of a display element of said at least one content item”).

6. Claim 6 “The method of claim 4, wherein said at least one discrete low level rendering command includes: a specification of a location

and a size of a screen area; and an identification of a rendering block from said plurality of rendering blocks of said graphical user interface to operate within said screen area.”

190. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, the *Hariki-Harris* combination discloses low level rendering commands in HTML files as explained in claim [1e]. See Section VIII.A.1.f.

191. Additionally, *Hariki* discloses “[u]ser interface descriptor information comprising the customizable UI elements are downloaded to a mobile device, such as 108 or 109. In one embodiment, the UI descriptor information can comprise screen parameter definitions (e.g., size, aspect ratio, icon definitions, and so on).” *Hariki*, ¶27. *Harris* discloses “a WAX ‘Date-Chooser’ renders one way on the limited screen sizes of mobile phones (possibly through a series of WML decks[]) and another way on a PDA (all on one Screen).” *Harris*, ¶18. *Harris* further discloses “some screens are large and some are small. For example, it may be best to limit the size of a restaurant review to one sentence when communicating with a mobile phone, but one paragraph may be more appropriate when communicating with a PDA with much [sic] larger screen.” *Harris*, ¶32.

192. It is my opinion that the *Hariki-Harris* combination specifies location and size of screen areas in client devices to optimally display content received from a server.

193. Furthermore, *Hariki* explains that the UI skin allows configuration of “backgrounds, title bars, buttons, alert sounds, and so on.” *Id.*, ¶24. It is my opinion that these backgrounds, title bars, buttons, and alert sounds comprise “rendering blocks” within the context of the ’865 patent.

194. It is my opinion that to render content on user interfaces of mobile devices using HTML, a HTML file can reference multiple rendering blocks.

195. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses specifying screen parameter definitions and WML decks (corresponding to the claimed “a specification of location and a size of a screen area”) and configuration of title bars, buttons, and alert sounds (corresponding to the claimed “an identification of a rendering block from said plurality of rendering blocks of said graphical user interface to operate within said screen area”) for display on a client device.

7. Claim 7 “The method of claim 6, wherein said at least one discrete low level rendering command includes parameters associated with said rendering block from said plurality of rendering blocks.”

196. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, *Harris* describes examples of WAX programming code that includes attributes for elements, such as buttons. *Harris*, ¶29. Furthermore, *Harris* explains that the WAX code is translated into HTML with other certain attributes.

Id., ¶¶30-32. It is my opinion that *Harris* shows that a “button” includes a “keytype” and “align” attribute. *Harris*, ¶29.

197. Furthermore, it is my opinion that a button comprises “rendering block” within the context of the ’865 patent. It is also my opinion that the button has associated parameters such as labels and positions that allow them to be rendered on a mobile device.

8. Claim 8 “The method of claim 1, wherein said page description includes at least one display rendering command and at least one audio rendering command.”

198. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. As explained in claim [1d], the *Hariki-Harris* combination discloses generating HTML files (corresponding to “generating a page description”) from WAX content from WAX and device stylesheets. *See* VIII.A.1.e. It is my opinion that *Harris* further discloses that “in the translation process, elements (or tags) in the WAX file are translated to be rendered in different ways, specific to the mobile device. For example, “[t]he <wax:button> element is displayed as a ‘soft-key’ for WAP devices, but as a ‘link’ for devices which understand only HTML.” *Harris*, ¶30. Accordingly, it is my opinion that *Harris* discloses a display rendering command.

199. Furthermore, *Hariki*’s UI content package “allows a user to customize the ‘look and feel’ or application program environment of a device by altering

display and/or sound output aspects of the device, such as backgrounds, title bars, buttons, alert sounds, and so on.” *Hariki*, ¶24. The resource files in the UI content package “can contain various types of data objects relating to the user interface elements of the mobile devices, such as image files, sound files, screen layouts, icons, movies, and so on.” *Id.*, 31. Accordingly, it is my opinion that *Hariki* discloses that renderable content comprises audio content. It is also my opinion that to render audio content, an audio rendering command is required.

200. Therefore, it is my opinion that the *Hariki-Harris* combination discloses display and audio rendering commands.

9. Claim 9 “The method of claim 1, wherein said application program is wireless device generic and said wireless device generic template is a wireless device generic page description template.”

201. It is my opinion that *Hariki* in combination with *Harris* discloses this feature.

202. As explained above, the *Hariki-Harris* combination discloses a web server process that sends HTML files to mobile devices. *See* Sections VIII.A.1.a, VIII.A.1.b, and VIII.A.1.g.. As I also explained above, the mobile devices may be computers or mobile devices from different manufacturers. *See* Section VIII.A.1.a. Accordingly, it is my opinion that *Hariki* discloses the web server process (i.e., “application program”) is mobile device generic.

203. Furthermore, *Harris* explains that WAX is “a generic markup language, referred to as Wireless Abstract XML (hereafter WAX) that is easily extended and can be translated into a variety of different mobile device markup languages.” *Harris*, ¶7. *Harris* further discloses that “WAX is designed to enable the content developer to describe content at a more abstract level than that used in individual protocols. The greater level of abstraction enables the subsequent conversion of WAX into the languages used by the requesting mobile devices such as WML, HDML and HTML.” *Id.* Therefore, it is my opinion that the WAX code (i.e., “wireless device generic template”) describes a page to be displayed in an abstract way that is not dependent on the type of device requesting the page of information (corresponding to the claimed “wireless device generic page description template”).

10. Claim 10 “The method of claim 1, wherein the wireless device is operable to configure a plurality of rendering blocks of a graphical user interface based on said custom configuration.”

204. It is my opinion that *Hariki* in combination with *Harris* discloses this feature.

205. *Hariki* explains that the UI skin may consist “of a particular screen configuration, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection.” *Hariki*, ¶26. *Hariki*, explains that “[m]any different aspects of a mobile device may be customizable however, such as the format and display of menus, commands, subwindows, and so on,” and that the server can provide the

customized UI. *Id.* *Hariki* explains that the UI skin allows configuration of “backgrounds, title bars, **buttons**, **alert sounds**, and so on.” *Id.*, ¶24. It is my opinion that these backgrounds, title bars, buttons, and alert sounds comprise “rendering blocks” within the context of the ’865 patent.

206. Indeed, the ’865 patent explains that rendering blocks include but are not limited to:

an edit box for entering text, static text for displaying text, an image, a pop-up menu which may appear in response to a user interaction, a drop-down menu list, tabbed menu for displaying several pages where each tab may display a text and an optional icon, **sound** for controlling audio (e.g., pause, rewind, stop, play and the like), video to display a video with visual control panel (e.g., pause, rewind, stop, play and the like), ticker to display horizontal scrolling text, **check box/radio button** to enable selection/de-selection of items, rating control for rating content (e.g., movies), poll control for displaying the current poll result, canvas for drawing objects, a tree for displaying hierarchical data, scroll bar for scrolling up/down and/or left/right, a progress bar to display download progress, a table for displaying data in a tabular form, a calendar for displaying and enabling selection/de-selection of a date and the like.

’865Pat, 8:27-43.

207. It is my opinion that that *Hariki*’s configurable buttons and sounds, for example, fall squarely within the ’865 patent’s examples of “rendering blocks.”

208. Furthermore, in the *Hariki-Harris* combination, *Hariki* discloses engines in a resource API that “process the applicable resource using the appropriate playback format depending upon the type of data or program elements in the

resource.” *Hariki*, ¶43. For example, with reference to Figure 5, below, *Hariki* discloses a general block diagram of the mobile device. *Hariki*, ¶41.

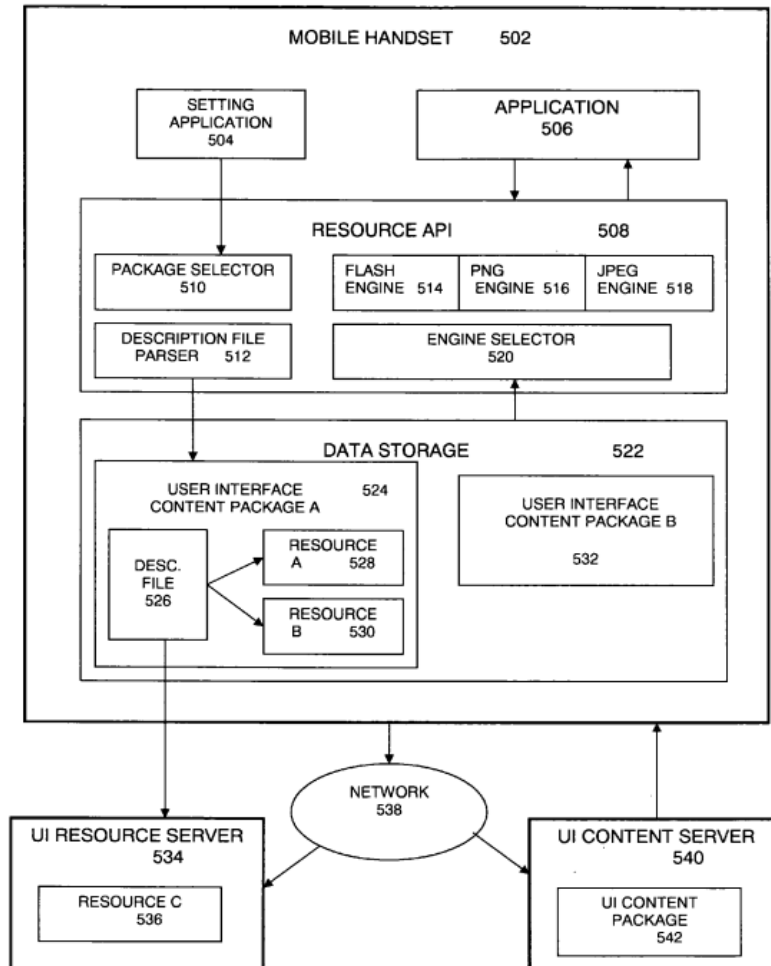


FIGURE 5
Hariki, FIG. 5.

209. It is my opinion that *Hariki* explains that the mobile device includes application 506 that “may be a software program or utility that alters the appearance or functionality of the mobile device, or it may be a program that, when executed, provides a service to the user,” such as the web browser discussed above. The

application, e.g., a web browser that, receives converted resources (i.e., “custom configuration”) in the UI content package,²⁸ interacts with a “resource API 508, which contains a number of functional components such as package selector 510, description file parser 512, an engine selector 520, and one or more engines.” *Hariki*, ¶43. *Hariki* explains that the engines “process the applicable resource using the appropriate playback format,” such as the Flash engine for an applet or JPEG engine for a photograph. *Id.* The engine “converts each resource file to a format or embodiment that is compatible with the application 506,” which is “in terms of parameters such as image size, color, position, and so on.” *Id.*, ¶47.

210. It is also my opinion that *Hariki*’s resource API and engines use the converted resources in the UI content package and format them into renderable content for the application for display as “the ‘look and feel’ or application program environment of a device.” *Id.*, ¶24. It is my opinion that the “look and feel”

²⁸ *Hariki* uses several terms interchangeably for UI content package. For example, *Hariki* describes it as “UI content” ¶19, “UI skin content” (*Hariki*, ¶27), “UI skin package data” (*id.*, ¶27), and “UI package” (*id.*, ¶40). For the purposes of this declaration, I will uniformly refer to the downloadable UI content as “UI content package.”

environment of the device was a user interface displayed to the user and corresponded to the “graphical user interface,” as claimed. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses using converted resources (corresponding to “custom configuration”) to configure a UI content package’s backgrounds, title bars, buttons, alert sounds (corresponding to “configure a plurality of rendering blocks”) to render content on a user interface (corresponding to “of a graphical user interface”).

11. Claim 11 “The method of claim 1, wherein said server stores a plurality of device profiles, said method further comprises: identifying a device profile from said plurality of device profiles associated with the wireless device based on said type of said wireless device.”

211. It is my opinion that *Hariki* in combination with *Harris* discloses this feature.

212. For example, in the *Hariki-Harris* combination, *Harris* explains that “[a]ll mobile devices and their capabilities are kept in a device and capability registry 24 [of a server’s MCF]. The device and capability registry 24 is an XML file, and allows the content developer to group devices together, while also keeping track of the Specific features of each device.” *Harris*, ¶21. Furthermore, *Harris* discloses that “a web server accepts a connection and an HTTP request from a mobile device... The headers of the HTTP request are examined by the MCF 8 in order to determine the type of device and a suitable XSL stylesheet to use for later device-specific

translation (step 62). An XML registry of devices is consulted to determine device capabilities (Screen size, browser version, etc.) (step 64).”

213. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses a server’s MCF storing mobile devices and their capabilities in a registry (corresponding to the claimed “stores a plurality of device profiles”), identifying a content-requesting mobile device’s capabilities from the registry (corresponding to the claimed “identifying a device profile from said plurality of device profiles associated with the wireless device”) based on an identified type of the mobile device (corresponding to the claimed “based on said type of said wireless device”).

12. Claim 31 “The method of claim 1, wherein the sending the custom configuration to the wireless device is in response to receiving an indication from the wireless device that the custom configuration is not stored at the wireless device.”

214. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, *Hariki* discloses that its “mobile device UI configuration system [] allows a UI skin package content provider to easily generate a new UI resource file for new or different mobile devices.” *Hariki*, ¶8. Furthermore, *Hariki* discloses “[t]ypically customized UI content is downloaded upon demand, such as when the user requests a UI upgrade or modification from a particular content provider.” *Hariki*, ¶27. It is my opinion that an upgrade from the content provider is needed when the updated UI content is not available at the mobile device.

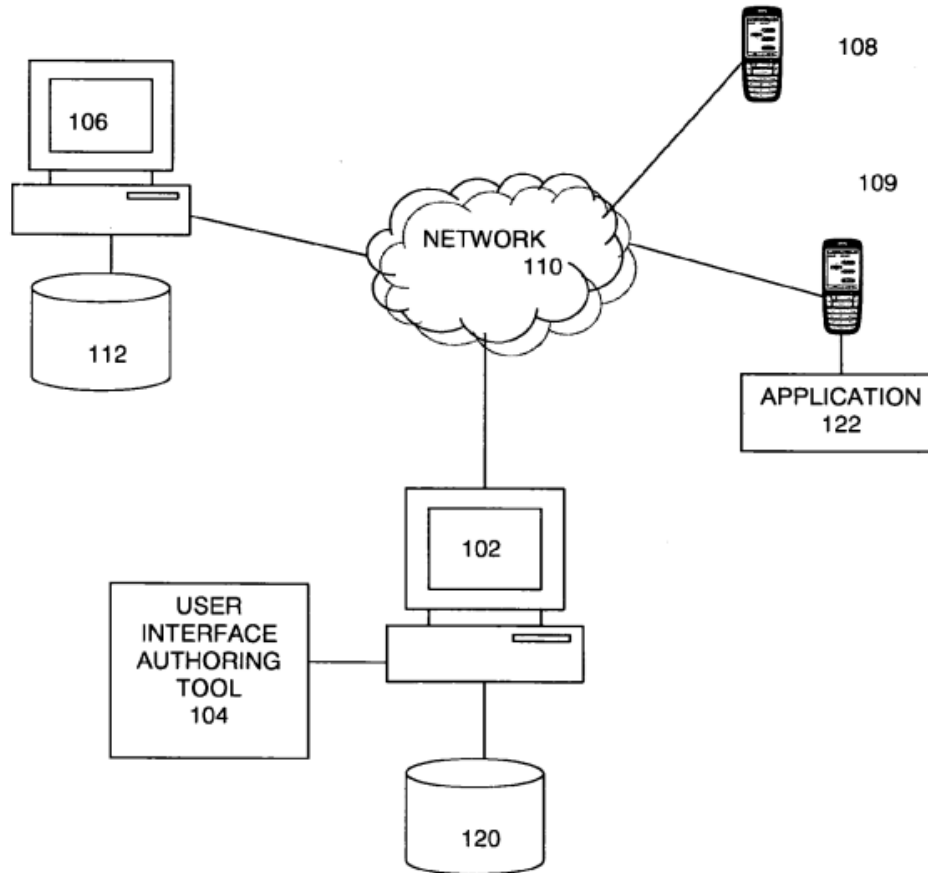
215. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses a content provider sending customized UI content to a mobile device (corresponding to “sending the custom configuration to the wireless device”) upon the mobile device requesting an upgrade or modification to the content (corresponding to “is in response to receiving an indication from the wireless device that the custom configuration is not stored at the wireless device”).

13. **Claim 12 “A non-transitory processor-readable medium storing code representing instructions to cause a processor to: receive a request for an application program from a wireless device, said request including an indication of a type of the wireless device; execute said application program to generate a wireless device generic template including a plurality of content items; request, using said indication of said type of the wireless device, a device profile from a library of device profiles, said device profile providing an indication of a rendering capability of the wireless device; eliminate, using said device profile, at least one content item from said plurality of content items from said wireless device generic template to define a device specific template associated with the wireless device, said at least one content item being unsupported by the wireless device; translate said device specific template into a page description having at least one discrete low level rendering command that is within said rendering capability of the wireless device but that is of a syntax that is wireless device generic; and send said page description to the wireless device.”**

- a. **[12pre] A non-transitory processor-readable medium storing code representing instructions to cause a processor to:**

216. It is my opinion that *Hariki* in combination with Harris discloses claim 12. To the extent the preamble is limiting, *Hariki* discloses this feature. For example, *Hariki* discloses, with reference to Figure 1 below, “server computer 102 . . .

provides content data, application programs, diagnostic tools, program components, or any other content or executable objects to the mobile device 108. . . . For example, server 102 can execute[] a web server process to serve web pages,” such as HTML files over a network to the mobile devices. *Hariki*, ¶21. Furthermore, the server “can generate a customized user interface for a plurality of different makes and types of mobile devices.” *Id.*, ¶24. *Hariki* explains that “the customized user interface is referred to as a ‘UI skin’” to “allow a user to customize the ‘look and feel’ or application program environment of a device by altering display and/or sound output aspects of the device.” *Id.*

**FIGURE 1**

217. Hariki further discloses “while only one aspect of the system is recited as embodied in machine-readable medium, other aspects may likewise be embodied in machine-readable medium.” *Hariki*, ¶59. It is my opinion that *Hariki* discloses a processor-readable medium.

218. Accordingly, it is my opinion that *Hariki* discloses generating UI skins and providing content data by a server through a machine-readable medium (corresponding to “non-transitory processor-readable medium storing code representing instructions to cause a processor”) to a mobile device.

- b. **[12a] receive a request for an application program from a wireless device, said request including an indication of a type of the wireless device;**

219. It is my opinion that each of these elements is substantively identical to that in claim [1a]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.1.b.

- c. **[12b] execute said application program to generate a wireless device generic template including a plurality of content items;**

220. It is my opinion that each of these elements is substantively identical to that in claim [1b]. Compare [12b] (“execute said application program to generate a wireless device generic template including a plurality of content items”) with [1b] (“executing . . . said application program to generate a wireless device generic template including a plurality of content items”). Accordingly, it is my opinion that *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.1.c.

- d. **[12c] request, using said indication of said type of the wireless device, a device profile from a library of device profiles,**

221. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, in the *Hariki-Harris* combination, *Harris* explains that “[a]ll mobile devices and their capabilities are kept in a device and capability registry 24 [of a server’s MCF]. The device and capability registry 24 is an XML file, and allows the content developer to group devices together, while also keeping track of the specific features of each device.” *Harris*, ¶21. Furthermore, *Harris* discloses that

“[A] web server accepts a connection and an HTTP request from a mobile device The headers of the HTTP request are examined by the MCF 8 in order to determine the type of device and a suitable XSL stylesheet to use for later device-specific translation (step 62). An XML registry of devices is consulted to determine device capabilities (screen size, browser version, etc.) (step 64).” *Harris*, ¶24.

222. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses storing mobile devices and their capabilities in a registry and identifying a content-requesting mobile device’s capabilities from the registry (corresponding to the claimed “request a device profile from a library of device profiles”) based on an identified type of the mobile device (corresponding to the claimed “using said indication of said type of the wireless device”).

e. **[12d] said device profile providing an indication of a rendering capability of the wireless device;**

223. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, in the *Hariki-Harris* combination, *Harris* explains that “[a]n XML registry of devices is consulted to determine device capabilities (screen size, browser version, etc.) (step 64).” *Harris*, ¶24. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses the stored mobile device information including capability information (corresponding to the claimed “device profile providing an indication of a rendering capability of the wireless device”).

- f. [12e] eliminate, using said device profile, at least one content item from said plurality of content items from said wireless device generic template to define a device specific template associated with the wireless device,

224. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, *Harris* discloses, regarding the MCF, that “[t]ags can be added and removed from the WAX language. . . . Alternately, a subscription plan can ensure that an enterprise always obtains the most recent version of the device file. Database classes may be added which allow many different options for pooling database connections in order to perform fast database access.” *Harris*, ¶28. *Harris* further discloses in the context of translating WAX content to HTML that a “‘welcome’ may be translated to ‘Bonjour’ for users who specify French as their preferred language or possibly ‘Welcome to our site, please take a look around’ for wireless devices, having a larger display.” *Id.*, ¶30. It is my opinion that that content on WAX stylesheets can be removed, not converted to a language, or reduced in length to eliminate portions of the content (corresponding to the claimed “eliminate, using said device profile, at least one content item”) based on a specific mobile device’s capability such as small display size stored in registry of devices.

225. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses not converting content to a language, reducing content in length, removing tags (corresponding to the claimed “eliminate . . . at least one content item from said

plurality of content items from said wireless device generic template”) based on device capability such as small screen size (corresponding to the claimed “using said device profile”) to create a WAX stylesheet and content specifically for the mobile device (corresponding to the claimed “to define a device specific template associated with the wireless device”).

g. **[12f] said at least one content item being unsupported by the wireless device;**

226. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. As explained above, it is my opinion that *Harris* discloses removing, not converting, or reducing in length content items in WAX stylesheets based on a specific mobile device’s capability such as small display size stored in registry of devices. It is also my opinion that content on WAX stylesheets can be removed (i.e., eliminate all other terms except “welcome”), when a small screen size is not capable of displaying the entire text (i.e., “please take a look around”).

227. Accordingly, it is my opinion that the *Hariki-Harris* combination discloses eliminating extra text that is unsupported by a mobile device’s small screen size (corresponding to the claimed “said at least one content item being unsupported by the wireless device”).

h. **[12g] translate said device specific template into a page description having at least one discrete low level rendering command that is within said rendering capability of the wireless device but that is of a syntax that is wireless device generic; and**

228. It is my opinion that each of these elements is substantively identical to that in claim [1d] and [1e]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.1.e and VIII.A.1.f.

i. **[12h] send said page description to the wireless device.**

229. It is my opinion that *Hariki* in combination with *Harris* discloses each of these elements is substantively identical to that in claim [1f] and [1e]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Section VIII.A.1.g. and VIII.A.1.f.

14. Claim 13 “The non-transitory processor-readable medium of claim 12, wherein said wireless device generic template contains relative screen location and display object size information independent of screen dimensions of the wireless device.”

230. It is my opinion that each of these elements is substantively identical to that in claim [2]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.2.

15. Claim 14 “The non-transitory processor-readable medium of claim 12, wherein said wireless device generic template is substantially compliant with extensible markup language (XML) syntax.”

231. It is my opinion that each of these elements is substantively identical to that in claim [3]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.3.

16. Claim 15 “The non-transitory processor-readable medium of claim 12, wherein said at least one discrete low level rendering command of said page description is operable to be executed by a plurality of

rendering blocks of a graphical user interface of the wireless device.”

232. It is my opinion that each of these elements is substantively identical to that in claim [4]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.4.

17. Claim 16 “The non-transitory processor-readable medium of claim 15, wherein said at least one discrete low level rendering command includes a physical screen position of a display element of a content item from the plurality of content items.”

233. It is my opinion that each of these elements is substantively identical to that in claim [5]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.5.

18. Claim 17 “The non-transitory processor-readable medium of claim 15, wherein said at least one discrete low level rendering command includes: a specification of a location and a size of a screen area; and an identification of a rendering block from said plurality of rendering blocks of said graphical user interface.”

234. It is my opinion that each of these elements is substantively identical to that in claim [6]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.6.

19. Claim 18 “The non-transitory processor-readable medium of claim 17, wherein said at least one discrete low level rendering command includes at least one parameter associated with said rendering block from said plurality of rendering blocks.”

235. It is my opinion that each of these elements is substantively identical to that in claim [7]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.7.

- 20. Claim 19 “The non-transitory processor-readable medium of claim 12, wherein said page description includes at least one display rendering command and at least one audio rendering command.”**

236. It is my opinion that each of these elements is substantively identical to that in claim [8]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.8.

- 21. Claim 20 “The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause said processor to: send, in response to said request, a custom configuration to said wireless device, the wireless device is operable to configure a plurality of rendering blocks of graphical user interface based on said custom configuration.”**

- a. [20a] The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause said processor to: send, in response to said request, a custom configuration to said wireless device,**

237. It is my opinion that each of these elements is substantively identical to that in claim [1c]. Compare [20a] (“send, in response to said request, a custom configuration to said wireless device”) with [1c] (“sending a custom configuration to the wireless device”). Accordingly, it is my opinion that *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.1.d.

- b. [20b] the wireless device is operable to configure a plurality of rendering blocks of graphical user interface based on said custom configuration.**

238. It is my opinion that of these elements is substantively identical to that in claim [10]. Accordingly, *Hariki* discloses these features for the same reasons discussed in Sections VIII.A.10.

22. Claim 34 “The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause the processor to: send a custom configuration to the wireless device, said custom configuration being specific to said application program, the wireless device operable to present a content item from said plurality of content items using said page description and said custom configuration.”

a. [34a] The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause the processor to: send a custom configuration to the wireless device, said custom configuration being specific to said application program,

239. It is my opinion that the elements of this claim are substantively identical to those in claim [1c]. Accordingly, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section VIII.A.1.d.

b. [34b] the wireless device operable to present a content item from said plurality of content items using said page description and said custom configuration.

240. It is my opinion that the elements of this claim are substantively identical to those in claim [1f]. Accordingly, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section VIII.A.1.g.

B. Ground 2: *Hariki* in view of *Harris* and *Dees* renders obvious Claims 1-20, 31, 34

1. **Claim 1 “A server implemented method for processing data for a wireless device, comprising: receiving from the wireless device a request for an application program, said request including an indication of a type of the wireless device; executing, in response to receiving said request, said application program to generate a wireless device generic template including a plurality of content items; sending a custom configuration to the wireless device, said custom configuration being specific to said application program; generating a page description based on said wireless device generic template and a capability of the wireless device, said page description having at least one discrete low level rendering command that is within said rendering capability of said wireless device but that is of a syntax that is wireless device generic; and sending said page description to the wireless device such that the wireless device is capable of presenting at least one content item from said plurality of content items using both said page description and said custom configuration.”**

241. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Sections VIII.A.1.a – VIII.A.1.g.

2. **Claim 2 “The method of claim 1, wherein said wireless device generic template includes relative screen location and display object size information and is independent of screen dimensions of the wireless device.”**

242. It is my opinion that Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.2.

3. **Claim 3 “The method of claim 1, wherein said wireless device generic template is substantially compliant with extensible markup language (XML) syntax.”**

243. Each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.3.

4. **Claim 4 “The method of claim 1, wherein said at least one discrete low level rendering command of said page description is operable to be executed by a plurality of rendering blocks of a graphical user interface of the wireless device.”**

244. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.4.

5. **Claim 5 “The method of claim 4, wherein said at least one discrete low level rendering command includes a physical screen position of a display element of said at least one content item.”**

245. It is my opinion that *Hariki* in combination with *Harris* discloses this feature for the same reason as explained above in Section VIII.A.5. For example, in the *Hariki-Harris* combination, *Harris* discloses that the MCF “generate[s] images that are displayed correctly on each device. The image-scaling framework scales and crops images to the right size and translates between image formats.” *Harris*, ¶22.

246. Moreover, it is my opinion that *Harris* discloses aligning content, such as a button, to the “center” of the screen. See, e.g., *id.*, ¶29 (showing that the “wax : button” is center aligned (i.e., “<wax:p align=‘center’”)). It is my opinion that the center of the screen is a “physical screen position,” as claimed. It is also my opinion that to correctly display images on a device, for example in the center of the screen, *Harris*’ HTML file would include comparable alignment commands that specify content position and location on the device. Additionally, as I described in the background section on CSS (*see* Section VII.D), it was well-known that HTML files were commonly used alongside of CSS files, and CSS files had express “absolute”

positioning commands used for positioning HTML content. It was also well-known that on mobile devices, HTML applications often ran in full-screen mode, meaning that the window would fill the screen. Consequently, any positioning that was specified for the window would be positioned for the screen in the same way.

247. *Dees*, like the *Hariki-Harris* combination of a transmitting HTML files and WAX stylesheets from server to client, discloses a client device that receives HTML, CSS files, and stylesheet documents from a server and renders a user interface based on the received documents. *Dees*, ¶51. Furthermore, *Dees* explains that “style sheet document [] can comprise style information in any language suitable for marking up style information, such as CSS or DSSSL. It is also possible to use XML to mark up style information.” *Id.*, ¶53. Additionally, *Dees* discloses that “HyperText Markup Language HTML also provides [] ways to provide a user interface definition.” *Id.*, ¶37. *Dees* further states that “[u]sing a language such as CSS or DSSSL, style properties such as the font or the color of a button or the position of a checkbox on the screen can easily be defined.” *Id.*, ¶3.

248. It is my opinion that the system disclosed by *Dees* would be compatible with and could be adapted and applied to the *Hariki-Harris* combination because it has the same general architecture as *Hariki-Harris*’ client-server system because both client-server systems are directed towards rendering user interfaces on client devices based on commands transmitted from the server. Thus, to the extent that

Patent Owner contends that the *Hariki-Harris* combination does not disclose “discrete low level rendering command includes a physical screen position of a display element of said at least one content item,” it is my opinion that there is motivation to combine the *Hariki-Harris* client-server system with the teachings of *Dees* directed to a server transmitting HTML and CSS commands that define positions of content display elements on the client device’s screen.

249. In particular, to the extent that Patent Owner contends that, while the *Hariki-Harris* combination discloses proper display of content on a mobile device, it does not expressly disclose the converted HTML file, it is my opinion that there is motivation to find a method to specify screen position of a content display element on a client device and look to *Dees* for that method.

250. Moreover, it is my opinion that there is a reasonable expectation of success in implementing *Dees*’ system of receiving HTML, CSS files, and stylesheet documents from a server and rendering a user interface based on the received documents with *Hariki-Harris*’ content display system.

251. For example, if Patent Owner should contend that while the *Hariki-Harris* combination discusses aligning content, such as a button, to the “center” of the screen (*Harris*, ¶29), it does not show that the HTML file commands specifying the physical screen position of “center” on the screen, it is my opinion that *Dees* disclose such commands (*Dees*, ¶76).

252. Accordingly, to the extent that Patent Owner argues the *Hariki-Harris* combination does not disclose “discrete low level rendering command includes a physical screen position of a display element of said at least one content item,” it is my opinion that *Dees* fills such a gap in the *Hariki-Harris* combination by explaining it “provide[s] style information” such as “the X- and Y- coordinates on the screen.” *Dees*, ¶38. It further explains that “[u]sing a language such as CSS or DSSSL, style properties such as the font or the color of a button or the position of a checkbox on the screen can easily be defined.” *Dees*, ¶3. Furthermore, *Dees* discloses that “generic style definitions such as the font to be used can be provided in a generic style sheet document, and platform-specific style definitions such as the positioning of the user interface elements in a window on the screen can be provided in specific style sheet documents.” *Dees*, ¶4. As I described above in the background section on CSS VII.D., it was well-known that CSS includes absolute positioning commands that specify the position of an item. And furthermore, on mobile devices, windows were often full-screen so that the position of an item that was specified for a window was thus also specified for the screen. Therefore, it is my opinion that there is motivation to combine *Dees* and the *Hariki-Harris* combination because, while each discloses proper display of content on a device, *Dees*’ disclosure of using translated commands included in HTML files to specify content location on a device further specifies a way for the *Hariki-Harris* combination to correctly display content.

253. Accordingly, it is my opinion that the *Hariki-Harris-Dees* combination discloses a server transmitting HTML and CSS files (corresponding to the claimed “at least one discrete low level rendering command”) that identify user interface elements (corresponding to the claimed “of a display element of said at least one content item”) on a client device’s screen (corresponding to “includes a physical screen position”).

6. **Claim 6 “The method of claim 4, wherein said at least one discrete low level rendering command includes: a specification of a location and a size of a screen area; and an identification of a rendering block from said plurality of rendering blocks of said graphical user interface to operate within said screen area.”**

254. It is my opinion that *Hariki* in combination with *Harris* and *Dees* discloses this feature. As explained in claim [1e], *Harris* discloses a discrete low level rendering command. *See* Section VIII.A.1.f. As I also explained in claim [5], *Dees* discloses a physical screen position of a display element. *See* Section VIII.B.5. Furthermore, as shown below, *Dees* illustrates a device screen center and left location in an embodiment of the device-specific style sheet. *Dees*, ¶62.

```

(1)  <?xml version="1.0"?>
(2)  <uisl level="2">
(3)    <4x3>
(4)      <halign>
(5)        <all class="settings">center</all>
(6)        <all class="effects">left</all>
(7)      </halign>
(8)    <level3 stylesheet="stylesheet-landscape-4x3.uisl"/>
(9)  </4x3>
(10) </uisl>

```

255. *Dees* further explains “style sheet document 212 provides more specific style information for a graphical display with a landscape orientation, by providing style information for a graphical display with a 4:3 aspect ratio.” *Dees*, ¶65. *Dees* explains that “[t]he user interface module 420 parses the second style sheet document 212 and determines that more specific style information is available for displays with resolutions in a ratio of 4:3.” *Id.* Furthermore, “[s]ince the display 452 has a 640x480 resolution, which corresponds to a ratio of 4:3, the user interface module 420 collects this information.” *Id.* *Dees* further explains “style sheet document provides specific style sheet information for graphical devices with a 640x480 or with a 1024x768 pixel resolution display. This style sheet document could also be made valid by defining an appropriate DTD.” *Id.*, ¶68. Accordingly, it is my opinion that the *Hariki-Harris-Dees* combination provides HTML files that specify location such as center, and size such as 640x480 pixel resolution display of a mobile device’s display.

256. *Dees* further discloses “style properties such as the font or the color of a button or the position of a checkbox on the screen can easily be defined” (*Dees*, ¶3), and “platform-specific style definitions such as the positioning of the user interface elements in a window on the screen can be provided in specific style sheet documents” (*id.*, ¶4). Accordingly, it is my opinion that the *Hariki-Harris-Dees*

combination identifies interface elements to be positioned in specific locations on a user's display window.

257. Accordingly, it is my opinion that the *Hariki-Harris-Dees* combination provides HTML files that specify location such as center, and size such as 640x480 pixel resolution display of a mobile device's display (corresponding to the claimed "wherein said at least one discrete low level rendering command includes a specification of a location and a size of a screen area") and identifies interface elements to be positioned in specific locations on a user's display window (corresponding to the claimed "identification of a rendering block from said plurality of rendering blocks of said graphical user interface to operate within said Screen area").

7. Claim 7 "The method of claim 6, wherein said at least one discrete low level rendering command includes parameters associated with said rendering block from said plurality of rendering blocks."

258. It is my opinion that *Hariki* in combination with *Harris* and *Dees* discloses this feature.

259. For example, *Harris* describes examples of WAX programming code that includes attributes for elements, such as buttons. *Harris*, ¶29. Furthermore, *Harris* explains that the WAX code is translated into HTML with other certain attributes. *Id.*, ¶¶30-32. For example, *Harris* shows that a "button" includes a "keytype" and "align" attribute. *Id.* Furthermore, it is my opinion that a button

comprises “rendering block” within the context of the ’865 patent. It is also my opinion that the button have associated parameters such as labels and positions that allow them to be rendered on a mobile device. Additionally, *Dees* also lists (shown below) an “interpretation of the elements and attributes used in the...style sheet document.” *Dees*, ¶75.

Tag	Attributes	Meaning
<general> </general>	none	Contains general screen based style attributes, such as background colors.
<bgcolor> </bgcolor>	none	Defines the background color of the user interface elements. This is done by mapping all or certain UI elements to a color value. Color values are defined using the following syntax: '#' + 2-char hex value for red (between 00 to ff) + 2-char hex value for green + 2-char hex value for blue, E.g.: #ff00ff for purple color.
 	none	Specifies a hint about which kind of font should be used. The possible values are: - serif-proportional - sansserif-proportional - serif-nonproportional - sansserif-nonproportional Serif fonts are e.g. "Times" and "Garamond" Sans-serif fonts are e.g. "Arial" and "Helvetica". All these fonts are proportional. An example of a non-proportional font is "Courier".
<portrait> </portrait>	none	Provides style hints for portrait screens (i.e. when the top is narrower than the sides in the position normally used by people). The style attributes that can be defined here are the layout and the orientation of the elements. Others are defined in level 2 of the UI stylesheet language (referred to by a <level2>-tag inside this <portrait>-tag).
<landscape> </landscape>	none	Provides style hints for landscape screens (i.e. when the top is broader than the sides in the position normally used by people). The style attributes, which can be defined here, are the layout and the orientation of the elements. Others are defined in level 2 of the UI stylesheet language (referred to by a <level2>-tag inside this <landscape>-tag).
<layout> </layout>	none	Specifies how the elements should be ordered inside the containers. All layout "algorithms" which can be chosen here, start in the upper-left corner of the screen and place the elements one by one in vertical direction (default) or horizontal if the orientation of this container is specified as horizontal (see <orientation>-tag). The "algorithms" which can be chosen are: - panelsplit: place the elements one after another including complete subcontainers with their elements and if the next element does not fit anymore, place it on a new panel, in this way splitting the user interface over several panels. It is required for a client to provide some means of navigation between the different panels if this layout algorithm is used. - panelgrid: place the elements one by one and continue in another column (orientation: vertical) or in another row (orientation: horizontal) if the next element does not fit anymore, forming a grid. A grid means that all columns and all rows have the same number of grid cells the size of which is determined by the biggest element in the row and the column. If no more columns (or rows) fit on the screen, the client can place them on another panel (i.e. split the UI over several panels) or provide a means of scrolling. - paneltree: place the elements one by one and continue in the next column (orientation: vertical) or in the next row (orientation: horizontal), but independent from the previous column/row, i.e. does not have to form a grid, so each column can contain a different number of UI elements. - menusplit: same as panelsplit, but now the subcontainers can be hidden. Only the label of the subcontainer has to be shown and not all its elements, in this way forming a hierarchy of menus and submenus. - menutree: (the default algorithm for containers)

260. It is, therefore, my opinion that the element interpretation and attributes in *Dees* are the parameters associated with interface elements (i.e., “rendering blocks”).

- 8. Claim 8 “The method of claim 1, wherein said page description includes at least one display rendering command and at least one audio rendering command.”**

261. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.8.

- 9. Claim 9 “The method of claim 1, wherein said application program is wireless device generic and said wireless device generic template is a wireless device generic page description template.”**

262. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.9.

- 10. Claim 10 “The method of claim 1, wherein the wireless device is operable to configure a plurality of rendering blocks of a graphical user interface based on said custom configuration.”**

263. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.10.

- 11. Claim 11 “The method of claim 1, wherein said server stores a plurality of device profiles, said method further comprises: identifying a device profile from said plurality of device profiles associated with the wireless device based on said type of said wireless device.”**

264. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.11.

12. **Claim 31 “The method of claim 1, wherein the sending the custom configuration to the wireless device is in response to receiving an indication from the wireless device that the custom configuration is not stored at the wireless device.”**

265. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.12.

13. **Claim 12 “A non-transitory processor-readable medium storing code representing instructions to cause a processor to: receive a request for an application program from a wireless device, said request including an indication of a type of the wireless device; execute said application program to generate a wireless device generic template including a plurality of content items; request, using said indication of said type of the wireless device, a device profile from a library of device profiles, said device profile providing an indication of a rendering capability of the wireless device; eliminate, using said device profile, at least one content item from said plurality of content items from said wireless device generic template to define a device specific template associated with the wireless device, said at least one content item being unsupported by the wireless device; translate said device specific template into a page description having at least one discrete low level rendering command that is within said rendering capability of the wireless device but that is of a syntax that is wireless device generic; and send said page description to the wireless device.”**

266. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.13.

14. **Claim 13 “The non-transitory processor-readable medium of claim 12, wherein said wireless device generic template contains relative screen location and display object size information independent of screen dimensions of the wireless device.”**

267. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.14.

- 15. Claim 14 “The non-transitory processor-readable medium of claim 12, wherein said wireless device generic template is substantially compliant with extensible markup language (XML) syntax.”**

268. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.15.

- 16. Claim 15 “The non-transitory processor-readable medium of claim 12, wherein said at least one discrete low level rendering command of said page description is operable to be executed by a plurality of rendering blocks of a graphical user interface of the wireless device.”**

269. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.16.

- 17. Claim 16 “The non-transitory processor-readable medium of claim 15, wherein said at least one discrete low level rendering command includes a physical screen position of a display element of a content item from the plurality of content items.”**

270. It is my opinion that each of these elements is substantively identical to that in claim [5]. Accordingly, the combination of *Hariki*, *Harris*, and *Dees* discloses these features for the same reasons discussed in Sections VIII.B.5.

- 18. Claim 17 “The non-transitory processor-readable medium of claim 15, wherein said at least one discrete low level rendering command includes: a specification of a location and a size of a screen area; and an identification of a rendering block from said plurality of rendering blocks of said graphical user interface.”**

271. It is my opinion that each of these elements is substantively identical to that in claim [6]. Accordingly, the combination of *Hariki*, *Harris*, and *Dees* discloses these features for the same reasons discussed in Sections VIII.B.6.

- 19. Claim 18 “The non-transitory processor-readable medium of claim 17, wherein said at least one discrete low level rendering command includes at least one parameter associated with said rendering block from said plurality of rendering blocks.”**

272. It is my opinion that each of these elements is substantively identical to that in claim [7]. Accordingly, the combination of *Hariki*, *Harris*, and *Dees* discloses these features for the same reasons discussed in Sections VIII.B.7.

- 20. Claim 19 “The non-transitory processor-readable medium of claim 12, wherein said page description includes at least one display rendering command and at least one audio rendering command.”**

273. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.20.

- 21. Claim 20 “The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause said processor to: send, in response to said request, a custom configuration to said wireless device, the wireless device is operable to configure a plurality of rendering blocks of graphical user interface based on said custom configuration, the wireless device is operable to configure a plurality of rendering blocks of graphical user interface based on said custom configuration.”**

274. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.21.

- 22. Claim 34 “The non-transitory processor-readable medium of claim 12, further comprising code representing instructions to cause the processor to send a custom configuration to the wireless device, said custom configuration being specific to said application program, the wireless device operable to present a content item from said plurality of content items using said page description and said custom configuration.”**

275. It is my opinion that each of these elements is disclosed by the combination of *Hariki* and *Harris* as explained in Section VIII.A.22.

XI. SECONDARY CONSIDERATIONS

276. Based on my expertise and experience of the industry and review of the asserted prior art and other references cited in my declaration, I am not aware of any evidence of secondary considerations that would support a finding of non-obviousness. The asserted prior art demonstrates that the elements recited in the challenged claims of the '865 patent were readily available in the prior art, and there was no evidence of failure by others.

XII. CONCLUSION

277. Based on my review and above analysis of the prior art, it is my opinion that claims 1-20, 31 and 34 of the '865 patent would have been obvious to a POSITA in view of *Hariki* and *Harris*, and also would have been obvious to a POSITA in view of *Hariki*, *Harris*, and *Dees*.

278. I declare under penalty of perjury that the foregoing is true and correct. I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed this April 20, 2023.

A handwritten signature in black ink, appearing to read "Ben B. Bederson", written over a horizontal line.

Benjamin B. Bederson