

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NETFLIX, INC.,
Petitioner,

v.

GOTV STREAMING, LLC
Patent Owner.

Case No. IPR2023-00758

U.S. Patent No. 8,478,245

**DECLARATION OF BENJAMIN B. BEDERSON
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,478,245**

TABLE OF CONTENTS

PETITIONER’S EXHIBIT LIST	i
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
B. Wireless Devices	24
C. Client-Server Architectures	25
D. Overview of U.S. Patent Application Publication No. 2007/0150617 to Hariki, filed on July 25, 2006 (“ <i>Hariki</i> ”).	30
E. Overview of U.S. Patent Application Publication No. 2003/0023755 to Harris et al., filed on December 18, 2001 (“ <i>Harris</i> ”).	34
F. Other Evidence Regarding the State of the Art.....	38
VIII. SUMMARY OF THE ’245 PATENT	38
A. Background of the Patented Subject Matter.....	38
B. The ’245 Patent Specification	48
C. The ’245 Patent File History (Ex. 1004).....	52

D.	Challenged Claims	53
E.	Claim Construction (37 C.F.R. § 42.104(b)(3)).....	60
IX.	PRIORITY DATE OF THE '245 PATENT	60
X.	APPLICATION OF THE PRIOR ART TO CLAIMS 1–33.....	61
A.	Ground 1: <i>Hariki</i> in view of <i>Harris</i> renders obvious Claims 1–33	61
1.	Claim 1 A method of generating content that is renderable by a wireless device, said method comprising: transmitting, to said wireless device, an identification of a custom configuration of a plurality of rendering blocks of said wireless device, wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application; and transmitting, to said wireless device, compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application, wherein said compiled content is generated in part from execution of said application, wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device, and wherein said custom configuration is applicable to said first and second compiled content, wherein said compiled content and said custom configuration are usable by a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration.	61
2.	Claim 2 “A method as described in claim 1 wherein said using a graphical user interface comprising said plurality of rendering blocks to generate renderable content comprises: processing said compiled content using a reader of said wireless device; and issuing commands from said reader to individual rendering blocks of said graphical user interface based on said rendering commands of said compiled content.”	92

3. Claim 3 “A method as described in claim 1 wherein said render able content comprises audio content and display content.” 95
4. Claim 4 “A method as described in claim 1 wherein said compiled content is partially resultant from said application operating on a remote server.” 95
5. Claim 5 “A method as described in claim 1 wherein said compiled content is specific to the rendering capabilities of said wireless device.” 96
6. Claim 6 “A method as described in claim 1 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.” 98
7. Claim 7 “A method as described in claim 6 wherein said custom configuration comprises a syntax that is generic regarding said wireless device type.” 102
8. Claim 8 “A method as described in claim 1 wherein said custom configuration comprises configuration information and content specific to said application.” 102
9. Claim 9 “A method as described in claim 1 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored by said wireless device and wherein further said identifying said custom configuration comprises receiving an identifier that identifies said custom configuration.” 104
10. Claim 10 “A method as described in claim 1 further comprising receiving and storing said custom configuration.” 105
11. Claim 11 “A method as described in claim 1 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across a display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.” 106

12. Claim 12 “A non-transitory computer usable medium comprising instructions therein that when executed by a processor implement a method of rendering content on a wireless device, said method comprising: receiving an identification of a custom configuration of a plurality of rendering blocks of said wireless device, wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application; receiving compiled content generated in part from execution of said application wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device; using a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration; and rendering said renderable content on said wireless device, wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content.” 107
13. Claim 13 “A non-transitory computer usable medium as described in claim 12 wherein said using a graphical user interface comprising said plurality of rendering blocks to generate renderable content comprises: processing said compiled content using a reader of said wireless device; and issuing commands from said reader to individual rendering blocks of said graphical user interface based on said rendering commands of said compiled content.” 110
14. Claim 14 “A non-transitory computer usable medium as described in claim 12 wherein said renderable content comprises audio content and display content.” 110
15. Claim 15 “A non-transitory computer usable medium as described in claim 12 wherein said compiled content is partially result ant from said application operating on a remote server.” 110

16. Claim 16 “A non-transitory computer usable medium as described in claim 12 wherein said compiled content is specific to the rendering capabilities of said wireless device.” 110
17. Claim 17 “A non-transitory computer usable medium as described in claim 12 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type. 111
18. Claim 18 “A non-transitory computer usable medium as described in claim 17 wherein said custom configuration comprises a Syntax that is generic regarding said wireless device type.” 111
19. Claim 19 “A non-transitory computer usable medium as described in claim 12 wherein said custom configuration comprises configuration information and content specific to said application.” 111
20. Claim 20 “A non-transitory computer usable medium as described in claim 12 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored by said wireless device and wherein further said identifying said custom configuration comprises receiving an identifier that identifies said custom configuration.” 111
21. Claim 21 “A non-transitory computer usable medium as described in claim 12 wherein said method further comprises receiving and storing said custom configuration.” 112
22. Claim 22 “A non-transitory computer usable medium as described in claim 12 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across a display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.” 112

23. Claim 23 “A wireless device operable to communicate with a remote server, said wireless device comprising: a transceiver coupled to a bus and operable to receive a custom configuration that is associated with an application, said transceiver also operable to receive compiled content generated in part from execution of said application and comprising a plurality of rendering commands expressed in a syntax that is generic to said wireless device; a memory coupled to said bus and operable to store said compiled content and said custom configuration; a processor coupled to said bus and operable to implement the following: a graphical user interface comprising a plurality of rendering blocks and operable to generate renderable content based on said compiled content and said custom configuration wherein said custom configuration is operable to configure said plurality of rendering blocks to render content in a manner customized to said application; an engine for reading said compiled content and responsive thereto for causing said graphical user interface to generate said renderable content based on said render commands; a display device coupled to said bus and operable to render a portion of said renderable content, wherein said compiled content comprises: a first compiled content specific to a first page of said application; and a second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content. 112
24. Claim 24 “The wireless device as described in claim 23 wherein said engine is further operable to issue commands to individual rendering blocks of said graphical user interface based on said plurality of rendering commands of said compiled content.” 124
25. Claim 25 “The wireless device as described in claim 23 further comprising an audio rendering device coupled to said bus and wherein said renderable content comprises audio content and display content.” 124

- 26. Claim 26 “The wireless device as described in claim 23 wherein 25 said compiled content is partially resultant from said application operating on a remote server.” 126
- 27. Claim 27 “The wireless device as described in claim 23 wherein said compiled content is specific to the rendering capabilities of said wireless device.” 127
- 28. Claim 28 “The wireless device as described in claim 23 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.” 127
- 29. Claim 29 “The wireless device as described in claim 28 wherein said custom configuration is expressed in a syntax that is generic to said wireless device type.” 127
- 30. Claim 30 “The wireless device as described in claim 23 wherein said custom configuration comprises configuration information and content specific to said application.” 127
- 31. Claim 31 “The wireless device as described in claim 23 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored in said memory.” 128
- 32. Claim 32 “The wireless device as described in claim 23 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across said display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.” 128
- 33. Claim 33 “The wireless device as described in claim 23 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of a scroll bar on said display screen of said wireless device; a second block that controls the rendering of an input box on said display screen of said wireless device; and a third block that

	control the rendering of a text display region on said display screen of said wireless device.”.....	128
34.	Motivation to Combine	130
XI.	Secondary Considerations	130
XII.	CONCLUSION.....	130

PETITIONER'S EXHIBIT LIST

Exhibit Number	Description
1001	U.S. Patent No. 8,478,245 (“’245Pat”)
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. 8,478,245 (“’245FH”)
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 No. 7,447,486 (“ <i>Tamura</i> ”)
1008	U.S. Patent No. 6,996,627 (“ <i>Carden</i> ”)
1009	U.S. Patent No. 6,669,564 (“ <i>Young</i> ”)
1010	U.S. Patent No. 6,732,183 (“ <i>Graham</i> ”)
1011	U.S. Patent No. 6,507,727 (“ <i>Henrick</i> ”)
1012	Webpage excerpt from Computer History Museum at https://www.computerhistory.org/tdih/april/6/ (last accessed March 29, 2023)
1013	https://www.webdesignerdepot.com/2009/05/the-evolution-of-cell-phone-design-between-1983-2009/
1014	Excerpts from Dan R. Olsen, Jr., Developing user interfaces (1998)
1015	Alok Sinha, Client-server computing, in Communications of the ACM, 35, 7 (1992)
1016	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

Exhibit Number	Description
1017	Eric Kasten, HTML: A Gentle Introduction, Linux Journal (July 1995), available at https://www.linuxjournal.com/article/1081
1018	U.S. Patent Application Publication No. 2013/0124961 (“ <i>Linburn</i> ”)
1019	U.S. Patent No. 8,111,326 (“ <i>Talwar</i> ”)
1020	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)
1021	Complaint in <i>GovTV Streaming, LLC v. Netflix, Inc.</i> , No. 2:22-cv-07556-RGK-SHK (C.D. Cal. October 17, 2022)
1022	U.S. Patent Application Publication No. 2003/0106022 (“ <i>Goodacre</i> ”)
1023	https://web.archive.org/web/20070406171650/http://www.zenzui.com/products.html

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 1–233(the “Challenged Claims”) of U.S. Patent No. 8,478,245 (“the ’245 patent”) (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 ’245patent 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 am also co-founder and prior Chief Technology Officer of Hazel Analytics, 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 '245 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

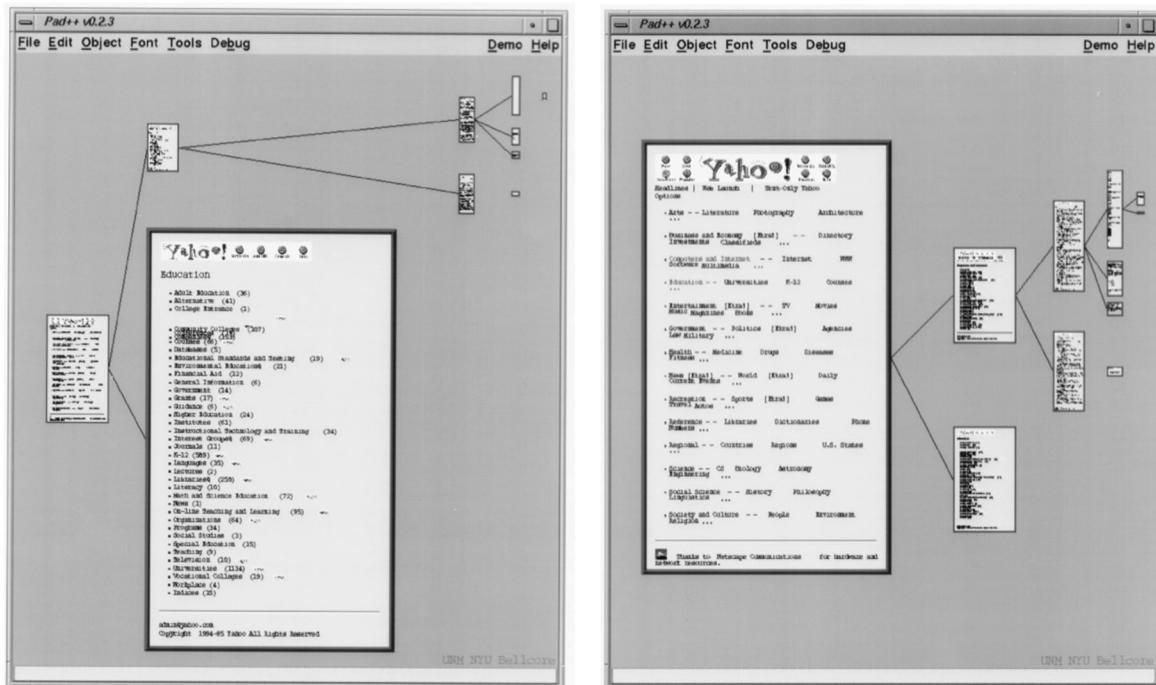


Figure 5. Many different HTML documents loaded in Pad++. Their layout implicitly shows the history of the user's interaction. The two views show the same tree focused on different nodes

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.

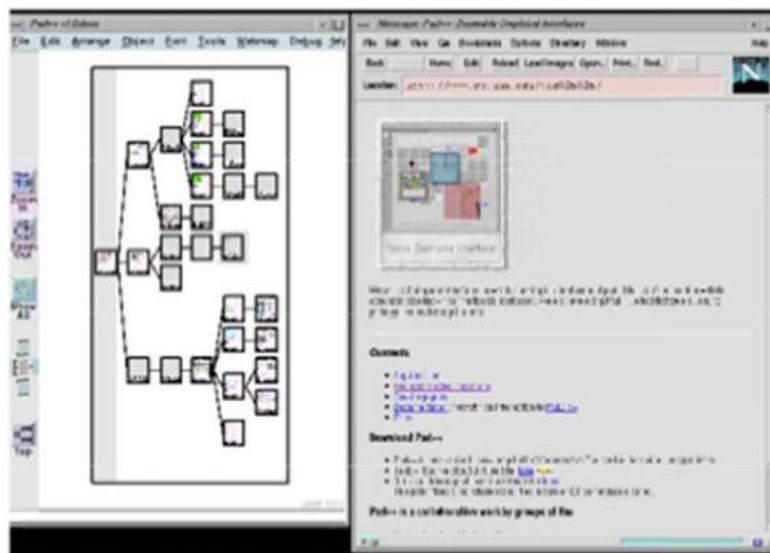


Figure 1 from PadPrints paper

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>

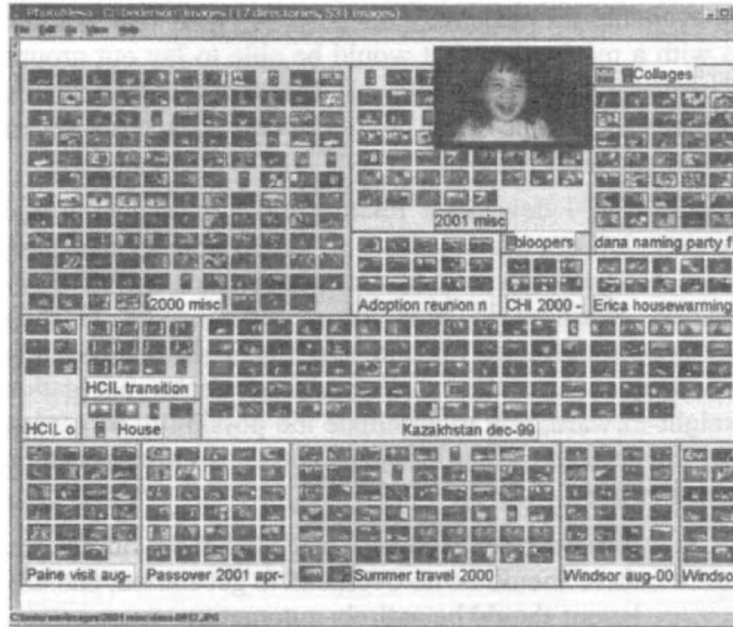


Figure 1: Screen snapshot of PhotoMesa with over 500 images in 17 groups.

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>.

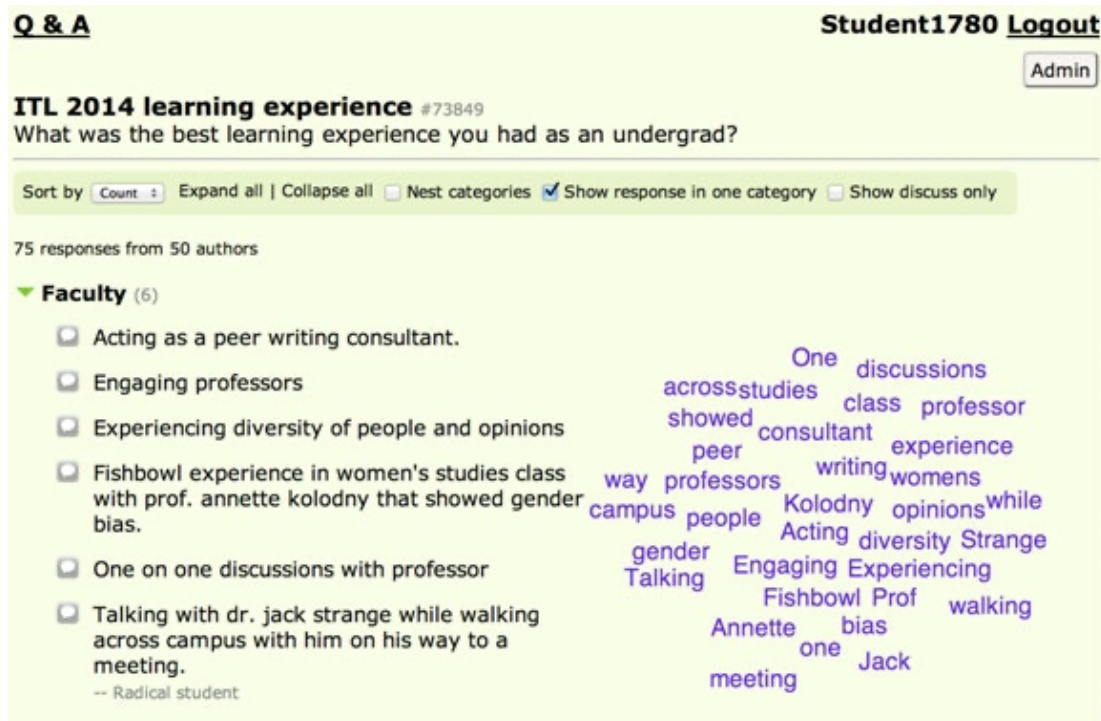
21. In 2007 and 2008, I wrote an app for Apple iOS called “ICDL on iPhone” which provided a scrollable and zoomable interface for reading children's picture books from my International Children's Digital Library on iPhones. At the time, this was the only effort of its kind to provide the ability read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

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. The 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

Ph.D. degrees in Computer Science in 1989 and 1992, both from New York University.⁷

III. SCOPE OF OPINIONS

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

- Claims 1-33 of the '245 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);

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 1-33 of the '245 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 '245 patent, the '245 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.

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

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 '245 patent would have had a bachelor's degree in electrical or computer engineering, or a closely related scientific field such as computer science, and at least 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 1-33 were already known in the prior art before the priority date of the '245 patent. Accordingly, I conclude that there is nothing novel or non-obvious about the alleged invention disclosed in the '245 patent.

45. The '245 Patent is directed to well-known methods and systems for “rendering applications on a wireless device.” Ex-1001, 1:15-16.

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 '245 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:

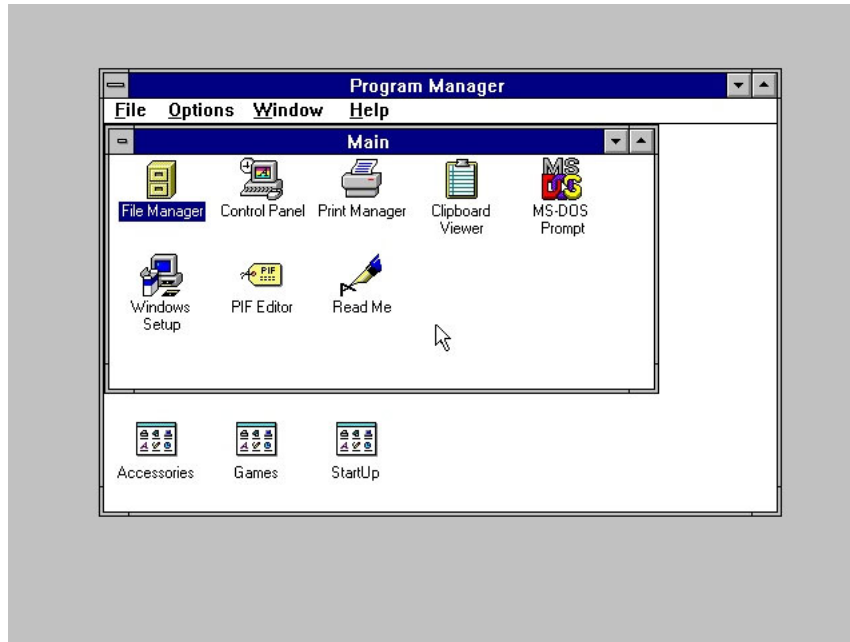


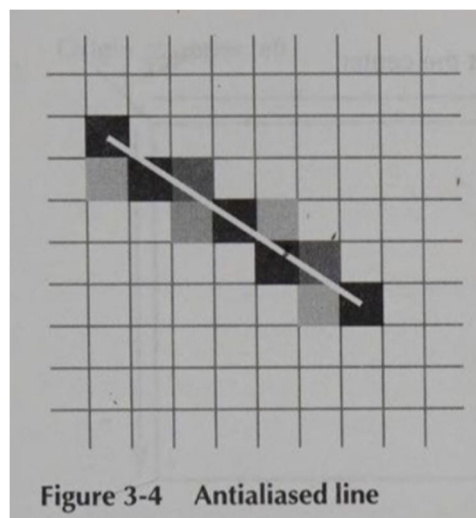
Figure A. Screenshot of Microsoft Windows 3.1 taken by Benjamin B. Bederson on March 28, 2023 using an emulator running in VMWare Fusion.

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

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

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 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. 3-5

⁹ (EX1014) 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. 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, those clients had greater computation capability, and in the early 1980s with the

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

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 “provides network transparency so that an application displaying a chart on a screen

¹¹ (EX1015) 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*”)

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 Article* at 76-78. Web servers used the Hypertext Transfer Protocol (HTTP), a simple request/response network protocol, which allows an HTTP connection to be made from a client to a host.

¹²(EX1016) 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>

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

¹³ See *Berners-Lee* at 76-78.

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 “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

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

¹⁵ see *Kasten* at 3

Foodocument

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.

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

66. 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

¹⁶ See *Kasten* at 1-3.

device configuration system that allows modification of mobile device user interfaces or application programs without modification of the application programs themselves.” *Hariki*, ¶9.

67. *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 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* further explains that a mobile device 108 receives content data, application programs, diagnostic tools, program components, or any other content or executable objects from “a server computer 102 operated by the service provider or a third party vendor.” *Id.*, ¶21. The mobile devices receives HTML pages from a server executing a web server process. *Id.* Additionally, the mobile device receives “customized user interface [generated] for a plurality of different makes and types of mobile devices” by a server that “allow[s] 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. The customized “look and feel” is received by the mobile device 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.

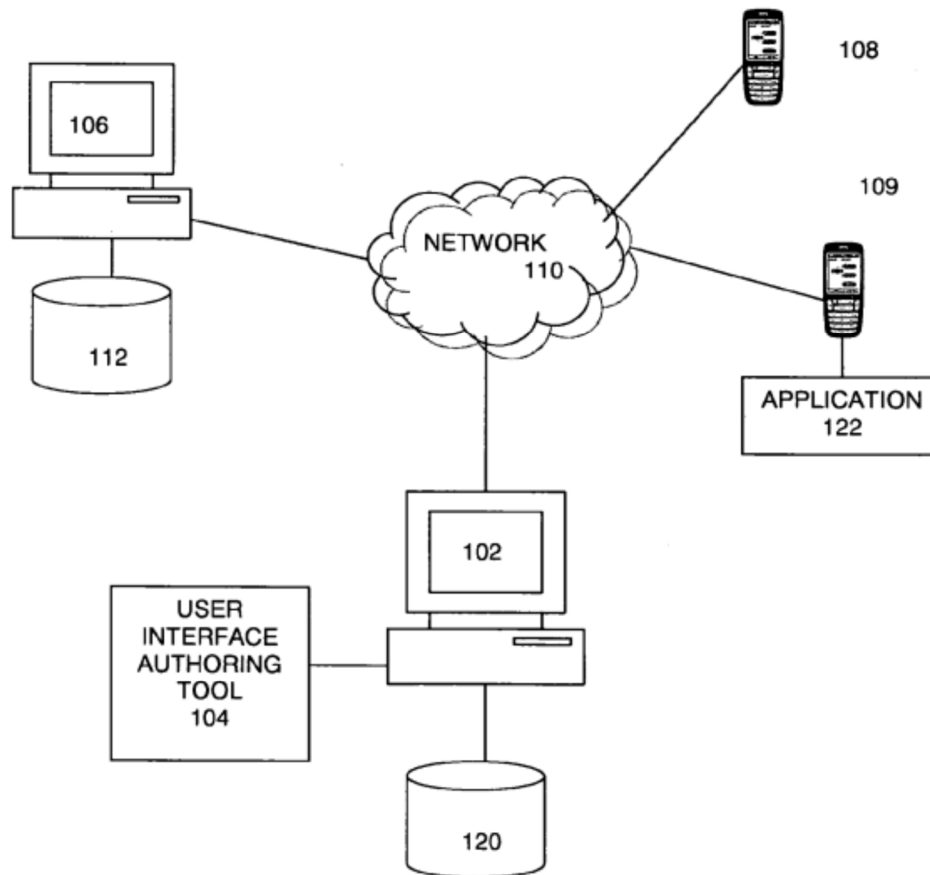


FIGURE 1

Hariki, FIG. 1

68. *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*.

69. 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.

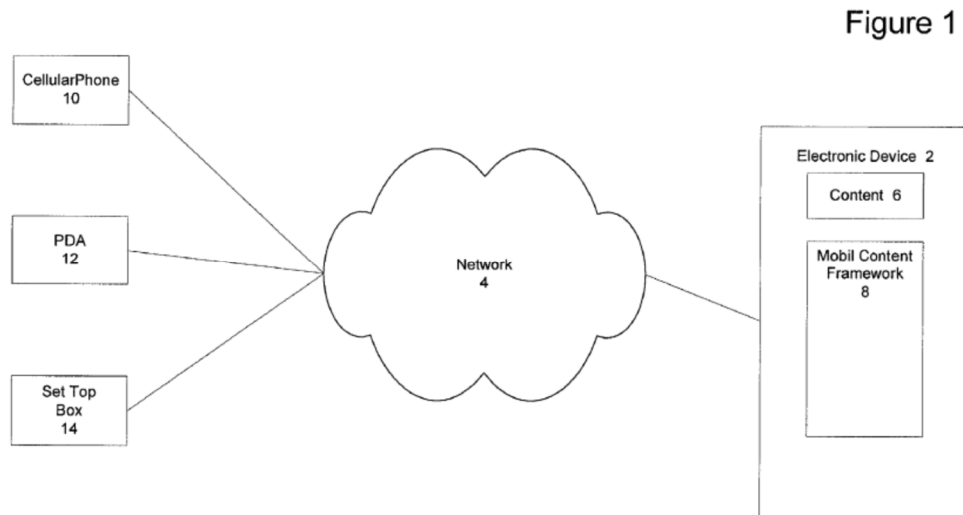
E. Overview of U.S. Patent Application Publication No. 2003/0023755 to Harris et al., filed on December 18, 2001 (“Harris”).

70. *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* at 2.

71. *Harris* discloses a content delivery system where mobile devices request content from a platform, which “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 based on the resource availability of the requesting devices, “[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 requesting mobile device’s “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.*

72. 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.*

73. *Harris* discloses its environment with reference to Figure 1, reproduced below. The environment includes a plurality of mobile devices (cellular phone 10, PDA 12) interfaced through a network to a server (electronic device 2) with content and an MCF. *Id.*, ¶15.



Harris, FIG. 1

74. *Harris* further explains, with reference to Figure 3, below, that a wireless device requests content from the sever (step 60). 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, the requesting device receives the customized content (step 76).”
Id.

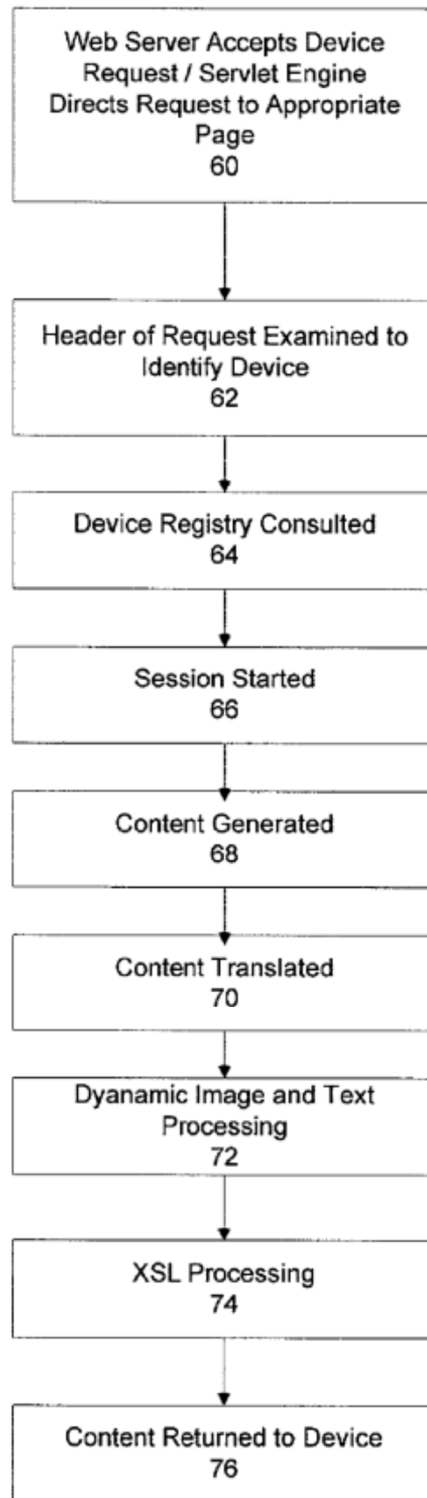


Figure 3

F. Other Evidence Regarding the State of the Art

75. Exhibits 1007-1019; 1022-1023 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 '245 patent and my invalidity analysis.

VIII. SUMMARY OF THE '245 PATENT**A. Background of the Patented Subject Matter**

76. The '245 patent is directed to “rendering applications on a wireless device.” Exhibit 1001 at 1:16.

77. The process of downloading content from a remote device through a network was well-known before 2007. By 2000, following the dot-com boom of the late 1990s, the popularity of the Internet increased, and users were turning from traditional media sources, such as television and newspaper, to the Internet to receive media content. Ex. 1007, 1:12-16. A user may go to media servers containing information from various content providers. *Id.*, 1:16-19. For example, techniques for transmitting/receiving data via a cellular phone were developed well before 2003, so that users can experience content with a large amount of data, such as video and music, on their cellular phones. Ex. 1008, 1:17-20.

78. Those in the art recognized that, by 2003, servers could provide to a remote client automatic delivery of multimedia content, such as entertainment

content. Ex. 1007, 1:29-34; Ex. 1009, Abstract; see also Ex. 1010, Abstract. Prior to the '245 patent, it was a known problem that transmission of content was limited by, for example, bandwidth and connection rates of mobile devices. Ex. 1011, 1:42-57. A common goal was to improve network efficiency from a transmitter to a terminal. Id. Further, deleting content from a device that was expired was well-known and common sense. *See, e.g.*, Ex. 1008, 5:8-15.¶¶.

79. Client/server architectures, wireless devices of myriad sizes and computing power were known, as were rendering technologies and methods of delivering content to diverse clients. In line with the proliferation of wireless devices, POSITA's had already recognized the need to render applications on them and developed ways to do so that took in to account the diversity of receiving devices. For example, in 2006, I co-founded Zumobi (initially named ZenZui), a company that designed and built mobile software with a focus on interactive graphics. As our website explained as of April 2007,¹⁷ the company provided an innovative GUI that enabled users to “fly in and out of your Zoomspace... Through some clever engineering, we constantly refresh your Tiles in the background, so that they're always fresh, available, and ready to be Zoomed.” It goes on to explain that the system's Tiles “work on a wide array of mobile devices and platforms, all with

¹⁷(EX1023)<https://web.archive.org/web/20070406171650/http://www.zenzui.com/products.html>

a consistent, intuitive, and gorgeous user interface.” Our product worked on Microsoft, Blackberry, and Symbian – which all provided the ability for us to render rich graphics on commercial phones with “Tiles” that were downloaded over a wireless network using data that updated the Tiles in the background.

A. Wireless Mobile Devices

80. A common example of a wireless mobile device, are mobile phones that have been around since the 1980s. During the 1990s, great improvements were made in mobile phone technology that made mobile phones smaller and more energy-efficient. In the 2000s, 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. EX1013. PDAs (Personal Digital Assistants) were another common kind of wireless device available in the early 2000s.

B. Graphical User Interfaces (GUIs) and UI Rendering

81. In computing, no matter what the device is, a user interface is the boundary where interactions between humans and computers take place. 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 had its advantages, for instance, it was simple and efficient, but it suffered from several drawbacks. For one thing, the interface was difficult for users to learn, because to operate the computer, they first had to learn all the commands.

82. 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 use the user interface by simply experimenting with the interface, as they would be unlikely to guess such commands. The only real way for a user to master the system was by committing a long list of these unintuitive commands.

83. An advancement to the command-line interface was the Graphical User Interface (“GUI”). GUIs overcome these drawbacks because they 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 3.1¹⁸, released by Microsoft in 1992, was one example of such a GUI:

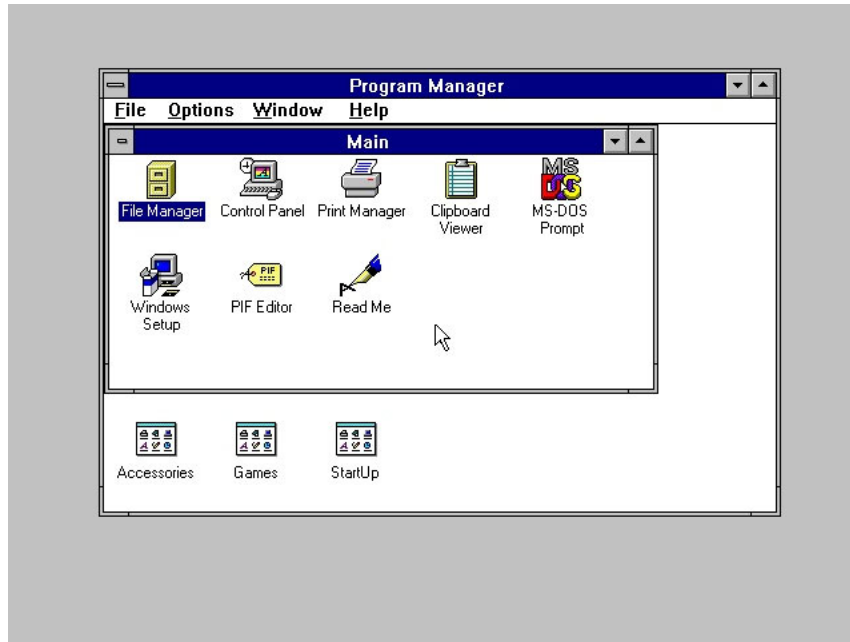


Figure A. Screenshot of Microsoft Windows 3.1 taken by Benjamin B. Bederson on March 28, 2023 using an emulator running in VMWare Fusion

84. Seeking to improve on and cut down the learning curve involved with command line interface, 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. With GUIs there is no longer a need to memorize the unintuitive and arbitrary commands involved in the command line interface, user only need

¹⁸See, e.g., <https://www.computerhistory.org/tdih/april/6/> (last accessed March 29, 2023) (EX1012).

analogize to intuitive real world actions. With GUIs, a user **could** learn their way around the UI interface by experimenting and finding their way around the interface.

85. 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” describes many of the technical elements of this rendering. Dan R. Olsen, Jr., *Developing user interfaces* (1998) (EX1014). 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.

86. By 2007, such rendering technologies were fairly standardized and built into essentially every computer operating system that supported GUIs, including desktop and mobile devices such as cell phones. While many systems rendered 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 graphics commands to a client computer to be rendered.

C. Client-Server Architectures

87. 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, 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.

88. 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-server computing was described in a 1992 article in Communications of the ACM – a broad interest magazine of the main professional international computer society. See EX1015 (“Sinha”). Sinha characterized some of the key elements of client-side 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. On the server-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. 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 “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.*, 91.

89. 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. *See* EX1016.

90. 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* EX1016, 76-78. Web servers used the Hypertext Transfer Protocol

(HTTP), a simple request/response network protocol that was developed by Berners-Lee, which allows an HTTP connection to be made from a client to a host.

91. 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. 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, such as audio files, 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 additional documents 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. *See* EX1016, 76-78.

92. 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.” *See* EX1017. When this HTML is interpreted by a web browser, it would be displayed as following:

Here is some **bold** text.

93. 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 “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 (*see* EX1017, 3):

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

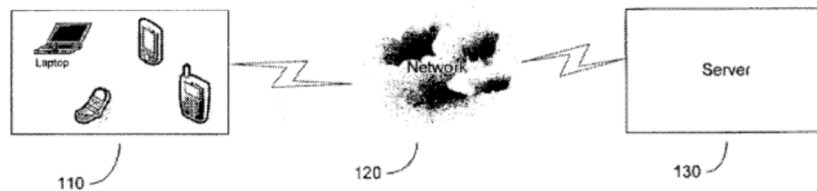
94. As shown in these examples, HTML provides a syntax that is generic and can be used to define applications in a way that can work on client computer. In general, web sites are collections of HTML-based web pages that are stored or generated on a web server that is typically accessed by a single domain 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. *See* EX1017, 1-3. 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.

B. The '245 Patent Specification

95. The '245 patent is directed to “[a] method of rendering content on a wireless device.” *'245Pat*, 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



'245Pat, FIG. 1.

96. 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. The 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

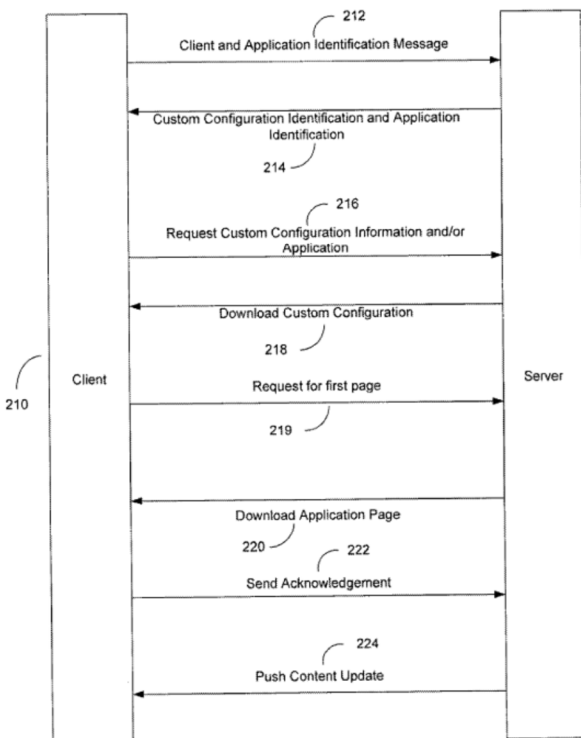


FIGURE 2A

200B

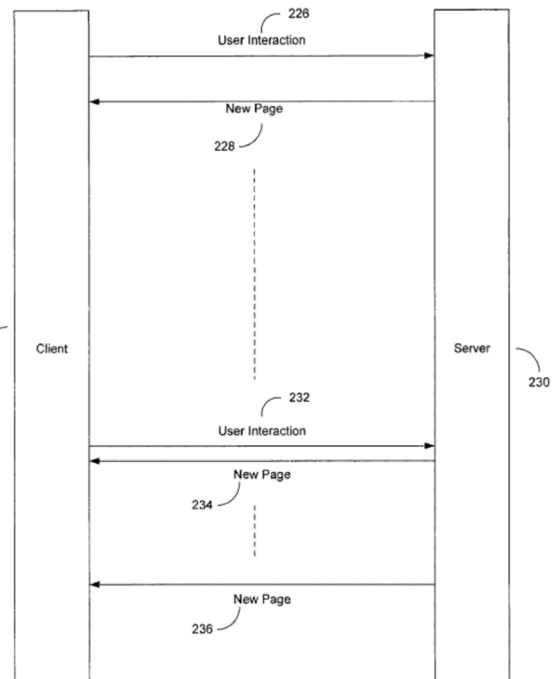


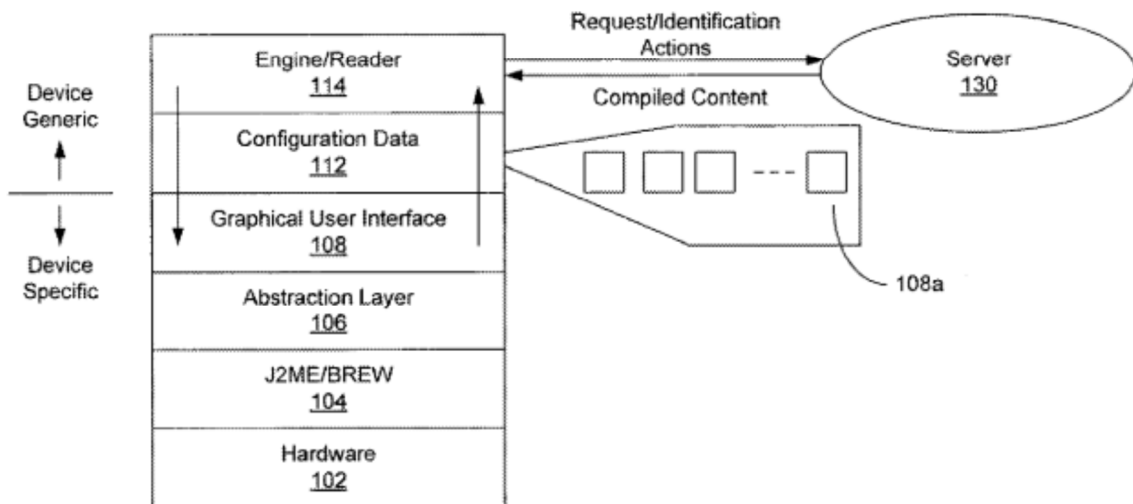
FIGURE 2B

(p) '245Pat, FIGs. 2A & 2B

The 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



'245Pat, FIG. 1B

97. 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.

C. The ’245 Patent File History (Ex. 1004)

98. The ’245 patent issued on July 2, 2013, from U.S. Patent Application 11/888,803 (“the ’803 application”), filed August 1, 2007. The application that lead to the ’245 patent was prosecuted under the pre-AIA statutory regime. ’245FH, 206.

99. On December 13, 2012, Applicant amended the independent claims to recite “wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content.” *Id.*, 189. Applicant also changed “computer readable media” to “non-transitory computer usable medium” in eleven claims. These amendments overcame a previous non-final rejection mailed by the USPTO on June 14, 2012. ’245FH, 34-38. On December 31, 2012, following the amendments, the claims were allowed.

D. Challenged Claims

100. The '245 patent has three independent claims, claims 1, 12, and 23, and seventeen dependent claims, claims 2-11, 13-22, and 19-20. I have been asked to opine on the validity of all claims 1-33, which are listed below:

Claim 1	
[1pre]	A method of rendering content on a wireless device, said method comprising:
1[a]	receiving an identification of a custom configuration of a plurality of rendering blocks of said wireless device,
1[b]	wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application;
[1c]	receiving compiled content generated in part from execution of said application
[1d]	wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device;
[1e]	using a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration; and
[1f]	rendering said renderable content on said wireless device, wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application
[1g]	wherein said custom configuration is applicable to both said first and second compiled content.
Claim 2	
[2]	A method as described in claim 1 wherein said using a graphical user interface comprising said plurality of rendering blocks to generate

	renderable content comprises: processing said compiled content using a reader of said wireless device; and issuing commands from said reader to individual rendering blocks of said graphical user interface based on said rendering commands of said compiled content.
Claim 3	
[3]	A method as described in claim 1 wherein said render able content comprises audio content and display content.
Claim 4	
[4]	A method as described in claim 1 wherein said compiled content is partially resultant from said application operating on a remote server.
Claim 5	
[5]	A method as described in claim 1 wherein said compiled content is specific to the rendering capabilities of said wireless device.
Claim 6	
[6]	A method as described in claim 1 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.
Claim 7	
[7]	A method as described in claim 6 wherein said custom configuration comprises a syntax that is generic regarding said wireless device type.
Claim 8	
[8]	A method as described in claim 1 wherein said custom configuration comprises configuration information and content specific to said application.
Claim 9	
[9]	A method as described in claim 1 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored by said wireless device and wherein further said identifying said custom

	configuration comprises receiving an identifier that identifies said custom configuration.
Claim 10	
[10]	A method as described in claim 1 further comprising receiving and storing said custom configuration.
Claim 11	
[11]	A method as described in claim 1 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across a display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.
Claim 12	
[12pre]	A non-transitory computer usable medium comprising instructions therein that when executed by a processor implement a method of rendering content on a wireless device, said method comprising:
[12a]	receiving an identification of a custom configuration of a plurality of rendering blocks of said wireless device,
[12b]	wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application;
[12c]	receiving compiled content generated in part from execution of said application
[12d]	wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device;
[12e]	using a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration; and
[12f]	rendering said renderable content on said wireless device, wherein said receiving compiled content comprises: receiving first compiled content

	specific to a first page of said application; and receiving second compiled content specific to a second page of said application,
[12g]	wherein said custom configuration is applicable to both said first and second compiled content.
Claim 13	
[13]	A non-transitory computer usable medium as described in claim 12 wherein said using a graphical user interface comprising said plurality of rendering blocks to generate renderable content comprises: processing said compiled content using a reader of said wireless device; and issuing commands from said reader to individual rendering blocks of said graphical user interface based on said rendering commands of said compiled content.
Claim 14	
[14]	A non-transitory computer usable medium as described in claim 12 wherein said renderable content comprises audio content and display content.
Claim 15	
[15]	A non-transitory computer usable medium as described in claim 12 wherein said compiled content is partially result ant from said application operating on a remote server.
Claim 16	
[16]	A non-transitory computer usable medium as described in claim 12 wherein said compiled content is specific to the rendering capabilities of said wireless device.
Claim 17	
[17]	A non-transitory computer usable medium as described in claim 12 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.
Claim 18	

[18]	A non-transitory computer usable medium as described in claim 17 wherein said custom configuration comprises a Syntax that is generic regarding said wireless device type.
Claim 19	
[19]	A non-transitory computer usable medium as described in claim 12 wherein said custom configuration comprises configuration information and content specific to said application.
Claim 20	
[20]	A non-transitory computer usable medium as described in claim 12 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored by said wireless device and wherein further said identifying said custom configuration comprises receiving an identifier that identifies said custom configuration.
Claim 21	
[21]	A non-transitory computer usable medium as described in claim 12 wherein said method further comprises receiving and storing said custom configuration.
Claim 22	
[22]	A non-transitory computer usable medium as described in claim 12 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across a display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.
Claim 23	
[23pre]	A wireless device operable to communicate with a remote server, said wireless device comprising:
[23a]	a transceiver coupled to a bus and operable to receive a custom configuration that is associated with an application,
[23b]	said transceiver also operable to receive compiled content generated in part from execution of said application and

[23c]	comprising a plurality of rendering commands expressed in a syntax that is generic to said wireless device;
[23d]	a memory coupled to said bus and operable to store said compiled content and said custom configuration;
[23e]	a processor coupled to said bus and operable to implement the following: a graphical user interface comprising a plurality of rendering blocks and operable to generate renderable content based on said compiled content and said custom configuration
[23f]	wherein said custom configuration is operable to configure said plurality of rendering blocks to render content in a manner customized to said application;
[23g]	an engine for reading said compiled content and responsive thereto for causing said graphical user interface to generate said renderable content based on said render commands;
[23h]	a display device coupled to said bus and operable to render a portion of said renderable content, wherein said compiled content comprises: a first compiled content specific to a first page of said application; and a second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content.
Claim 24	
[24]	The wireless device as described in claim 23 wherein said engine is further operable to issue commands to individual rendering blocks of said graphical user interface based on said plurality of rendering commands of said compiled content.
Claim 25	
[25]	The wireless device as described in claim 23 further comprising an audio rendering device coupled to said bus and wherein said renderable content comprises audio content and display content.
Claim 26	
[26]	The wireless device as described in claim 23 wherein 25 said compiled content is partially resultant from said application operating on a remote server.
Claim 27	

[27]	The wireless device as described in claim 23 wherein said compiled content is specific to the rendering capabilities of said wireless device.
Claim 28	
[28]	The wireless device as described in claim 23 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.
Claim 29	
[29]	The wireless device as described in claim 28 wherein said custom configuration is expressed in a syntax that is generic to said wireless device type.
Claim 30	
[30]	The wireless device as described in claim 23 wherein said custom configuration comprises configuration information and content specific to said application.
Claim 31	
[31]	The wireless device as described in claim 23 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored in said memory.
Claim 32	
[32]	The wireless device as described in claim 23 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across said display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.
Claim 33	
[33]	The wireless device as described in claim 23 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of a scroll bar on said display screen of said

	wireless device; a second block that controls the rendering of an input box on said display screen of said wireless device; and a third block that control the rendering of a text display region on said display screen of said wireless device.
--	---

E. 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 '245 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 '245 patent.

102. I understand that there are no other litigations that involve or had involved the '245 patent, and none of the terms in the '245 patent have been construed.

IX. PRIORITY DATE OF THE '245 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 '245 patent, filed on July 2, 2013, claims priority to US Patent Application No. 11/888,803, filed on Aug. 1, 2007. The references I have used in Ground 1 are prior art to the August 1st priority date.

X. APPLICATION OF THE PRIOR ART TO CLAIMS 1–33

A. Ground 1: *Hariki* in view of *Harris* renders obvious Claims 1-33

- 1. Claim 1 A method of generating content that is renderable by a wireless device, said method comprising: transmitting, to said wireless device, an identification of a custom configuration of a plurality of rendering blocks of said wireless device, wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application; and transmitting, to said wireless device, compiled content comprising (i) first compiled content specific to a first page of said application and (ii) second compiled content specific to a second page of said application, wherein said compiled content is generated in part from execution of said application, wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device, and wherein said custom configuration is applicable to said first and second compiled content, wherein said compiled content and said custom configuration are usable by a graphical user interface comprising said plurality of rendering**

blocks to generate renderable content based on said compiled content and said custom configuration.

a. [1pre] A method of rendering content on a wireless device, said method 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,” such as HTML files over a network to the mobile devices. *Hariki*, ¶21. The mobile devices “run[] a web browser program to access the web pages served by server computer 102 and any other available content provider” *Id.* Additionally, 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.* The UI skin for a target mobile device is contained in a “UI content package”. *Id.*, ¶33. Furthermore, the “UI content package

can [] be downloaded to the target mobile device” *Id.*, ¶36. Additionally, “the mobile device ... is configured to run one or more application programs 122 or software modules to execute or access the resources provided by the UI content package”. *Id.*, ¶40. Thus, the server in *Hariki* “can provide a customized UI that allows the user to change certain aspects of the UI skin.” *Id.*, ¶26.

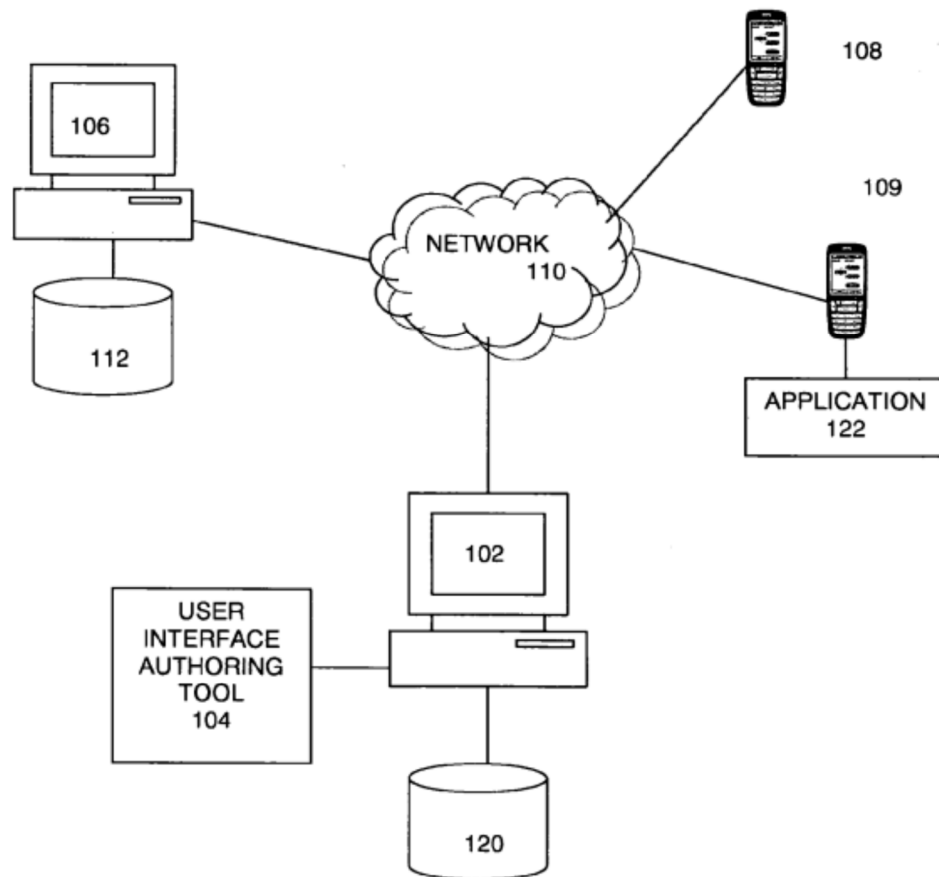


FIGURE 1

Hariki, FIG. 1

108. The '245 patent does not disclose what “content that is renderable” is, however, it does acknowledge that “renderable content comprises audio content and display content. *See, e.g., ' 245Pat*, 21:6-7 (Claim 3). A POSITA would have thus understood, based on the specification of the '245 patent, that “renderable content” merely means anything that can eventually be displayed “e.g., rendered) on a mobile device.

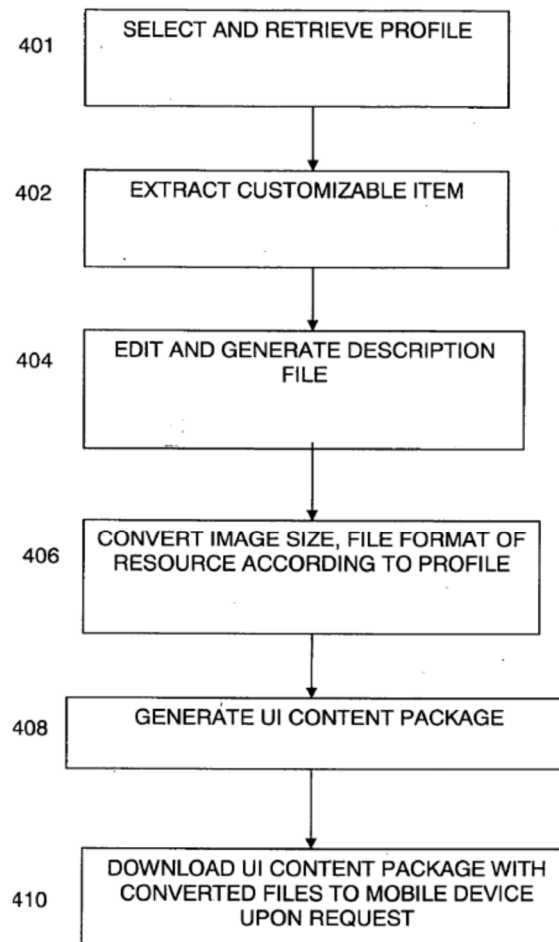
109. The '245 patent functionally describes “renderable content.” More particularly, it discloses that “the wireless device uses a graphical user interface to generate renderable content based on the received compiled content and the custom configuration.” *' 245Pat*, 18:63-65. The specification then explains that “content is rendered by processing the compiled content using the reader and by issuing commands from the reader to rendering blocks of the graphical user interface based on the rendering commands of the compiled content.” *Id.*, 18-66-19:2.

110. A POSITA would have understood that the mobile device in *Hariki* renders content, because the mobile device in *Hariki* downloads and runs executable content such as a UI skin that alters the “look and feel” of its display and/or sound, and comprises audio content and display content.

111. *Hariki* discloses that mobile devices 108 and 109 may be wide range of devices, including 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.”). A POSITA would have thus understood that the mobile devices are wireless devices.

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

**FIGURE 4***Hariki*, FIG. 4

113. It is therefore my opinion *Hariki* discloses a method of rendering a UI skin to change the look and feel of display or sound (corresponding to the claimed “method of rendering content”) of a mobile device (corresponding to the claimed “wireless device”).

- b. 1[a] receiving an identification of a custom configuration of a plurality of rendering blocks of said wireless device,

114. It is my opinion that *Hariki* also discloses this claim feature. As explained above, the mobile device receives HTML files and customized UI skin (e.g., UI package) that alters its look and feel. *See* Section X.A.1.b. *Hariki* discloses that this alteration is based on receiving at the mobile device a downloadable UI content package consisting of [a] description file and [] converted resource files.” *Hariki*, ¶19. As explained below, a POSITA would have understood that the converted resource files and the description file in *Hariki*’s UI content package¹⁹ corresponded to the claimed “custom configuration” and “identification of [the] custom configuration,” respectively.

115. *Hariki* explains that a specific content packages that is downloaded at a requesting mobile device is sent by a “server computer 102 [that] is operated by a content provider, and executes a user interface authoring tool 104 that generates [that] content package for each mobile device.” *Hariki*, ¶25. The user interface authoring tool “can represent a program or **suite of programs**, or even hardware

¹⁹ *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). These terms all refer to the same kind of UI content package, “The UI authoring system allows content providers to create content packages embodying UI skins for distribution and downloading to mobile devices on the network.” *Hariki* ¶28; *see also* Fig. 2. A POSITA would have understood that these refer to the same custom configuration and identification system. For consistency, I will use UI content package throughout.

circuits, or any combination thereof **embodying instructions** executed by one or more processing units in server 102.” *Id.*, ¶27.

116. *Hariki* explains that the UI skin may consist “of a particular screen configuration, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection.” *Id.*, ¶26. *Hariki* also 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.* Furthermore, *Hariki* explains that the UI skin can configure “backgrounds, title bars, **buttons, alert sounds**, and so on.” *Id.*, ¶24.

117. Accordingly, a POSITA would have understood that these backgrounds, title bars, buttons, and alert sounds that are configurable within the UI skin, represents configuration data programmed into “rendering blocks,” as described in the ’245 patent. Indeed, the ’245 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.

'245*Pat*, 8:27-43.

118. A POSITA would have thus understood that *Hariki's* configurable buttons and sounds, for example, fall squarely within the '245 patent's exemplary list of "rendering blocks."

119. When received by the mobile device, the "[u]ser interface descriptor information comprising the customizable UI elements are downloaded."²⁰ *Hariki*,

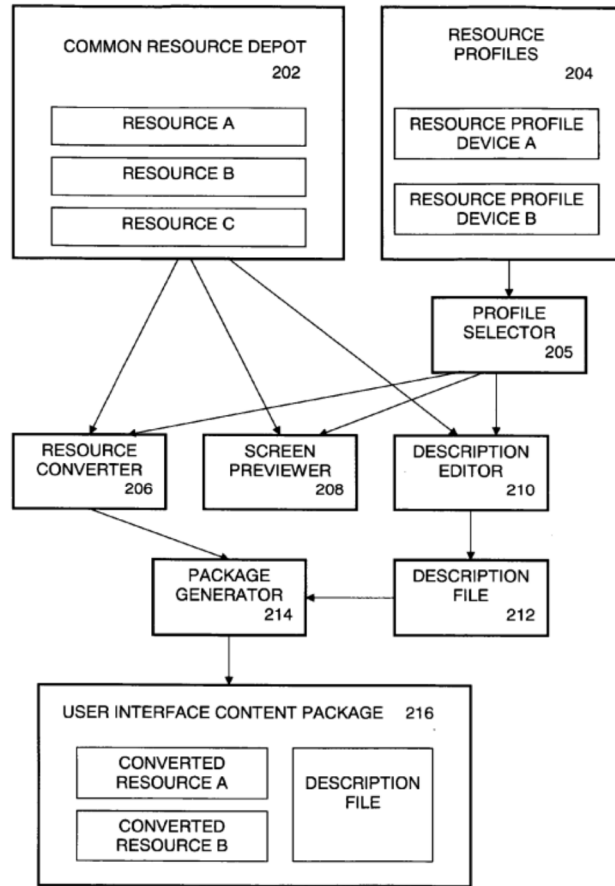
¶27. For example, *Hariki* discloses that the UI content package:

can comprise screen parameter definitions (e.g., size, aspect ratio, **icon definitions**, and so on), **images, video clips, music or other sound clips**, ringtones, games, small applications (applets), utilities, diagnostic tools, or any other similar data or applications, all of which are referred to as "UI content objects."

Id.

120. The UI content can either be downloaded by the mobile device on demand. *Id.*

²⁰ Here *Hariki* uses the term "UI descriptor information" interchangeably with UI content package.

**FIGURE 2***Hariki*, FIG. 2

121. With reference to Figure 2, above, *Hariki* explains that “UI skin package data specific to each mobile device is generated from common resource data using resource profile information for each mobile device model.” *Id.*, ¶29. *Hariki* further explains that “description editor component 210 produces description files 212 based on the selected resource profile 204 and resource file 202” (*id.*, ¶33) and “resource converter 206 converts each resource into a format corresponding to the resource profile” (*id.*, ¶32). Furthermore, “**description file 212 specifies the**

resources and the file type, and file path name.” *Id.*, ¶33. In other words, “[t]he description file contains references to one or more [converted] resources.” *Id.*, ¶46. “The UI content package comprises the appropriate converted resources and the description file.” *Id.*, ¶33.

122. As explained above, the configurable aspects of the user interface may be configurable buttons and sounds (i.e., “rendering blocks”). *Id.*, ¶24. *Hariki* explains that the 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. A POSITA would have thus understood the converted resources in the UI package configure the configurable aspects of the UI. Therefore, a POSITA would have understood the converted resources as a “custom configuration” of the configurable buttons and sounds (i.e., “plurality of rendering blocks”).

123. *Hariki* also explains that the description file “may reference resources contained within the same UI content package, or it may reference resources in other content packages that are either in pre-installed in the mobile handset or on an external server computer.” *Hariki*, ¶46. *Hariki* additionally discloses that a mobile device receives the UI content package containing the description file from a content server. *Id.*, ¶45. The description file received by the mobile device identifies the resources with “certain items of information, such as identifier (ID), path (location),

and file type, among other parameters, relating to a particular resource.” *Id.*, ¶48. The mobile device then uses the received description file to download resource by “request[ing] a resource by specifying the resource ID (e.g., ID 1).” *Id.*, ¶50. A POSITA would have understood that the description file received by the mobile device identifies the file resources (i.e., “custom configuration”) in the UI skin.

124. Therefore, *Hariki* discloses a mobile device receiving (corresponding the claimed “receiving” at “wireless device”) a UI package including converted resources (corresponding to the claimed “custom configuration”) a description file (corresponding to the claimed “identification”), wherein the UI skin package data configures the look and feel of display and sound aspects of the mobile device (corresponding to the claimed “rendering blocks”).

c. **1[b] wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application;**

125. In my opinion, *Hariki* discloses this claim limitation as well. For example, as explained above, *Hariki* discloses converted resources (“custom configuration”) in a UI content package that alter the look and feel of aspects of the display and sound of the mobile device, such as backgrounds and sounds (“configures said plurality of rendering blocks”). *See* Section X.A.1.a. *Hariki* discloses that the UI skin allow customization of the look and feel of the “application

program environment of a device.” *Hariki*, ¶24. The customization is effected by the UI content package. *Id.*, ¶27.

126. *Hariki* further explains that “the resources referenced by the descriptor file and **utilized by the application** must be referenced in some way by the application and then retrieved and converted to an appropriate format **for use by the application.**” *Hariki*, ¶39. The application may be “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.*, ¶21. Accordingly, a POSITA would have understood that the converted resources (i.e., “custom configuration”) in the UI content package are associated with the web browser (i.e., “an application”).

127. To the extent Patent Owner contends that the “associated” application must be remoteto the mobile device, *Hariki* discloses this as well. *Hariki* discloses “the resources comprising the UI content package to be downloaded to the mobile device for use by the application may be provided by an external content provider, as shown in FIG. 1. Alternatively, the UI content package may be pre-installed on the mobile device itself.” *Id.*, ¶38. A POSITA would have understood that the “application” is not limited to residing **only** on a client or only on a server, but rather anywhere, like for instance on a client **or** on a server.

128. For instance, as explained above, *Hariki* discloses that the user interface authoring tool represents a suite of programs on the server and that the server also provides content data such as HTML files through a web server process. *See* Section X.A.1.b. A POSITA would have recognized the user interface authoring tool and web server process were part of the same suite of programs because they reside on the server together and that the authoring tool generated the custom UI for mobile devices accessing the web server process. Thus, a POSITA would have understood that the converted resources (i.e., “custom configuration”) in the UI content package were associated also with the web server process that serves HTML files to the mobile device.

129. Furthermore, *Hariki* explains that the UI content package is specific to the mobile device type and the application programs. *Hariki*, ¶27. A POSITA would therefore have understood that the UI content package is customized to the web server-web browser interaction and consequently customized to those applications.

d. 1[c]: receiving compiled content generated in part from execution of said application,

130. *Hariki* in combination with *Harris* discloses this feature. As explained above, *Hariki* serves HTML files from a web server that are received at a web browser. *Hariki*, ¶21. In addition to serving the HTML file from a web server, *Hariki* expressly discloses that the server involved “executes” the web server process (*id.*, ¶21) and the mobile device “executes” the application, i.e., web browser (*id.*, ¶38).

A POSITA would have readily understood that receiving the subsequent HTML files was due at least in part from executing the web server process and/or the web browser because execution of the applications is required to operate the server-browser system.

131. *Hariki* also discloses that “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.* A POSITA would have understood that HTML files include tags that include low level commands for the client device to execute. A POSITA would also have understood that HTML files sent by a web server are rendered into pages in a web browser on the mobile device. Additionally, a POSITA would have understood that 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.

132. *Harris*, like *Hariki*, also discloses “a platform that enables a content developer to distribute uniform content to multiple types of requesting mobile devices.” *Harris*, Abstract. As explained above in Section VII.E, *Harris* describes a wireless device template independent of mobile device type 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).

133. In *Harris*, the generation of the HTML page begins when the “web server accepts a connection and an HTTP request from a mobile device and the servlet engine directs the request to the appropriate page or servlet destined to generate WAX (step 60).” *Harris*, ¶24. A POSITA would have understood, that just as in *Hariki*, the web server and mobile device are each executing applications that cause the generation of the HTML page (i.e., “compiled content”).

134. Thus, the *Hariki-Harris* system generates HTML (i.e., “compiled content”) by executing web server and mobile device browser applications (i.e., “said application”) and received at a mobile device’s web browser. Therefore, the HTML is received by the wireless device at least in part from executing one or both applications.

e. **1[d]: wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device,**

135. *Hariki* in combination with *Harris* discloses this feature. The ’245 patent explains that the compiled content, i.e., render commands, 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.” ’245*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., 13:26-34.

136. The ’245 patent shows an example of basic commands in a “compiled page description” in Figure 4, reproduced below. The ’245 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.”

Id., 15:59-67.

400

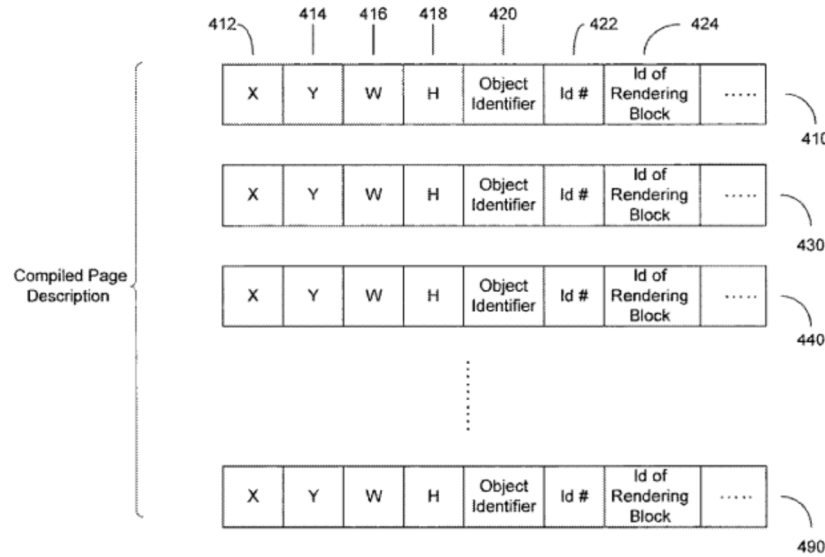


FIGURE 4

'245Pat, FIG. 4

137. In the *Harris-Hariki* system, *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:

```
<?xml version=" 1. 0" encoding="utf-8"?>
<wax:wax xmlns:wax="http://www.kargo.com/wax" version="O.9">
  <wax:doc version="1.0">
    <wax:title>NY Nightlife</wax:title>
    <wax:block id="splash">
      <!--EXAMPLE 1 -->
      <wax : button href="index?a=rnain"
keytype="accept" type="go" labelid="enter" />
      <wax:p align="center">
        <!--EXAMPLE 2 -->
        <wax: img srcid="logo" alt="My Nightlife"
          border="O" />
    </wax:block>
  </wax:doc>
</wax:wax>
```

-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

138. A POSITA would have looked at the above Table and understood 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. *Harris* further elaborates on how the syntax used by the WAX language is both device generic such that devices of different display sizes will have content displayed differently potentially in different languages:

WAX itself does not determine the “physical image’ to display, only the “logical image. Example 3 illustrates the same principal, but with text. The id “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.

Harris ¶30.

139. Accordingly, in the *Hariki-Harris* system the content data received by the mobile device is expressed in a device generic syntax, regardless of whether it is WAX or HTML, both using a syntax that is device independent (i.e., “expressed in a syntax that is generic to said wireless device.”).

f. **[1e] using a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration; and**

140. In my opinion *Hariki* in combination with *Harris* discloses this feature as well. The '245 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.” '245*Pat*, 2:60-64. The '245 patent further states that the graphical user interface rendering blocks “render content and enable a user to interact” with an application, for example by allowing a user to click on a “submit” button. *Id.*, 8:23-26. With respect to Figure 6 of the '245 patent, the '245 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 continues: “In one embodiment, content is rendered by processing the compiled content using the reader and by issuing commands from the reader to rendering blocks of the graphical user interface based on the rendering commands of the compiled content.” *Id.*, 8:65-19:2.

600

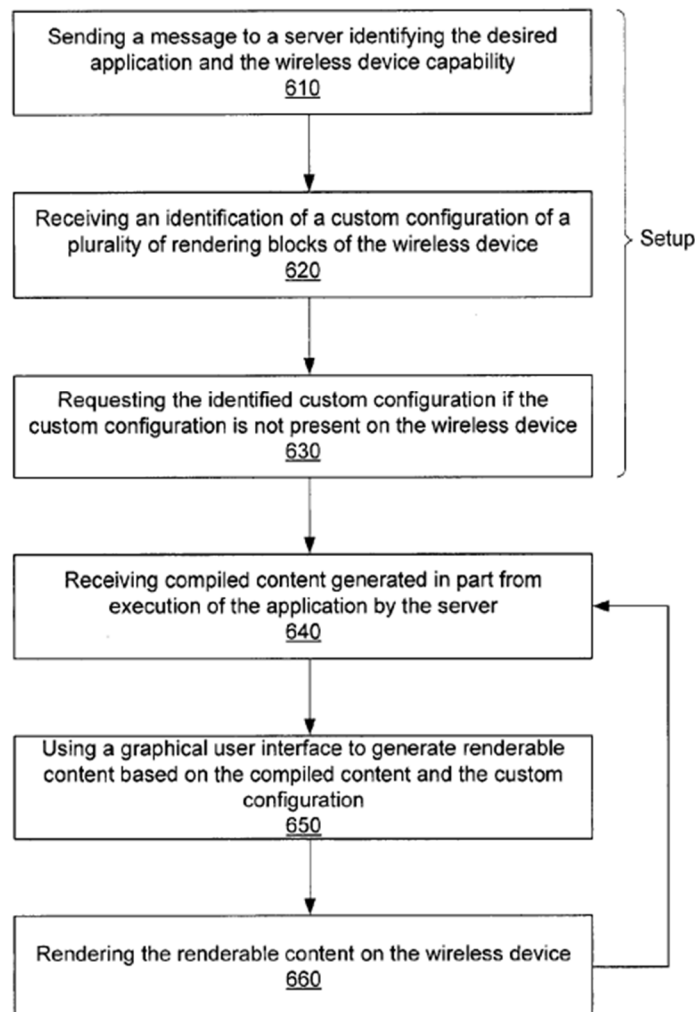


FIGURE 6

'245 patent FIG. 6

141. In the *Hariki-Harris* system, *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.

comprising said plurality of rendering blocks”) that use the UI skin (i.e., “custom configuration”) and render it to a display. For example, with reference to Figure 5, below, *Hariki* discloses a general block diagram of the mobile device. *Hariki*, ¶41.

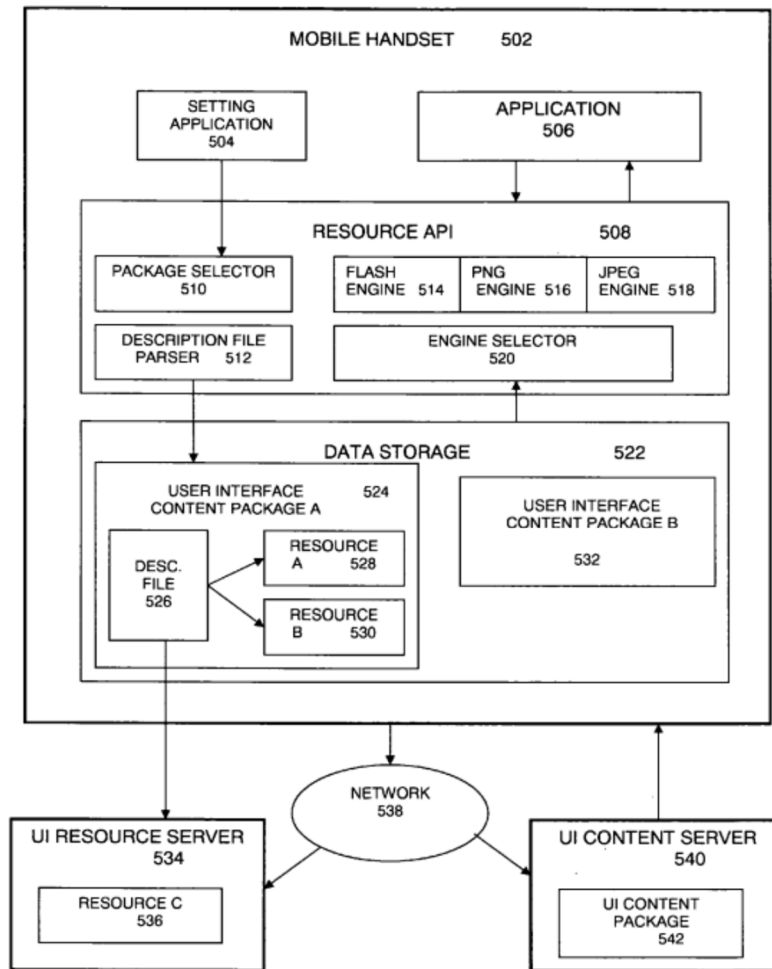


FIGURE 5

Hariki, FIG. 5

142. *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.* The application, e.g., a web browser that, as explained above, receives both HTML files (i.e., “compiled content”) and converted resources (i.e., “custom configuration”) in the UI content package, 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.

143. A POSITA would have thus understood that *Hariki*’s resource API and engines use the converted resources in the UI content 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). A POSITA would have also thus understood that that “look and feel” or application program environment of the device described in *Hariki* was a user interface displayed to the user and corresponded to the “graphical user interface,” as claimed in the ’245 patent. *Id.*

144. To the extent the Patent Owner argues that *Hariki* does not expressly disclose using the engines in the resource API to generate renderable content from the received HTML files, a POSITA would have been motivated to modify *Hariki* to do so. For example, *Hariki* expressly contemplates modifying the resource API to include additional engines for rendering content. *Hariki*, ¶51. Specifically, *Hariki* explains:

elements of the resource API 508 **can be modified or upgraded** without requiring a change to the application program. For example, **different or new versions of engines for different** types of data objects or **programs** can be implemented in the resource API by adding or modifying the engine components.

Id.

145. *Harris* further discloses displaying the content generated for the specific mobile devices. *See, e.g., Harris*, claim 1 (“providing content, said content capable of being displayed to a user of a mobile device interfaced with said network”). *Harris* explains that for HTML documents, “an automated testing environment will make sure content will display[] correctly before they are deployed to a device.” *Id.*, ¶35. *Harris* further acknowledges that it was well-known in the art that mobile device displayed content in different ways. *Id.*, ¶¶3-4. A POSITA would have been motivated by the disclosure of *Harris* to modify *Hariki*’s resource API to include additional engines to convert and render HTML files for display on the mobile device.

146. Accordingly, a POSITA would have been motivated to include engines in the resource API to format HTML files and objects, for example, to provide expanded functionality to *Hariki* mobile device to render and display content data that *Hariki* contemplates receiving at the mobile device. A POSITA would have had a reasonable expectation of success because engines for rendering and displaying HTML files were commonplace and well within a POSITA's knowledge. (citing EX 1018 *Linburn*, ¶39 and Ex 1019 *Talwar*, 12:58-64). Thus, modifying *Hariki*'s resource API to include engines to render and display HTML files is nothing more than applying the known technique of including rendering engines in a mobile device ready for improvement to yield predictable results.

g. **[1f] rendering said renderable content on said wireless device, wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application,**

147. It is my opinion that *Hariki* also discloses this feature. The '245 patent describes "compiled content" as including "a series of basic commands for rendering the requested application." '245Pat, 7:43-45; *see also* 2:46-50 ("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"). As explained above, *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.*

148. A POSITA would have understood that HTML files include tags that include low level commands to render content on a client device. *See* Section VII.A.C. A POSITA would have further understood that HTML files received by a mobile device are rendered into pages in a web browser on the mobile device. Therefore, a POSITA would have understood that 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. Accordingly, a POSITA would have understood that such a compilation of commands is how the ’715 patent describes the compiled content. ’715*Pat*, 17:65-76 (“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.”).

149. Furthermore, a POSITA would have understood that a client receives multiple HTML pages from a server when a client accesses the server (i.e., application) to receive HTML pages. *See* Section VII.A.C. For example, it was common knowledge by 2007 that HTML files included links that would cause

subsequent HTML files to be sent to the client after selecting the link. *See* Section VII.A.C. Accordingly, a POSITA would have understood *Hariki* to disclose or suggest that the mobile device receives multiple (corresponding to the claimed “first” and “second”) HTML files (corresponding to the claimed “compiled content”).

150. Additionally, a POSITA would have understood that HTML pages received by the mobile device would have been specific to the application (e.g., a web browser) executed on the mobile device because that browser application requested the content in the first place.

151. *Harris*, like *Hariki*, discloses “a platform that enables a content developer to distribute uniform content to multiple types of requesting mobile devices.” *Harris*, Abstract. As explained above in Section **Error! Reference source not found.** VII.E, *Harris* describes 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). Thus, *Harris* discloses a detailed process of receiving

HTML files at a client mobile device where the HTML files are tailored to the particular mobile device.

152. A POSITA would have understood that the methods disclosed by *Harris* were compatible with and could be adapted and applied to *Hariki* because *Harris*'s platform is of the same general construction as *Hariki*'s server-client system. Thus, to the extent that the Patent Owner might contend that *Hariki* does not disclose "compiled content," a POSITA would have been motivated to combine the server-client system of *Hariki* with the teachings of *Harris* directed to receiving tailored HTML pages at a particular mobile device. In particular, 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 receiving mobile device and looked to *Harris* because, while *Hariki* discloses receiving HTML files at a particular mobile device that are sent from a server, it does not expressly disclose how the HTML files are generated and customized to the particular receiving mobile device.

153. Additionally, a POSITA would have had a reasonable anticipation of success in implementing *Harris*' content generation scheme with *Hariki*'s customization scheme. For example, 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*. Thus, a POSITA would have been motivated to combine *Hariki* and *Harris* because, while each discloses sending specific content that is received by a mobile device, each describes in detail different aspect of the customization.

154. *Harris* further explains that “[c]ontent is generated specifically for each device, both from a display standpoint and a content navigation standpoint.” *Harris*, ¶7. *Harris* explains that to do this, a generic markup language called WAX is used, which is “easily extended and can be translated into a variety of different mobile device markup languages.” *Id.*, ¶7. Specifically, *Harris* explains that “the framework translates the WAX into a device-specific markup language (WML, HTML, etc.).” *Id.*, ¶27. *Harris* 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. A POSITA would have thus recognized that HTML tags specific to the mobile device (i.e., the link) are present in the HTML files in *Harris*.

155. In this context a POSITA would have understood “commands” as having its plain and ordinary meaning, for example, a directive to a computer program to perform a specific task.

156. *Harris* thus discloses sending such basic commands to render display items, such as a link, that is specific to the application being executed on the mobile device (corresponding to the claimed “compiled content”). The HTML is also specific to the web server application on the server because that application is generating the HTML files in the first instance.

157. *Harris* also contemplates serving multiple pages (e.g., HTML files) to the mobile client. For example, *Harris* explains that the mobile device will make “page requests.” *Harris*, ¶27. *Harris* explains that a caching system may be used such that “subsequent requests for the same page do not require processing.” *Id.* This caching is done for multiple pages. *Id.* Accordingly, a POSITA would have understood *Harris* to send at least first and second pages of HTML files which are received by the mobile device, following multiple requests by the mobile device. Therefore, a POSITA would have also understood that the HTML was specific to the pages served by the web server or consumed by the web browser (i.e., “said application.

h. [1g] wherein said custom configuration is applicable to both said first and second compiled content.

158. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For instance, *Hariki* discloses that “[i]n general, UI skins 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.” *Hariki*, ¶24. *Hariki* further explains, with reference to Figure 5, that “[t]hrough the resource API functionality, the mobile device is able to change the **UI skin of the application 506** by UI content packages that are downloaded from a UI content server or are pre-installed in the mobile handset, without requiring modification of the application itself.” *Id.*, ¶42.

159. A POSITA would have understood that customizing the UI skin, which is the application program environment or display and sound aspects of the device, using the UI content package would mean that the converted resources (i.e., “custom configuration”) of the UI content package were applicable to all HTML files received from the web server and displayed in application.

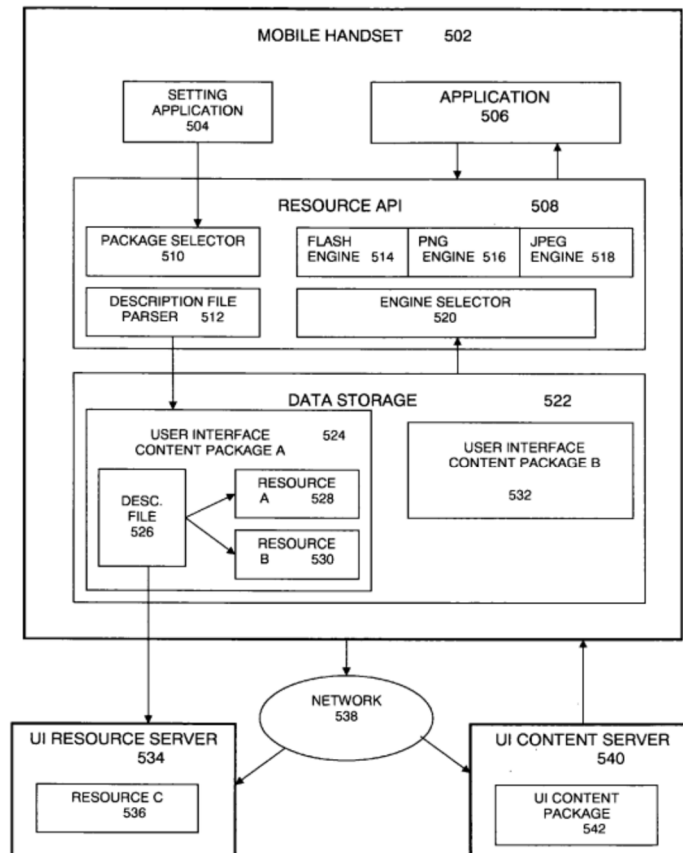


FIGURE 5

Hariki, FIG. 5

2. **Claim 2** “A method as described in claim 1 wherein said using a graphical user interface comprising said plurality of rendering blocks to generate renderable content comprises: processing said compiled content using a reader of said wireless device; and issuing commands from said reader to individual rendering blocks of said graphical user interface based on said rendering commands of said compiled content.”

160. It is my opinion that *Hariki* in combination with Harris discloses claim 2. The '245 patent describes “reader of said wireless device” as an “engine/reader 114 [that] passes commands to [] rendering blocks to perform the rendering of the application content.” '245Pat, 8:24-25. Furthermore, the wireless device

“process[es] the compiled content using the reader. ’245*Pat*, 18:66-67. The reader itself is shown as a generic block 114 in Fig. 1B of the ’245 patent.

100B

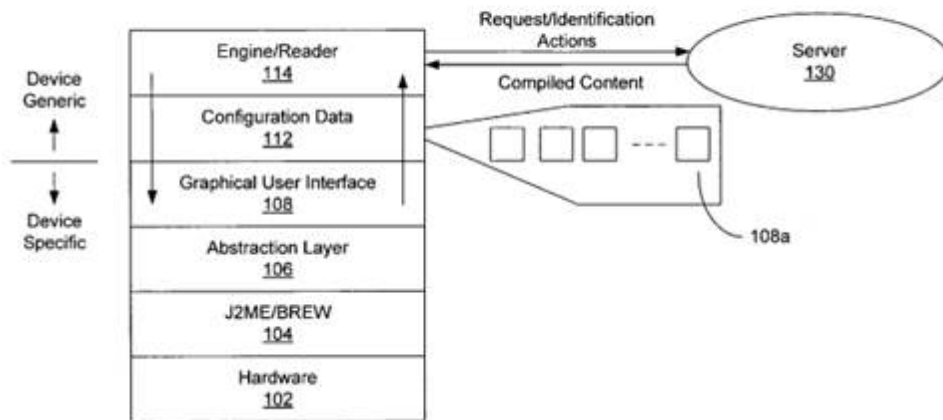


FIGURE 1B

’245 patent, FIG. 1b

161. In the ’245 patent, the wireless device’s reader processes received compiled content, and issues commands to rendering blocks based on the processed compiled content.

162. In the *Hariki-Harris* system, as described in Sections X.A.1.b and X.A.1.f, *Harris* discloses a mobile device’s web browser receiving HTML files (i.e., “compiled content”). *Harris*, ¶24. Furthermore, as shown in *Hariki*’s Figure 5 below,

the mobile device's "resource API contains a number of different plug-in content engines and an engine selector component to allow an application program executed on the mobile handset to access the resources." *Hariki*, ¶19.

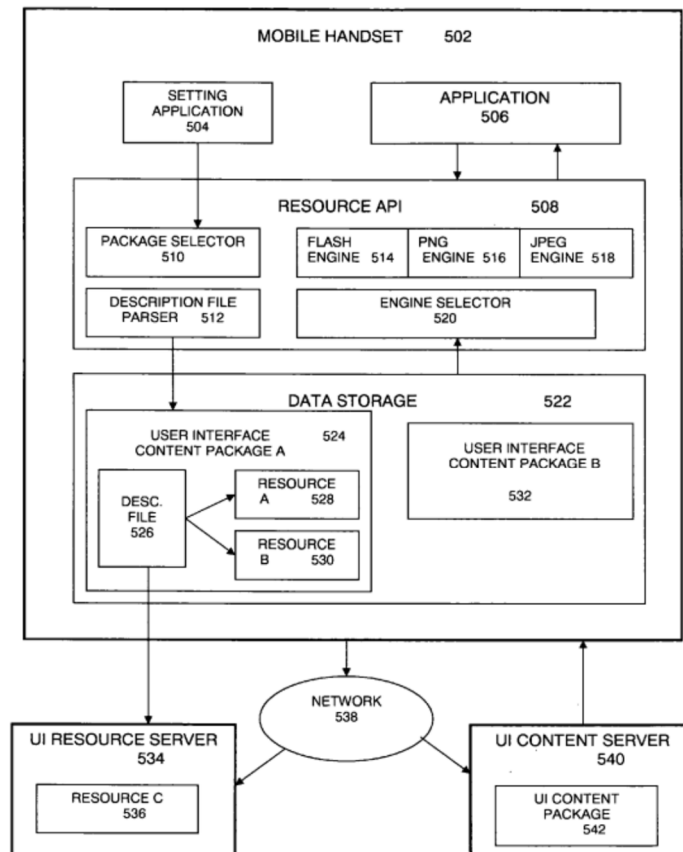


FIGURE 5

Hariki, FIG. 5

163. The resources “comprise 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. The resource API then “uploads all referenced resources to the application.” *Id.*, ¶51. *Hariki*’s engines, in other words, communicates with the files,

links to files, and/or data or program objects. A POSITA would have understood that these files or objects can be *Harris*' HTML-based rendering blocks such as WAX buttons. *Harris*, ¶19. Accordingly, a POSITA would have understood that *Hariki*'s engines (corresponding to the "reader") communicate with (corresponds to "issuing commands to") the files, links to files, and/or data or program objects such as buttons (corresponds to "rendering blocks") based on HTML files (corresponds to "compiled content comprising render commands").

3. Claim 3 "A method as described in claim 1 wherein said renderable content comprises audio content and display content."

164. It is my opinion that *Hariki* in combination with *Harris* claim 3. For example, with respect to the UI content package, *Hariki* discloses that it "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.

165. 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, the *Hariki-Harris* system discloses "renderable content comprises audio content and display content."

4. Claim 4 "A method as described in claim 1 wherein said compiled content is partially resultant from said application operating on a remote server."

166. It is my opinion that *Hariki* in combination with *Harris* discloses claim 4. As explained above in Sections X.A.1.c and X.A.1.f, HTML files (or pages) are sent from a remotely executed web server application when requested by a web browser application executed on a mobile device. As also explained above, the UI skin content package is “associated” with the web server application (*see* Section X.A.1.c) and the HTML files are specific to the web server application (*see* Section X.A.1.f).

167. Since the web browser must request the HTML files (i.e., “compiled content”) from the web server, a POSITA would have understood that the HTML files are generated as a result of both the web browser and web server. Furthermore, *Hariki* discloses that the web server is executed on server 102 and the web browser is executed on client 108 or 109, where the server and client are separated by network 110. *Hariki*, ¶27. Accordingly, the HTML files (i.e., “compiled content”) are “partially resultant” from the web server application running on a server that is not part of the client (i.e., “said application operating on a remote server”).

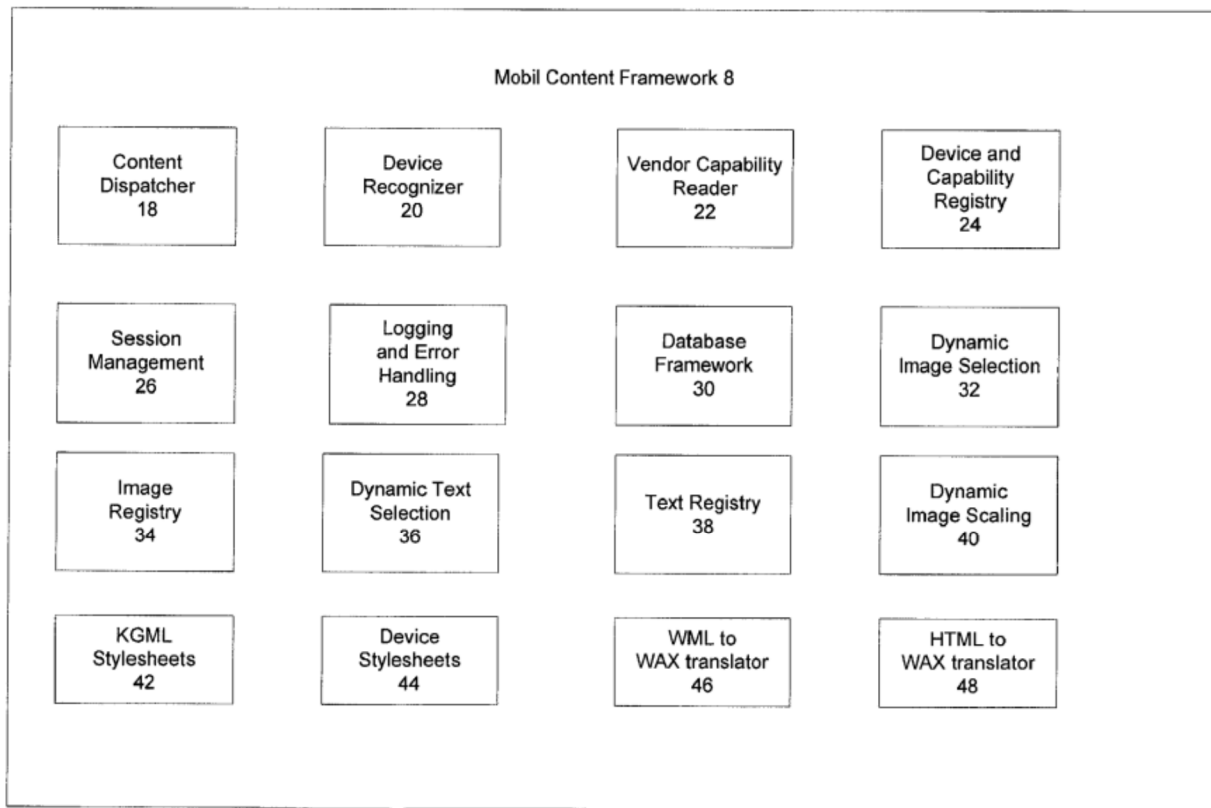
5. Claim 5 “A method as described in claim 1 wherein said compiled content is specific to the rendering capabilities of said wireless device.”

168. It is my opinion that *Hariki* in combination with *Harris* discloses this feature. For example, in the *Hariki-Harris* system, *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. 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 content to specific requesting devices” (*id.*, ¶19).

169. 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.* The registries in the MCF “are utilized to provide appropriate amounts and types of content to the requesting device.” *Id.*, 19.

Figure 2

*Harris*, FIG. 2

170. Accordingly, the *Hariki-Harris* system tailors the HTML pages to the specific attributes (corresponding to the claimed “rendering capabilities”) of the mobile device.

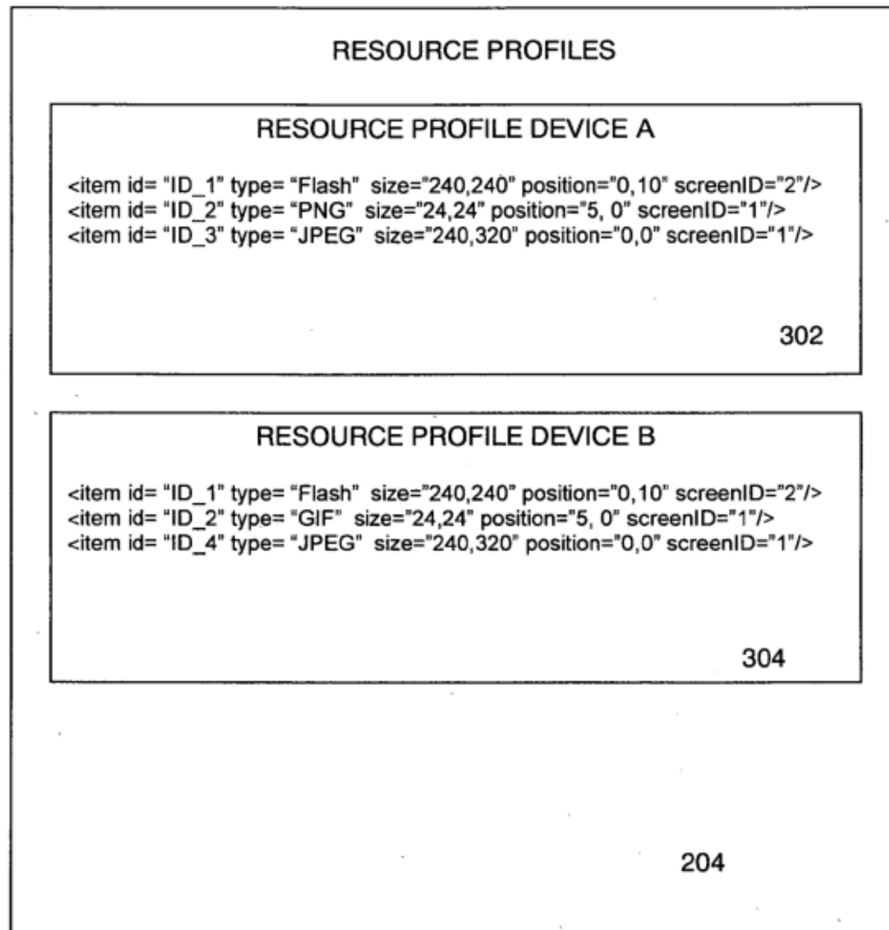
6. **Claim 6 “A method as described in claim 1 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.”**

171. It is my opinion that *Hariki* in combination with *Harris* discloses or claim 6. As explained above in Section X.A.1.b, *Hariki* explains that “UI skin

package data **specific to each mobile device** is generated from common resource data using **resource profile information for each mobile device model.**” *Id.*, ¶29. *Hariki*’s Figure 3, below, shows an example resource profile used to generate the UI content package. *Id.*, ¶35.

172. A POSITA would have recognized that the resource profiles used to generate the UI content package include specific rendering capabilities and attributes for a specific mobile device, such as size and position. *Id.*, ¶35. As shown below, these attributes determine how the resource, for instance “JPEG” as shown for “item id= ‘ID_3’,” are to be rendered on the specific mobile device. *Id.*, ¶32. The user interface authoring tool, as described above in Sections X.A.1.b and X.A.1.c, uses the resource profiles to generate a UI content package with specific resources and a descriptor file that instructs the mobile device on how to render the converted resources of the UI content package for the backgrounds, title bars, buttons, and alert sounds (i.e., “rendering blocks”) of the mobile device. *Hariki*, ¶¶33, 45-46.

173. Accordingly, the *Hariki-Harris* system discloses “rendering blocks operates specific to a wireless device type of said wireless device,” as recited in the claim.



Hariki, FIG. 3

174. *Hariki*'s Figure 6, below, further shows the UI content package sent to the mobile device. *Hariki*, ¶48. The package includes a "description file contains a number of item entries 601. Each item entry consists of certain items of information, such as identifier (ID), path (location), and file type, among other parameters, relating to a particular resource." *Id.* A POSITA would have recognized that the type of resource, e.g., Flash, PNG, JPEG, is not specific to any particular mobile device (e.g., device independent), although the resource itself themselves are included based on the type of mobile device. Although, inclusion of the resources themselves

is based on the type of mobile device. It was well-known to a POSITA before August 2007 that a Flash, PNG ,or JPEG file contained commands that could be rendered on many different device types. Syntax is merely the manner in which the code is written, and a POSITA would understand to use syntax that was usable by a wide range of devices, thus the format of commands in these files (i.e., “syntax”) was device independent. *Id.*

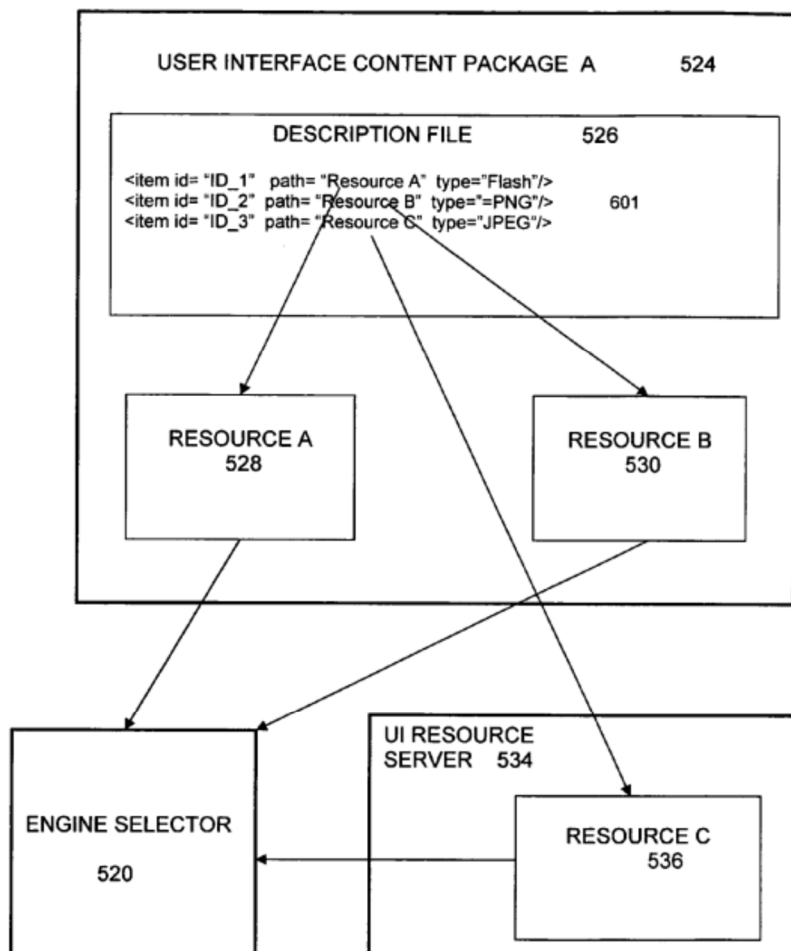


FIGURE 6

Hariki, FIG. 6

175. As explained above Section X.A.1.c, the converted resources in the UI content package, also called the UI skin, determine the configuration of the mobile device when used by the engines of the resource API. Consequently, a POSITA would have understood that the converted resources in the UI content package instruct the mobile device how to display the contents of the converted resource.

176. Furthermore, the resource API engines that convert content use the device independent command structure of the converted resources (corresponding to the claimed “syntax that is generic to said wireless device type”) to display content on backgrounds, title bars, buttons, and alert sounds (i.e., “rendering blocks”) on the mobile device.

7. Claim 7 “A method as described in claim 6 wherein said custom configuration comprises a syntax that is generic regarding said wireless device type.”

177. It is my opinion that *Hariki* in combination with *Harris* discloses claim 7. As explained above in Section X.A.5, the converted resources in the UI content package, use a device independent command structure. Accordingly, the converted resources (i.e., “custom configuration”) “comprises a syntax that is generic regarding said wireless device type.”

8. Claim 8 “A method as described in claim 1 wherein said custom configuration comprises configuration information and content specific to said application.”

178. It is my opinion that *Hariki* in combination with *Harris* discloses claim 8. As explained above in Section X.A.1.b, the “UI skin consisting of a **particular screen configuration**, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection” may be included with the mobile device. *Id.*, ¶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 also provide the customized UI to the mobile device. *Id.* As further explained above in Section X.A.1.h, the UI content package includes a converted resource such as a Flash, PNG, or JPEG 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, *Hariki* discloses that “custom configuration comprises configuration information.”

179. Furthermore as explained above in Section X.A.1.c, *Hariki* explains that “the resources referenced by the descriptor file and **utilized by the application** must be referenced in some way by the application and **then retrieved and converted to an appropriate format for use by the application.**” *Id.*, ¶39.

180. Therefore, *Hariki* discloses that the converted resource file (corresponding to “configuration information”) and resources that are formatted and utilized based on a specific application (corresponding to “content specific to said application”).

9. **Claim 9 “A method as described in claim 1 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored by said wireless device and wherein further said identifying said custom configuration comprises receiving an identifier that identifies said custom configuration.”**

181. It is my opinion that *Hariki* in combination with *Harris* discloses claim 9. *Hariki* discloses that “[t]he mobile device handset includes a resource API to **download...UI content that is locally stored.**” *Hariki*, ¶19. *Hariki* further discloses that “[t]he mobile handset 502 also includes a data storage component 522, which can be implemented as any type of fixed, removable or on-board memory device. **One or more user interface content packages 524 and 522 can be stored in this data storage.**” *Hariki*, ¶45. *Hariki* also discloses a “UI content package comprises the appropriate converted resources.” *Id.*, ¶33. In other words, *Hariki*’s mobile device stores UI content packages locally, which include converted resources.

182. Additionally, as explained in Sections X.A.1.a and X.A.1.b, *Hariki* discloses that to access stored converted resources, the server sends a description file containing a ID to the mobile device that identifies the converted resources of the UI content package to use in the mobile device. *Hariki* explains that the description file

received by the mobile device identifies resources with “certain items of information, such as identifier (ID), path (location), and file type, among other parameters, relating to a particular resource.” *Id.*, ¶48. The mobile device then accesses the resource by specifying the resource ID. *Id.*, ¶50.

183. A POSITA would have understood that the ID in the server-transmitted description file identifies converted resources in the mobile device’s locally stored UI content package, which are then used by the mobile device to render content.

184. Accordingly, *Hariki* discloses converted resources (corresponding to “custom configuration”) are stored locally in the mobile device (corresponding to “one of a plurality of memory-stored custom configurations stored by said wireless device”) and a server-transmitted ID (corresponding to “an identifier that identifies said custom configuration”) is used to identify a specific converted resource.

10. Claim 10 “A method as described in claim 1 further comprising receiving and storing said custom configuration.”

185. The elements of this claim are substantively identical to the elements in claims [1a], [1b] and 9. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Sections X.A.1.ba, X.A.1.b, X.A.1.c, X.A.9.

186. For example, as explained in Section X.A.1.a, *Hariki* discloses that a mobile device receives a “downloadable UI content package consisting of [a] description file and [] converted resource files.” *Hariki*, ¶19. POSITAs would have

understood that the converted resource files and the description file in *Hariki*'s UI content package correspond to the claimed "custom configuration" and "identification of [the] custom configuration," respectively that are received by the mobile device. Additionally, as explained in Section X.A.9, *Hariki* discloses that the mobile device includes a memory for storing received information including converted resource files. Accordingly, *Hariki* in combination with *Harris* discloses this feature for the same reasons discussed in Sections X.A.1.ba, X.A.1.b, X.A.1.c, X.A.9.

11. **Claim 11 "A method as described in claim 1 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across a display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device."**

187. It is my opinion that *Hariki* discloses claim 11. *Hariki* explains that the UI skin may consist "of a particular screen configuration, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection." *Id.*, ¶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 can configure "backgrounds, title bars, buttons, alert sounds, **and so on.**" *Id.*, ¶24.

188. As explained in Section X.A.1.b above, a POSITA would have understood that these backgrounds, title bars, buttons, and alert sounds comprise different types of “rendering blocks” within the context of the ’245 patent.

189. Furthermore, a POSITA would have understood that *Hariki*’s multiple rendering blocks including keypad buttons (corresponding to “button images” on the display screen); ringtones and alert sounds (corresponding to “rendering of audio” on the mobile device); and title bars (corresponding to “ticker information” on the display screen) cover many different customizable aspects of a mobile device, and fall squarely within the ’245 patent’s examples of displaying different types of rendering blocks on the mobile device.”

- 12. Claim 12 “A non-transitory computer usable medium comprising instructions therein that when executed by a processor implement a method of rendering content on a wireless device, said method comprising: receiving an identification of a custom configuration of a plurality of rendering blocks of said wireless device, wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application; receiving compiled content generated in part from execution of said application wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device; using a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration; and rendering said renderable content on said wireless device, wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content.”**

- a. [12pre] A non-transitory computer usable medium comprising instructions therein that when executed by a processor implement a method of rendering content on a wireless device, said method comprising:

190. It is my opinion that *Hariki* in combination with *Harris* discloses claim 12. To the extent the preamble is limiting, *Hariki* discloses this feature.

191. This preamble of claim 12 is substantively identical to the preamble in claim 1. *Compare* [12pre] (“[a] non-transitory computer readable medium comprising instructions therein that when executed by a processor implement a **method of rendering content on a wireless device**”) *with* [1pre] (“[a] **method of rendering content on a wireless device**”). Furthermore, *Hariki* discloses that “the various functions disclosed herein may be described using any number of combinations of hardware, firmware, and/or as data and/or **instructions embodied in various machine-readable or computer-readable media**, in terms of their behavioral, register transfer, logic component, and/or other characteristics.” *Hariki*, ¶54. *Harris* likewise discloses “medium holding computer-executable steps for a method.” *Harris*, claim 21. Thus, a POSITA would have understood that any methods disclosed in *Hariki* and *Harris* would have been implemented on a “non-transitory computer readable medium.”

192. Accordingly, *Hariki* discloses this feature for substantially the same reasons discussed in Section X.A.1.a.

- b. [12a] receiving an identification of a custom configuration of a plurality of rendering blocks of said wireless device, [12b] wherein said custom configuration is associated with an application and configures said plurality of rendering blocks to render content in a manner customized to said application;

193. Each of these elements is substantively identical to the elements in claims [1a] and [1b], respectively.

194. Therefore, *Hariki* discloses these features for the same reasons discussed in Sections X.A.1.b and X.A.1.c.

- c. [12c] receiving compiled content generated in part from execution of said application; [12d] wherein said compiled content comprises render commands expressed in a syntax that is generic to said wireless device; [12e] using a graphical user interface comprising said plurality of rendering blocks to generate renderable content based on said compiled content and said custom configuration; and [12f] rendering said renderable content on said wireless device, wherein said receiving compiled content comprises: receiving first compiled content specific to a first page of said application; and receiving second compiled content specific to a second page of said application,

195. Each of these elements is substantively identical to the elements in claims [1c], [1d], [1e], [1f], respectively.

196. Therefore, *Hariki* discloses these features for the same reasons discussed in Sections X.A.1.d, X.A.1.e, X.A.1.f, X.A.1.g.

- d. [12g] wherein said custom configuration is applicable to both said first and second compiled content.

197. This element is substantively identical to the elements in claim [1g].

198. Therefore, *Hariki* discloses this feature for the same reasons discussed in Section X.A.1.h.

- 13. Claim 13 “A non-transitory computer usable medium as described in claim 12 wherein said using a graphical user interface comprising said plurality of rendering blocks to generate renderable content comprises: processing said compiled content using a reader of said wireless device; and issuing commands from said reader to individual rendering blocks of said graphical user interface based on said rendering commands of said compiled content.”**

199. The elements of this claim are substantively identical to the elements in claim 2. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.2.

- 14. Claim 14 “A non-transitory computer usable medium as described in claim 12 wherein said renderable content comprises audio content and display content.”**

200. The elements of this claim are substantively identical to the elements in claim 3. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.3.

- 15. Claim 15 “A non-transitory computer usable medium as described in claim 12 wherein said compiled content is partially result ant from said application operating on a remote server.”**

201. The elements of this claim are substantively identical to the elements in claim 4. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.4.

- 16. Claim 16 “A non-transitory computer usable medium as described in claim 12 wherein said compiled content is specific to the rendering capabilities of said wireless device.”**

202. The elements of this claim are substantively identical to the elements in claim 5. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.5.

17. Claim 17 “A non-transitory computer usable medium as described in claim 12 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.”

203. The elements of this claim are substantively identical to the elements in claim 6. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.6.

18. Claim 18 “A non-transitory computer usable medium as described in claim 17 wherein said custom configuration comprises a Syntax that is generic regarding said wireless device type.”

204. The elements of this claim are substantively identical to the elements in claim 7. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.7.

19. Claim 19 “A non-transitory computer usable medium as described in claim 12 wherein said custom configuration comprises configuration information and content specific to said application.”

205. The elements of this claim are substantively identical to the elements in claim 8. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.8.

20. Claim 20 “A non-transitory computer usable medium as described in claim 12 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored by said wireless

device and wherein further said identifying said custom configuration comprises receiving an identifier that identifies said custom configuration.”

206. The elements of this claim are substantively identical to the elements in claim 9. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.9.

21. Claim 21 “A non-transitory computer usable medium as described in claim 12 wherein said method further comprises receiving and storing said custom configuration.”

207. The elements of this claim are substantively identical to the elements in claim 10. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.10.

22. Claim 22 “A non-transitory computer usable medium as described in claim 12 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across a display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.”

208. The elements of this claim are substantively identical to the elements in claim 9. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.11.

23. Claim 23 “A wireless device operable to communicate with a remote server, said wireless device comprising: a transceiver coupled to a bus and operable to receive a custom configuration that is associated with an application, said transceiver also operable to receive compiled content generated in part from execution of said application and comprising a plurality of rendering commands

expressed in a syntax that is generic to said wireless device; a memory coupled to said bus and operable to store said compiled content and said custom configuration; a processor coupled to said bus and operable to implement the following: a graphical user interface comprising a plurality of rendering blocks and operable to generate renderable content based on said compiled content and said custom configuration wherein said custom configuration is operable to configure said plurality of rendering blocks to render content in a manner customized to said application; an engine for reading said compiled content and responsive thereto for causing said graphical user interface to generate said renderable content based on said render commands; a display device coupled to said bus and operable to render a portion of said renderable content, wherein said compiled content comprises: a first compiled content specific to a first page of said application; and a second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content.

a. [23pre] A wireless device operable to communicate with a remote server, said wireless device comprising:

209. It is my opinion that *Hariki* in combination with *Harris* discloses claim 23. To the extent the preamble is limiting, *Hariki* in combination with *Harris* discloses this feature.

210. This element is substantively identical to the element in claim [1pre]. As explained above for claim [1pre] in Section X.A.1.a., *Hariki* discloses a server that includes a user interface authoring tool and web server process, which generates renderable content for a wireless device.

211. Accordingly, *Hariki* discloses this feature for substantially the same reasons discussed in Section X.A.1.a.

- b. [23a] a transceiver coupled to a bus and operable to receive a custom configuration that is associated with an application,

212. *Hariki* in combination with *Harris* discloses this feature for substantially the same reasons discussed in Sections X.A.1.a and X.A.1.b. *Hariki*'s mobile device receives customized UI skin package from a server that alters its “look and feel”. *Hariki*, ¶24. A POSITA would have understood that *Hariki*'s UI package comprises a “custom configuration”.

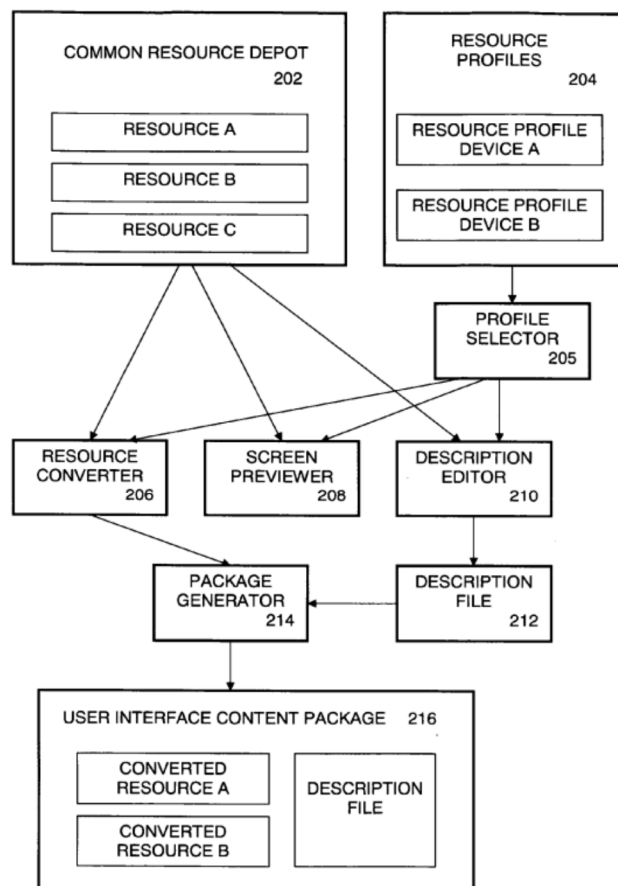


FIGURE 2

Hariki, FIG. 2

213. In particular, as discussed in Section X.A.1.a, *Hariki* explains with respect to Fig. 2 above that “UI skin package data specific to each mobile device is generated from common resource data using resource profile information for each mobile device model.” *Id.*, ¶29. *Hariki* explains that “description editor component 210 produces description files 212 based on the selected resource profile 204 and resource file 202” (*id.*, ¶33) and “resource converter 206 converts each resource into a format corresponding to the resource profile” (*id.*, ¶32). Additionally, “[t]he UI content package comprises the appropriate converted resources and the description file.” *Id.*, ¶33.

214. Additionally, as discussed in Section X.A.1.b, *Hariki* explains that “the resources referenced by the descriptor file and **utilized by the application** must be referenced in some way by the application and then retrieved and converted to an appropriate format **for use by the application.**” *Id.*, ¶39. The application may be “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.*, ¶21. Therefore, a POSITA would have understood that the converted resources in the UI package (i.e., “custom configuration”) are associated with the web browser (i.e., “an application”).

215. Accordingly, *Hariki* discloses a mobile device receiving (corresponding to the claimed “receiving” on the “wireless device”) a UI package

including converted resources (corresponding to the claimed “custom configuration”) that is utilized by an application such as a web browser (corresponding to “associated with an application”) from a server.

216. The only difference between [23a] and [1a] and [1b] is that [23a] recites “a transceiver coupled to a bus” in the mobile device. The *Hariki-Harris* system further discloses these features as well.

217. The '245 patent discloses, without providing any details about the transceiver itself, that “[t]he wireless device 300 further includes a transceiver 330 for facilitating wireless communication with a remote server. The transceiver 330 may receive a series of basic commands from a remote server that may be used to render application and/or content on the display 350.” '245*Pat*, 15:38-42. The '245 patent's Fig. 3 also does not provide any details about the transceiver and shows it as a generic box 330.

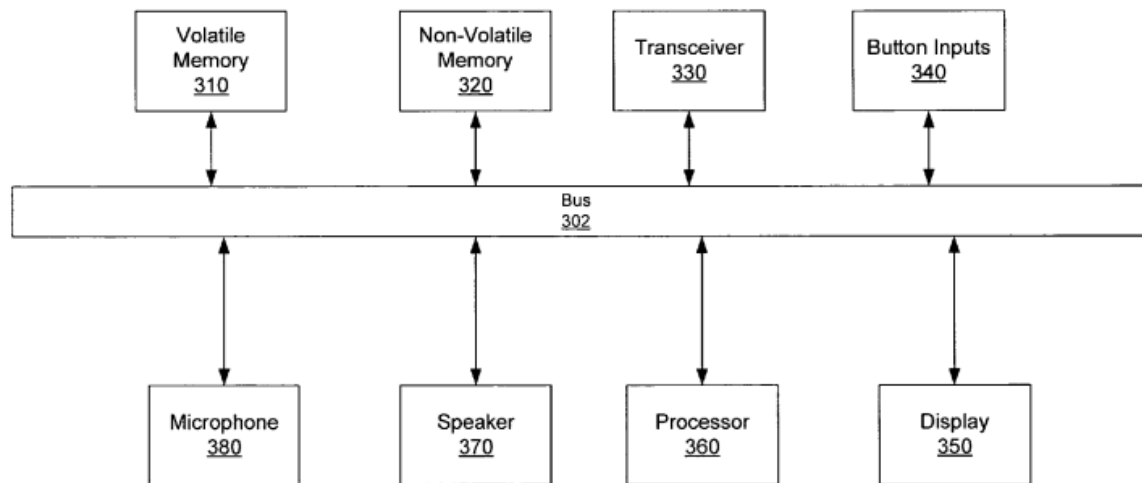


FIGURE 3

'245 patent FIG. 3

218. A POSITA would understand that *Hariki's* mobile devices would include a similar generic transceiver for communicating with the server and a bus connected to the transceiver. Furthermore, it was well known at the time of invention that mobile phones have transceivers and buses. For example, EX1022 (“*Goodacre*”) published over three years prior to the filing of the '245 patent, shows in Fig. 1A a telephonic device 100, which is similar to *Hariki's* mobile device, including an antenna 109 (corresponding to “transceiver”) and a bus 110.

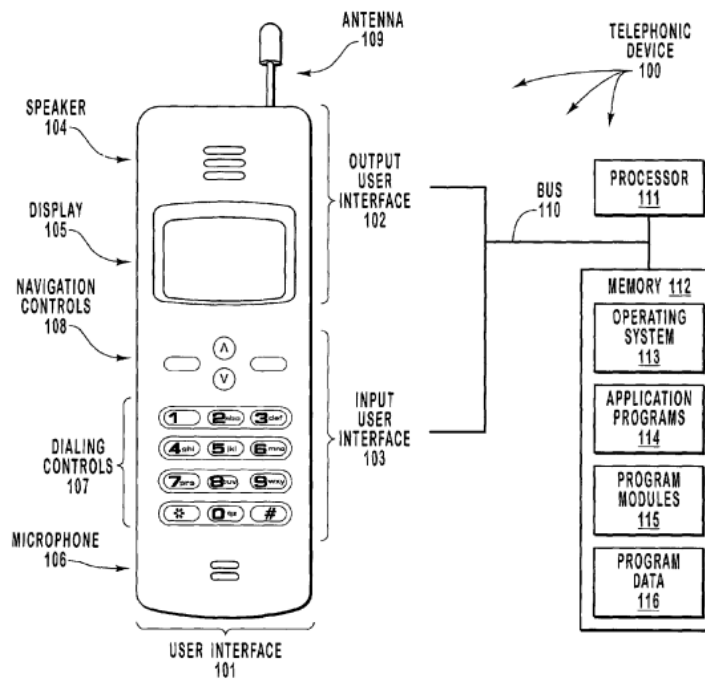
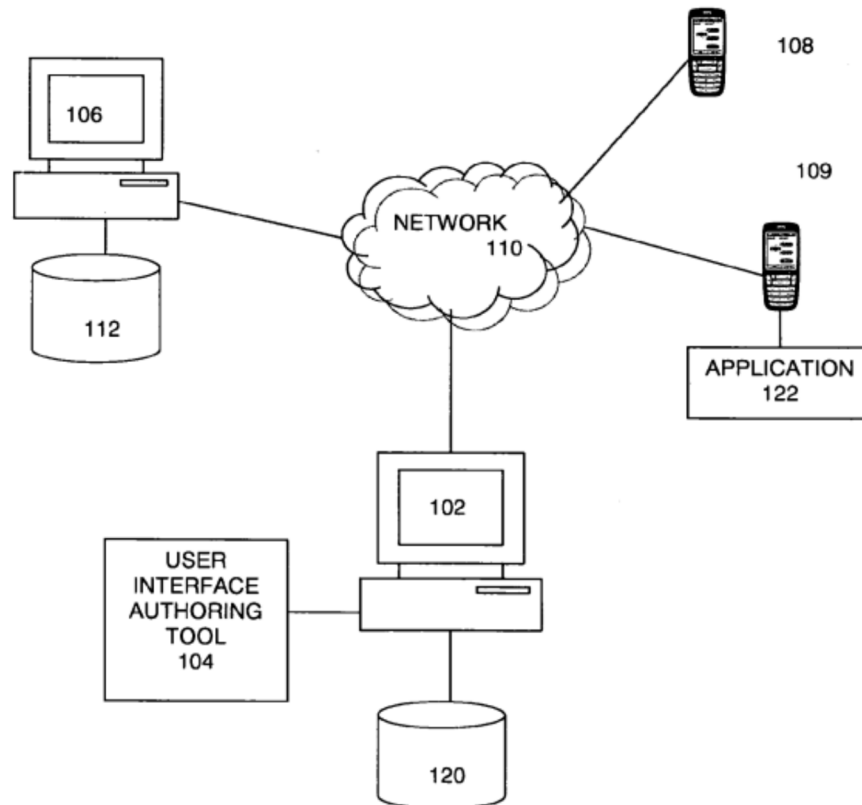


FIG. 1A

'245 patent FIG. 1A

219. The inventors in *Hariki* were well aware that the mobile devices 108 and 109 described in system 100, and illustrated in figure 1, could take many forms depending on the particular embodiment, one of those explicitly being that of mobile phones. *Hariki* ¶20. *Hariki* discloses that “present known devices, **such as the mobile phone...** are packed with greater capabilities to integrate functions beyond their **simple core functions.**” *Id.*, ¶4. A POSITA would understand that the wireless devices in the *Hariki-Harris* system included mobile phones and thus transceiver and a bus.

**FIGURE 1***Hariki*, FIG. 1

- c. [23b] said transceiver also operable to receive compiled content generated in part from execution of said application and

220. The elements of this claim are substantively identical to the elements in claim [1c]. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.1.d.

- d. 23[c] comprising a plurality of rendering commands expressed in a syntax that is generic to said wireless device;

221. The elements of this claim are substantively identical to the elements in claim [1d]. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.1.e.

e. [23d] a memory coupled to said bus and operable to store said compiled content and said custom configuration;

222. The elements of this claim are substantively identical to the elements in claim 9. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.9.

223. The only difference between [23d] and 9 is that [23d] specifically claims a “memory” in the mobile device. The *Hariki-Harris* system also discloses this feature.

224. The '245 patent discloses that the “[w]ireless device 300 also includes a volatile memory 310, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 302 for storing information and instructions to be executed by processor 360. Volatile memory 310 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 360.” ’245Pat, 15:11-17.

225. *Hariki* discloses a similar memory. *Hariki* explains that “[t]he mobile handset 502 also includes a data storage component 522, which can be implemented as any type of fixed, removable or on-board memory device.” *Hariki*, ¶45.

226. A POSITA would have understood that any data and commands received and processed by the mobile device, including compiled content and custom configuration, would have to be implemented through the mobile device's temporary or permanent memory.

- f. **[23e] a processor coupled to said bus and operable to implement the following: a graphical user interface comprising a plurality of rendering blocks and operable to generate renderable content based on said compiled content and said custom configuration**

227. The elements of this claim are substantively identical to the elements in claim [1e]. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.1.f.

The only difference between [23e] and [1e] is that [23e] specifically claims a “processor” in the mobile device. Regardless, the *Hariki-Harris* system discloses this feature as well.

228. *Hariki* discloses a processor. *Hariki* also explains that “[a]spects of the UI authoring and mobile handset system described herein may be implemented as functionality programmed into any of a variety of circuitry, including programmable logic devices (“PLDs”), such as field programmable gate arrays (“FPGAs”), programmable array logic (“PAL”) devices, electrically programmable logic and memory devices and standard cell-based devices, as well as application specific integrated circuits. Some other possibilities for implementing aspects of the method

include: **microcontrollers with memory (such as EEPROM), embedded microprocessors, firmware, software, etc.”** *Hariki*, ¶53.

- g. **[23f] wherein said custom configuration is operable to configure said plurality of rendering blocks to render content in a manner customized to said application;**

229. The elements of this claim are substantively identical to the elements in claim [1b]. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.1.c.

- h. **[23g] an engine for reading said compiled content and responsive thereto for causing said graphical user interface to generate said renderable content based on said render commands;**

230. The elements of this claim are substantively identical to the elements in claim [1e]. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.1.b.

231. The only difference between [23g] and [1e] is that [23g] claims an “engine for reading” in the mobile device. The *Hariki-Harris* system discloses this feature as shown in claim 2 in Section X.A.2 above.

- i. **[23h] a display device coupled to said bus and operable to render a portion of said renderable content, wherein said compiled content comprises: a first compiled content specific to a first page of said application; and a second compiled content specific to a second page of said application, wherein said custom configuration is applicable to both said first and second compiled content.**

232. Each of these elements is substantively identical to the elements in claims [1f] and [1g]. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Sections X.A.1.g and X.A.1.h.

233. The only difference between [23h] and [1f] and [1g] is that [23f] recites “a display device coupled to said bus” in the mobile device. The *Hariki-Harris* system also discloses these features.

234. The '245 patent discloses that the “[w]ireless device 300 may be coupled via bus 302 to a display 350, such as liquid crystal display (LCD), for displaying information on the wireless device.” '245*Pat*, 15:28-30.

235. *Hariki* discloses that “certain mobile phone service and equipment providers provide user interface (UI) customization capabilities that allow users to **personalize their phones or mobile devices with** custom ringtones, **background displays (wallpaper)**, menu configurations, and the like.” *Hariki*, ¶ 5. A POSITA would understand that *Hariki's* mobile devices would include a display for displaying backgrounds and menu configurations to a user of the mobile device. Furthermore, it was well known at the time of invention that mobile phones have displays connected to buses. For example, *Goodacre*, (EX1022) published over three years prior to the filing of the '245 patent, shows in Fig. 1A a telephonic device 100, which is similar to *Hariki's* mobile device, including a display 105 (corresponding to “a display device”) connected to a processor 111 thorough a bus 110.

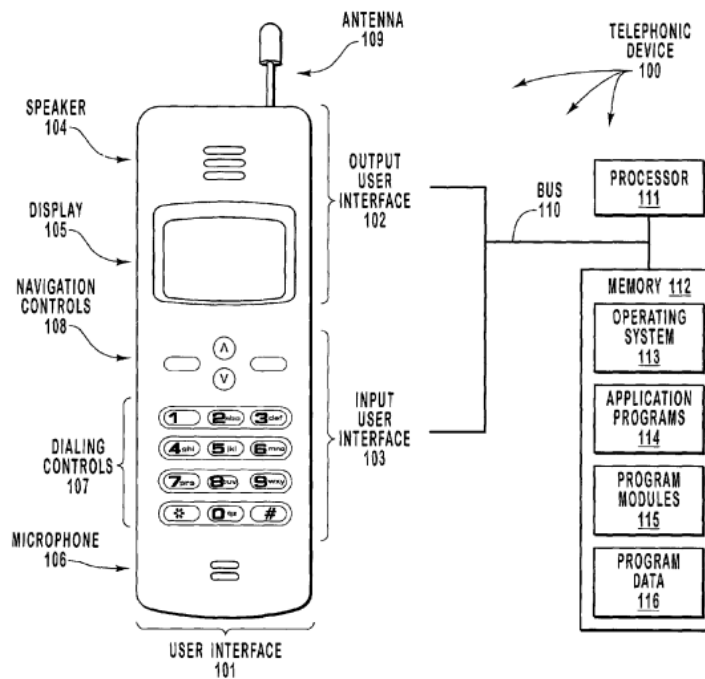


FIG. 1A

'245 patent FIG. 1A

- 24. Claim 24** “The wireless device as described in claim 23 wherein said engine is further operable to issue commands to individual rendering blocks of said graphical user interface based on said plurality of rendering commands of said compiled content.”

236. The elements of this claim are substantively identical to the elements in claim 2. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.1.2.

- 25. Claim 25** “The wireless device as described in claim 23 further comprising an audio rendering device coupled to said bus and wherein said renderable content comprises audio content and display content.”

237. The elements of this claim are substantively identical to the elements in claim 3. Accordingly, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.3.

238. The only difference between [25] and [3] is that [25] recites “an audio rendering device” in the mobile device. The *Hariki-Harris* system discloses this feature as well.

239. The ’245 patent discloses that the “hardware 102 may be the actual circuitry of the wireless device. For example, hardware 102 may be the processor, display components, user inputs, audio rendering devices, etc.” ’245*Pat*, 6:65-67.

240. *Hariki* discloses that “In system 100. User interface descriptor information comprising the customizable UI elements are down loaded 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), images, video clips, music or other sound clips, ringtones, games, small applications (applets), utilities, diagnostic tools, or any other similar data or applications, all of which are referred to as “UI content objects.” *Hariki*, ¶ 27. A POSITA would have understood that *Hariki*’s mobile devices would include an audio rendering device to play the video clips, ringtones, music, and sound clips described above. Furthermore, it was well known at the time of invention that mobile phones include speakers. As mentioned above in X.A.24, *Goodacre*, published over

three years prior to the filing of the '245 patent, shows in Fig. 1A a telephonic device 100, which is similar to *Hariki's* mobile device, including a speaker 104 (corresponding to “audio rendering device”).

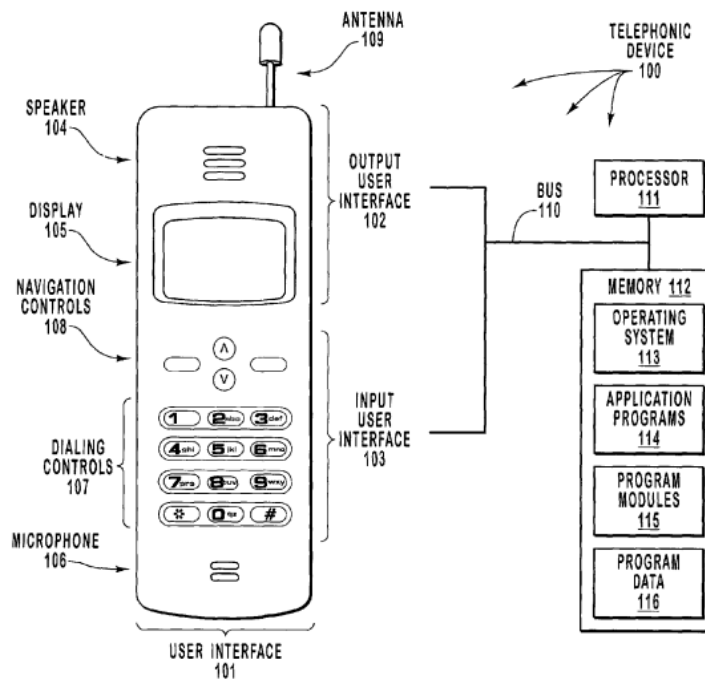


FIG. 1A

'245 patent FIG. 1A

26. Claim 26 “The wireless device as described in claim 23 wherein 25 said compiled content is partially resultant from said application operating on a remote server.”

241. The elements of this claim are substantively identical to the elements in claim 4. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.4.

- 27. Claim 27 “The wireless device as described in claim 23 wherein said compiled content is specific to the rendering capabilities of said wireless device.”**

242. The elements of this claim are substantively identical to the elements in claim 5. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.5.

- 28. Claim 28 “The wireless device as described in claim 23 wherein each of said plurality of rendering blocks operates specific to a wireless device type of said wireless device and each is instructed using a syntax that is generic to said wireless device type.”**

243. The elements of this claim are substantively identical to the elements in claim 6. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.6.

- 29. Claim 29 “The wireless device as described in claim 28 wherein said custom configuration is expressed in a syntax that is generic to said wireless device type.”**

244. The elements of this claim are substantively identical to the elements in claim 7. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.7.

- 30. Claim 30 “The wireless device as described in claim 23 wherein said custom configuration comprises configuration information and content specific to said application.”**

245. The elements of this claim are substantively identical to the elements in claim 8. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.8.

- 31. Claim 31 “The wireless device as described in claim 23 wherein said custom configuration is one of a plurality of memory-stored custom configurations stored in said memory.”**

246. The elements of this claim are substantively identical to the elements in claim 9. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.9.

- 32. Claim 32 “The wireless device as described in claim 23 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of ticker information across said display screen of said wireless device; a second block that controls the rendering of button images on said display screen; and a third block that controls the rendering of audio on a speaker of said wireless device.”**

247. The elements of this claim are substantively identical to the elements in claim 11. Therefore, *Hariki* in combination with *Harris* discloses these features for the same reasons discussed in Section X.A.11.

- 33. Claim 33 “The wireless device as described in claim 23 wherein said plurality of rendering blocks of said graphical user interface comprises: a first block that controls the rendering of a scroll bar on said display screen of said wireless device; a second block that controls the rendering of an input box on said display screen of said wireless device; and a third block that control the rendering of a text display region on said display screen of said wireless device.”**

248. It is my opinion that *Hariki* discloses claim 33. *Hariki* explains that the UI skin may consist “of a particular screen configuration, size, color scheme, font, menu scheme, keypad button assignment, ringtone selection.” *Id.*, ¶26. *Hariki* further 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 can configure “backgrounds, title bars, buttons, alert sounds, **and so on.**” *Id.*, ¶24.

249. As explained in section X.A.4., a POSITA would have understood that these backgrounds, title bars, buttons, and alert sounds comprise different types of “rendering blocks” within the context of the ’245 patent.

250. Additionally, a POSITA would have understood that *Hariki*’s multiple rendering blocks including title bars and display of menus and HTML content (corresponding to “scroll bar” on the display screen); screen configuration and size (corresponding to “input box on said display screen” on the mobile device); and screen color scheme and, font (corresponding to “text display region on said display screen” on the display screen) cover many different customizable aspects of a mobile device, and fall squarely within the ’245 patent’s examples of displaying different types of rendering blocks on the mobile device.” A POSITA would have known that menus sometimes do not fit on the screen and thus would be scrolled. Similarly, HTML content regularly does not fit on a screen and is scrolled by a device reader. Configurations of the screen including screen size would effect how all elements, including text boxes are rendered on the screen since, for example, smaller screens would require smaller rendering of text boxes. Finally, color scheme and font would be used throughout the display of the screen including text display region.

34. Motivation to Combine

251. A POSITA would have had a reasonable expectation of success of displaying content on different mobile devices using a generic syntax because it was well within a POSITA's knowledge that different client devices could receive HTML-based files from a server through the World Wide Web and render them for display. Therefore, in the *Hariki-Harris* system, 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., "expressed in a syntax that is generic to said wireless device.").

XI. SECONDARY CONSIDERATIONS

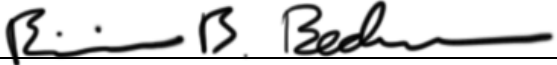
252. Based on my expertise and experience of the industry and review of the asserted prior art and other references cited in my declaration, I am not aware of any evidence of secondary considerations that would support a finding of non-obviousness. The asserted prior art demonstrates that the elements recited in the challenged claims of the '245 patent were readily available in the prior art, and there was no evidence of failure by others.

XII. CONCLUSION

253. Based on my review and above analysis of the prior art, it is my opinion that claims 1-33 of the '245 would have been obvious to a POSITA in view of *Hariki* and *Harris*.

254. 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 7, 2023.


Benjamin B. Bederson