ORIGINAL

JOSH A. KREVITT (CA SBN 208552)
jkrevitt@gibsondunn.com
H. MARK LYON (CA SBN 162061)
mlyon@gibsondunn.com
GIBSON, DUNN & CRUTCHER LLP
1881 Page Mill Road
Palo Alto, CA 94304-1211
Telephone: (650) 849-5300
Facsimile: (650) 849-5333

MICHAEL A. JACOBS (CA SBN 111664)
mjacobs@mofo.com
RICHARD S.J. HUNG (CA SBN 197425)
rhung@mofo.com
MORRISON & FOERSTER LLP
425 Market Street
San Francisco, California 94105-2482
Telephone: (415) 268-7000
Facsimile: (415) 268-7522

Attorneys for Plaintiff Apple Inc.

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

APPLE INC., a California corporation,

Plaintiff,

v.

SAMSUNG ELECTRONICS CO., LTD., a Korean corporation; SAMSUNG ELECTRONICS AMERICA, INC., a New York corporation; and SAMSUNG TELECOMMUNICATIONS AMERICA, LLC, a Delaware limited liability company,

Defendants.

CASE NO. 12-cv- CV12-00630 LHK

EXPERT DECLARATION OF DR .TODD C. MOWRY CONCERNING U.S. PATENT NO. 5,946,647

HRL

Hearing:
Date:
Time:
Place:
Judge:

PART 1 OF 7

TABLE OF CONTENTS

I. INTRODUCTION

1. I have been asked by counsel for Apple, Inc. ("Apple") to provide an opinion as to whether certain Samsung products, known as the Galaxy Nexus, infringe United States Patent No. 5,946,647 ("the'647 patent") which is assigned to Apple. I have limited my opinions in this declaration to claims 1 and 8 of the '647 patent. I have further been asked to provide my opinion as to the validity of such claims. I have also been asked to provide an opinion as to whether the Apple iOS devices incorporate features claimed in the '647 patent. My opinions are set forth below in this declaration.

2. If I am called as an expert witness, I expect to testify regarding my background, qualifications and experience relevant to the issues in this action, the technical subject matter of the '647 patent, the accused Samsung products, and the Apple products that incorporate features claimed in the '647 patent.

II. BACKGROUND AND QUALIFICATIONS

3. I am a Professor in the Department of Computer Science at the University of Carnegie Mellon University. I also have a courtesy appointment in the Department of Electrical and Computer Engineering. I have served on the faculty of the University of Carnegie Mellon University for fourteen (14) years starting in 1997 through the present (2012).

4. I also served on the faculty of the University of Toronto for four (4) years between 1993 and 1997, in the Department of Electrical and Computer Engineering and a courtesy appointment in the Department of Computer Science. Prior to that appointment, I served as a Graduate Research Assistant in the Department of Electrical Engineering at Stanford University for four (4) years between 1989 and 1993.

5. As a faculty member, I taught courses and directed research in computer systems and software, operating systems, distributed and network systems, object-oriented programming and design, and mobile computing.

6. I received a B.S. degree in Electrical Engineering with Highest Distinction from the University of Virginia in May 1988. I received an M.S. in Electrical Engineering from Stanford University in June 1989, and a Ph.D. in Electrical Engineering from Stanford University in March 1994.

7. I have worked in the computer industry in various capacities. I was a part-time Computer Architect and then Computer Architecture Consultant at Silicon Graphics, Inc. in Mountain View, California (formerly MIPS Computer Systems in Sunnyvale, California) from 1989 to 1993 and 1993 to 1996, respectively. I was a Visiting Scientist at IBM in Toronto from 1996 to 2004. During that same time period (1996 to 2004), I was also a Member of the Technical Advisory Board of SandCraft, Inc. in Santa Clara, California. I was the Director of the Intel Research Pittsburgh Lab at Intel Corporation in Pittsburgh, Pennsylvania from 2004 to 2007.

8. I have authored fifteen (15) journal articles, forty-four (44) conference papers, and twenty (20) technical reports. I am also an inventor on five (5) patents.

9. I am the recipient of several honors and awards: the Arthur Samuel Thesis Award (awarded by the Stanford Computer Science department to the top two Ph.D. theses in a given year), several IBM Faculty Development Awards (1996, 1997, 1998, 2000, 2001, 2002, and 2003), several Best Paper Awards (the Second Symposium on Operating Systems Design and Implementation in 1996; the 20th International Conference on Data Engineering (ICDE) in 2004), the Alfred P. Sloan Research Fellowship (awarded to researchers in recognition of distinguished performance and a unique potential to make substantial contributions to their field), the Most Thought-Provoking Idea

Award (awarded by the Architectural Support for Programming Languages and Operating Systems (ASPLOS), in 2004), and the TR100 Award (awarded by MIT's Technology Review magazine to the top 100 most promising young innovators in science and technology, in 1999).

10. I am a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM). I am an Associate Editor of the ACM Transactions on Computer Systems (since 2001). I was the Program Chair of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) in 2010. I was the Co-Program Chair of the International Conference on Parallel Architectures and Compilation Techniques (PACT) in 2001. I have been on the programming committee in various years for ASPLOS, the International Symposium on Computer Architecture (ISCA), the International Symposium on Microarchitectures, the Workshop on Architectural and System Support for Improving Software Dependability (ASID), the Workshop on Memory System Performance (MSP) in conjunction with Programming Language Design and Implementation (PLDI), the IBM CASCON (Centre for Advanced Studies Conference), and the First SUIF (Stanford University Intermediate Format) Workshop.

11. Overall, I have over eighteen (18) years of experience in the field of computer science and, specifically, computer architecture, compiler optimizations, operating systems, and parallel processing. A copy of my *curriculum vitae*, including references to the publications I authored, is attached hereto as Exhibit 1.

12. In 2010, I was retained by Apple in *In the Matter of Certain Personal Data and Mobile Communications Devices and Related Software,* Inv. No. 337-TA-710, an ITC investigation involving alleged infringement of Apple's patents by devices manufactured by HTC Corporation.

13. In 2010, I was also retained by Apple in *In the Matter of Certain Mobile Communications and Computer Devices and Components Thereof,* Inv. No. 337-TA-704, an ITC investigation involving alleged infringement of Apple patents by devices manufactured by Nokia Corp. and Nokia Inc.

14. In 2011, I was also retained by Apple in *Apple Inc. v. Motorola, Inc. and Motorola Mobility, Inc.*, Case No. 10-CV-662 (W.D. Wis.), involving alleged infringement of Apple patents by devices manufactured by Motorola, Inc. and Motorola Mobility, Inc.

15. 16. In 2011, I was also retained by Apple in *Nokia Corporation v. Apple Inc.*, Case Nos. 09-cv-791 and 1002 (D. Del.), 11-cv-015 (D. Del.), and ITC Docket Nos. 2702 and 2707, involving alleged infringement of Apple's patents by devices manufactured by Nokia Corp. and Nokia Inc.

16. 17. In 2011, I was also retained by Infineon in *Infineon Technologies AG and Infineon Technologies North America Corp. v. Atmel Corporation*, Civ. No. 1:11-cv-00307 (D. Del.), involving alleged infringement of Infineon's patents by devices manufactured by Atmel Corp.

17. Prior to 2010, I had not been retained as a legal expert in the preceding six years.

18. I am being compensated for the time I have spent on this litigation at my customary rate of $500 per hour. My compensation does not depend in any way upon the opinion I provide or the outcome of this litigation.

III. MATERIALS REVIEWED

19. In forming the opinions set forth in this declaration, I considered and relied upon my education, background, experience, and prior presentations and publications. I also reviewed and considered the '647 patent and its prosecution history, as well as the other documents or reference materials cited or listed in this declaration. In addition, I have evaluated representative samples of

the accused products. Exhibit 2 includes a list of materials I have reviewed. If I am called to testify or to give additional opinions regarding this matter, I reserve the right to rely upon any additional information or materials that may be provided to **me or that are relied upon by any of Samsung's** experts or witnesses.

20. I understand that discovery in this investigation has not yet commenced, and I reserve the right to supplement my opinions in this declaration should additional information become available to me.

IV. SUMMARY OF OPINIONS

21. Based on information currently available to me, it is my opinion that:

a. **The Samsung Galaxy Nexus (the "Accused Samsung Products") infringe at least claims 1 and 8 of the '647 patent;**

b. **Claims 1 and 8 of the '647 patent are valid; and**

c. **Apple's iOS devices incorporate the features claimed in the '647 patent.**

V. U.S. PATENT NO. 5,946,647

A. Description And Background Of The '647 Patent

22. Since before February 1996, computer users have received vast amounts of data in a **wide variety of formats. ('647 patent, col. 1:13-20.)** Very often this data contains useful pieces of information that the user would like to use in performing a subsequent task. (*Id.*) For example, a user may receive an email that contains phone numbers, email addresses, web addresses, and dates. (*Id.*) Prior technology required the user to visually search in files or documents for this information. (*Id.*, col. 1:20-24.) This way of handling useful information was time-consuming and difficult, especially if the user had to search a lengthy file or document. (*Id.*) Conventional systems were developed to help search a file or document for information using pattern analysis. (*Id.*, col. 1:25-

28.) Upon finding a useful piece of information, the user still had to then copy-and-paste this information into whatever field or application the user wished in order to use the information. (*Id.*, col. 1:42-50.) However, copying-and-pasting the information into a different application can be a laborious, error-prone task. Users needed a system and method for identifying structures in computer data, enabling user selection of those structures, associating actions with those structures, enabling user selection of an action, and performing the user selected action. (*Id.*, col. 1:66-2:2.)

23. The '647 patent overcame the limitations and deficiencies of the prior systems. (*Id.*, col. 1:5-20.) Generally, the '647 patent relates to a system and method for performing computer-based actions on structures identified in computer data. The patent claims a system and method of detecting structures, or instances of patterns, within various sources of computer data. Such sources include, for example, emails and word-processing documents. The patent further claims a system and method of enabling selection of a structure, linking actions to the structure, enabling selection of a linked action, and automatically performing the selected action.

24. The '647 patent allows the user to simply select the structure and select one of a number of linked actions for performance. This is particularly useful in today's mobile devices, which often prevent multiple applications from being shown simultaneously. Users no longer had to flip back and forth between screens or applications to copy a phone number from a web page to the phone dialer or to carefully highlight a number copy it into a device's memory and then paste it into a dialer application. Instead, the '647 patent combined the Internet and communications functionality of mobile devices to provide a convenient interface to users. Thus, the '647 patent provides a novel technology that increases user efficiency and enhances the computer system's functionality.

B. Asserted Claims Of The '647 Patent

25. I understand that Apple is asserting at least **claims 1 and 8 of the '647 patent. Those** claims recite:

> **Claim 1.** A computer-based system for detecting structures in data and performing actions based on detected structures, comprising:
>
> an input device for receiving data;
>
> an output device for presenting the data;
>
> a memory storing information including program routines including
>
> an analyzer server for detecting structures in the data, and for linking actions to the detected structures;
>
> a user interface enabling the selection of a detected structure and a linked action;
>
> an action processor for performing the selected action linked to the selected structure; and
>
> a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines.

> **Claim 8.** The system recited in Claim 1, wherein the user interface highlights detected structures.

VI. PERSON OF ORDINARY SKILL IN THE ART

26. **Based on my review of the '647 patent, it is** my opinion that a person of ordinary skill **in the art relevant to the '647 patent is a person with at least a Bachelor's of Science degree in** computer science or equivalent coursework and some academic or work experience in the field.

VII. CLAIM CONSTRUCTION

A. Legal Standard

27. I have been informed and understand that the language of a patent claim must be **interpreted in light of the patent's claims, s**pecification, and prosecution history, as well as other evidence extrinsic to the patent. More specifically, I understand that claim terms should be given

their plain and ordinary meaning as understood by one of ordinary skill in the art as of the effective filing date of the patent application (i.e., the day the application that led to the patent was filed), unless the claims, specification, or prosecution history indicate that a special meaning was intended.

28. As one skilled in the art as of the filing date of the '647 patent, I have been asked to review certain disputed terms and phrases in the patents and determine their meaning. I further understand that I am to offer my opinion as to how those terms and phrases would have been understood by one of ordinary skill in the art around February 1, 1996 in the case of the '647 patent.

29. I have been informed by counsel that, while claims must be interpreted in view of the specification, it is improper to import limitations from the specification into the claims. Although the specification often describes specific or preferred embodiments of the invention, claims should not be limited to those embodiments.

30. I have also been informed by counsel that the prosecution history should be considered during claim construction in conjunction with the claims and specification. For the prosecution history to limit a claim, the applicant must make a clear and unambiguous disavowal of claim scope.

B. Construction of '647 Patent Terms

31. The '647 Patent was the subject of an investigation by the ITC *In the matter of Certain Personal Data and Mobile Communications Devices and Related Software*, Investigation No. 337-TA-710. I understand that in the Initial Determination the ALJ ruled on the proper constructions for various terms in the '647 Patent. I further understand that in the Final Determination, the Commission approved those constructions. I use those constructions herein, but reserve the right to amend my opinions if this Court does not adopt those constructions or additional terms are construed.

32. The ALJ made the following determinations regarding the meaning of the terms of the '647 Patent (*See* Ex. 15 - Initial Determination at 124 – 133):

Claim Term	Construction
Detecting/Detected (claims 1, 8)	Finding and identifying/ Found and identified
Structure (claims 1, 8)	**An instance of a pattern, where a "pattern"** refers to data, such as a grammar, regular expression, string, etc., used by a pattern analysis unit to recognize information in a document such as dates, addresses, phone numbers, names, etc.
Input Device (Claims 1)	Plain and ordinary meaning (hardware only)
linking actions to the detected structures (Claim 1)	linking detected structures to computer subroutines that cause the CPU to perform a sequence of operations on the particular structures to which they are linked
analyzer server (Claim 1)	A program subroutine that receives data from a document having recognizable structures, and uses patterns to detect the structures.

VIII. LEGAL STANDARDS

A. Infringement

33. I understand that to determine whether there is infringement of a patent: (1) the claims of the patent must be construed; and (2) the properly construed claims must then be compared with the accused products.

34. Where constructions have not been proposed, I have interpreted the claims as one of ordinary skill in the art would have at the time the relevant patent was filed.

35. As the second step in the infringement analysis, I understand that the properly construed claim must be compared to the accused products. I understand that an accused product may infringe a claim either literally or equivalently. I understand from counsel that literal

infringement exists when the accused product embodies each and every limitation of a given asserted claim. To literally infringe a method claim, the product must perform every step of the claim. If a product does not literally perform a step of the claim, it can still infringe under the doctrine of equivalents if the step it performs is insubstantially different from the claimed step.

36. I understand that one test for determining equivalence is to determine whether the differences between the claimed limitation and the accused product are insubstantial. I understand that another test for determining equivalence is to examine whether the step used by the accused product performs substantially the same function in substantially the same way to achieve substantially the same result as the claimed step.

37. I further understand that infringement of a method claim can be either direct or indirect. I understand that direct infringement occurs when a party makes, uses, sells, offers for sale, or imports an article covered by the claims of a patent. I understand that an indirect infringement occurs either through inducement, where a party induces another to engage in acts that constitute direct infringement, or through contributory infringement, where a party sells an article that is made for use in an infringement of the patent's claims or, put otherwise, is not a staple article of commerce that has substantial non-infringing uses.

B. Invalidity

1. Presumption of Validity

38. I have been informed that in deciding whether to issue a patent, the U.S. Patent and **Trademark Office ("PTO") examines the patent specification, its claims, and relevant prior art** references to determine whether the patent application and its claims meet the requirements for patentability. I understand that if the PTO then issues the patent, it has determined that the claims are valid. I understand that the law recognizes that the PTO possesses special expertise to conduct such

determinations, and that, as a result, each claim of a patent issued by the PTO is presumed valid by law. This presumption of validity can only be overcome if the party seeking to invalidate a claim proves invalidity by clear and convincing evidence, which I understand to mean evidence that convinces the fact-finder that it is highly probable that the particular proposition is true. I also understand that the burden to prove invalidity by clear and convincing evidence is even higher when the allegedly invalidating prior art was already considered by the PTO or is cumulative of the prior art that the PTO considered so any attempt to invalidate a claim using that type of information will not likely succeed.

2. Patentable Subject Matter

39. I understand that a claim must be directed to a new and useful process, machine, manufacture, or composition of matter and that a claim cannot merely claim a mathematical algorithm or an abstract idea. However, I understand that a claim drawn to new and useful subject matter does not become non-patentable simply because it uses a mathematical formula, computer program, or digital computer. I further understand that in analyzing the patentability of a claim, one must consider the invention as a whole, rather than dissecting the claims into old and new elements and then ignoring the presence of the old elements in the analysis.

3. Anticipation

40. I understand that a person cannot obtain a patent if someone else already has made an identical invention. I further understand that to prove anticipation, Defendants must prove with clear and convincing evidence that the claimed invention is not new.

41. I understand that to anticipate a claim, each and every element in the claim must be present in a single item of prior art. In determining whether every one of the elements of the claimed invention is found in the prior art, I understand that one should take into account what a person of

ordinary skill in the art would have understood from his or her examination of the particular prior art. I also understand that the prior art reference alleged to be anticipatory must also enable one of ordinary skill in the art to make the claimed invention without undue experimentation.

42. I understand that anticipation must be found in a single reference, device, or process. In other words, anticipation does not allow an additional reference to supply a missing claim limitation. I further understand that the prior art reference must disclose all elements of the claim within the four corners of the document and must also disclose those elements arranged as in the claim.

43. Moreover, I understand that any differences between a prior art reference and a claimed invention invoke the question of obviousness, not anticipation. In other words, I understand it is not sufficient for a prior art reference to disclose part of a claimed invention or that it includes multiple, distinct teachings that one of ordinary skill in the art might somehow combine to achieve the claimed invention. The prior art reference must disclose the claimed invention without any need for combining various disclosures not directly related to each other.

44. I further understand that the burden of proving invalidity is especially difficult or particularly heavy when prior art reference was before the Examiner during prosecution. In other words, I understand there is an added burden of overcoming the deference due to the Patent Office.

4. Obviousness

45. I understand that a claimed invention is **invalid as "obvious" if** it would have been obvious to a person of ordinary skill in the art of the claimed invention at the time the invention was made. Unlike anticipation, which allows consideration of only one item of prior art, I have been informed that obviousness may be shown by considering more than one item of prior art.

46. I further understand that the following factors must be evaluated to determine whether Respondents have established that the claimed inventions are obvious: (1) the scope and content of the prior art; (2) the difference or differences, if any, between each claim of the asserted patent and the prior art; (3) the level of ordinary skill in the art at the time the invention of the asserted patent was made; and (4) the existence of secondary considerations that indicate that the invention was obvious or not obvious.

47. I also understand that Defendants must prove obviousness by clear and convincing evidence.

48. I have been informed that one should analyze whether there are any relevant differences between the prior art and the claimed invention from the view of a person of ordinary skill in the art at the time of the invention and that one must determine the impact, if any, of such differences on the obviousness or nonobviousness of the invention as a whole, and not merely some portion of it. Moreover, I understand that a patent claim composed of several requirements is not proved obvious merely by demonstrating that each of its requirements was independently known in the prior art. I understand that it is important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the requirements in the way the claimed new invention does. I understand that this is the case because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known. Therefore, I further understand that one may evaluate whether there was some teaching, suggestion, or motivation to arrive at the claimed invention before the time of the claimed invention. I also understand that one must be careful not to determine obviousness using hindsight and that one evaluating the prior art should put oneself in the

position of a person of ordinary skill in the field of the invention at the time the claimed invention was made.

49. I understand that secondary considerations that should be taken into account include the following: 1) commercial success of a product due to the merits of the claimed invention; 2) a long-felt, but unsolved, need for the solution provided by the claimed invention; 3) unsuccessful attempts by others to find the solution provided by the claimed invention; 4) copying of the claimed invention by others; 5) unexpected and superior results from the claimed invention; 6) acceptance by others of the claimed invention as shown by praise from others in the field of the invention or from the licensing of the claimed invention; and 7) disclosures in the prior art that criticize, discredit, or otherwise discourage the claimed invention.

C. Conception And Reduction To Practice

50. I understand that Defendants can attempt to show that a patent claim was not new because the invention described was first made or invented by someone else. However, I further understand that if the '647 patent inventors first conceived of the claimed invention, even if they reduced it to practice second, the '647 patent inventors are considered the first inventors if they worked with reasonable diligence to reduce the invention to practice from a time just before the other party's conception. Moreover, I understand that if the '647 inventors conceived and reduced to practice before the other party's conception, but the other party's conception predates the filing of a patent by the '647 inventors, the '647 patent inventors maintain their right to obtain a patent if they worked with reasonable diligence to perfect their invention by making refinements and improvements that are reflected in the final patent application.

51. In this context, I understand that conception is the formation in the mind of the inventor, of a definite and permanent idea of the complete and operative invention, as it is to be

applied in practice. I further understand that reduction to practice occurs either as of the filing of the patent application or when the invention was actually made and was shown to work for its intended purpose. I also understand that reasonable diligence means that the inventor worked continuously on reducing the invention to practice but that interruptions necessitated by the everyday problems and obligations of the inventor or others working with him or her do not prevent a finding of reasonable diligence.

D. Indefiniteness

52. I understand that the requirement that patent claims be definite means that patent claims must particularly point out and distinctly claim subject matter which the applicant regards as his invention. I further understand that a claim is sufficiently definite to satisfy this requirement if one skilled in the art would understand the bounds of the claim when read in light of the specification. I have also been informed that a claim is not indefinite unless it is not amenable to construction or is insolubly ambiguous. I further understand that a claim is not indefinite merely because it poses a difficult issue of claim construction.

IX. INFRINGEMENT

A. The Web Browser On The Samsung Galaxy Nexus Has Already Been Found to Infringe the '647 Patent

53. The same web browser that is found in the Galaxy Nexus was found to infringe claims 1 and 8 of the '647 Patent when used in HTC's devices during the ITC investigation *In the matter of Certain Personal Data and Mobile Communications Devices and Related Software*, Investigation No. 337-TA-710 (*See* Ex. 15 - Initial Determination at 1, 133). I understand both the finding of infringement and the reasoning were affirmed by the Commission in the Final Determination (*See* Ex. 16- Commission Opinion at 31).

54. Google, the developer of the Android telephone operating system, provides certain "stock" applications with each release of Android, including a web browser application. During the aforementioned investigation, the ITC found that the stock Android web browser application (referred to as "Browser" in the Android source code) provided with HTC's accused products infringed at least claims 1 and 8 of the '647 Patent (*See* Ex. 15 - Initial Determination at 133-157). I understand this was confirmed by the Commission in the Final Determination (*See* Ex. 16 – Commission Opinion at 31). While the version of Android running on the HTC accused devices is different from the version of Android running on the Galaxy Nexus, the applications found to infringe previously are identical in all relevant aspects to the version of the web browser application (Browser) found in Android installed on the Galaxy Nexus.

55. In his Initial Determination, the Honorable Carl C. Charneski concluded that the accused HTC products (a number of Android phones) infringed claims 1 and 8 of the '647 patent, and that the '647 patent was valid. This finding was recently affirmed by the Commission. The accused HTC products in the ITC 710 investigation were Android phones running either Android 2.1 or Android 2.2. Each of the phones ran the standard Android Browser application, and this Browser application (running on Android) was found to infringe the claims 1 and 8 '647 Patent.

56. My analysis in this declaration of the Browser application running on the Samsung Galaxy Nexus phone is virtually identical to my analysis in the ITC 710 investigation of the Browser application running on the accused HTC Android phones because the differences between the version of the Browser application in Android 4.0 (on the Samsung Galaxy Nexus) and the version of the Browser in Android 2.1 and 2.2 (on the accused HTC Android phones) are insignificant with respect to the '647 patent. There has been some minor refactoring of the code that I discuss below (e.g., moving the bulk of original onCreateContextMenu() method of the BrowserActivity class in Android

2.1 and 2.2 into a new onCreateContextMenu() method in a Controller class in Android 4.0, which is called directly by the onCreateContextMenu() method in the BrowserActivity class in Android 4.0). While these changes in the Browser application between Android 2.1 and 2.2 versus Android 4.0 have resulted in some of the line numbers changing in my code citations, the essential functionality related to the '647 patent is, for purposes of an infringement analysis, identical in both cases. Hence for the same reasons that the Commission concluded the Browser running on the accused HTC Android phones infringed claims 1 and 8 of the '647 patent, the Samsung Galaxy Nexus phone (running essentially the same Browser application on an updated version of Android) would also infringe claims 1 and 8 of the '647 patent.

57. It is my opinion that the Galaxy Nexus literally infringes each element of claims 1 and 8 of the '647 patent. To the extent that any element is found not to be literally infringed, it is my opinion that the Galaxy Nexus performs substantially the same function as required by each element, in substantially the same way, to achieve substantially the same result.

B. Infringement Of Claim 1 Of The '647 Patent

58. It is my opinion that the Samsung Galaxy Nexus, a physical exhibit of which is attached to this declaration as Exhibit 18, containing the Browser application infringes claim 1 of the '647 patent. A summary of my opinions can be seen in the chart at Exhibit 17. An element-by-element analysis is provided below.

1. A computer-based system for detecting structures in data and performing actions based on detected structures

59. I understand that the preamble to a claim typically does not limit the scope of the claim, except under certain circumstances. However, to the extent that the preamble of claim 1 is limiting, the Samsung Galaxy Nexus phones are computer-based systems for detecting structures in

data and performing actions based on detected structures. Today's smartphones are more powerful than consumer computers from the mid-1990s. The Samsung Galaxy Nexus contains a dual-core 1.2 GHz processor, 1 Gigabyte of random-access memory (RAM), 32 Gigabytes of memory for storage, a 4.65-inch 1280 x 720-pixel Super AMOLED HD touchscreen display, and WiFi and Bluetooth connectivity. (Ex. 3 – Samsung Galaxy Nexus Android Smartphone Specs, *available at* http://www.samsung.com/us/mobile/cell-phones/SCH-I515MSAVZW-specs)) The processor found in the Samsung Galaxy Nexus phone allows it to run a Linux-kernel operating system. The Linux kernel was first created for desktop computers, and today can be found on not only Samsung Galaxy Nexus phones but also on more than 90% of the systems on the Top 500 supercomputers list. (http://i.top500.org/stats)

2. an input device for receiving data

60. The Samsung Galaxy Nexus phone includes a touchscreen that receives data inputted by the user. In particular, it contains a 4.65-inch 1280 x 720-pixel Super AMOLED HD touchscreen display. (Ex. 3) The touchscreen not only allows the user to interact with applications running on the phone but also allows the user to enter text via an onscreen keyboard. The user simply taps a letter on the keyboard, which is interpreted as textual input. The below screenshot from a Samsung Galaxy Nexus depicts the onscreen keyboard:

Samsung Galaxy Nexus Onscreen Keyboard

61. Another input device contained in every Samsung Android phone is a radio. This hardware can be found in all cellular phones as it is what enables the phone to communicate with the cellular towers. Using this radio, the Samsung Galaxy Nexus phones can receive data from a network provider. The Android operating system **includes a Radio Interface Layer ("RIL") that "provides an abstraction layer** between Android telephony services (android.telephony) and radio **hardware." (Ex. 4 – Radio Layer Interface,** *available at* http://www.kandroid.org/online-**pdk/guide/telephony.html) The phones' radio and accompan**ying circuitry receive data that is used by, for example, the Browser application.

62. The Samsung Galaxy Nexus phones also include wireless Internet (WiFi) adapters that allow the phones to connect to wireless local area networks. (Ex. 3) The wireless Internet adapters allow the phones to receive data, such as web pages, that can be loaded by the WebKit framework.

63. **The Browser application's WebKit framework receives data from the cellular and** wireless local area networks to load and disp**lay webpages. (Ex. 5 – WebKit Package Summary**

available at http://developer.android.com/reference/android/webkit/package-summary.html)[1] WebKit, including its sub-component WebCore, is responsible for making http connections, downloading webpages, and rendering those pages for display.

3. an output device for presenting the data

64. The Samsung Galaxy Nexus phone includes an output device for presenting the data. As discussed and cited above, the Samsung Galaxy Nexus phone contains a touchscreen that displays the data to the user.

4. a memory storing information including program routines including

65. The Samsung Galaxy Nexus phone contains 1 Gigabyte of random-access memory (RAM) and 32 Gigabytes of memory for storage. (Ex. 3)

5. an analyzer server for detecting structures in the data and for linking actions to the detected structures

66. To assist the reader I have organized this part of my declaration into two sub-sections: **the first on the analyzer server's de**tection of structures in data, the second on its linking of actions to the detected structures. But to be clear, it is my opinion that the analyzer server for detecting structures in the data and for linking actions to the detected structures is a single claim element.

a. detecting structures in the data

---

[1] The Samsung Galaxy Nexus appears to run the Android 4.0 platform, including the web browsing application. The source code for the Android 4.0 operating system and the applications available with Android 4.0 and on the Samsung Galaxy Nexus can be found at http://source.android.com/source/downloading.html. All references to source code are to files available from that site. Where source code is described that provides instructions or results in certain functionality, that source code is an excerpt of the relevant source code, and is not necessarily the exclusive source code for those instructions or functionality.

67. For the Browser application, the analyzer server program routines for detecting structures in data comprise the FindPartialAddress(), FindPartialEMail(), FindPartialNumber() and isFocusableText() methods of the CacheBuilder class. (Ex. 6A - external/webkit/Source/WebKit/android/nav/CacheBuilder.cpp, L:2648-2860, L:1705-2522)

68. The CacheBuilder class detects phone numbers, email addresses, and postal addresses in computer data. The BuildFrame() function of the CacheBuilder class calls the isFocusableText() function on each text node[2] in the computer data. (Ex. 6A - CacheBuilder.cpp, L:984-1462, L:1207-1208) The isFocusableText() function calls the FindPartialAddress(), FindPartialEMail(), and FindPartialNumber() functions. (Ex. 6A -CacheBuilder.cpp, L:2707-2719) Each of these functions analyzes the text of the text node and detects U.S. postal addresses, email addresses, and phone numbers, respectively. (Ex. 6A - CacheBuilder.cpp, L:1705-2522)

b. linking actions to the detected structures

69. The Samsung Galaxy Nexus phone contains an analyzer server for linking actions to the detected structures.

70. **The Browser application's analyzer server's** program routines for linking actions to the detected structures consist of the onCreateContextMenu() method of the BrowserActivity class (Ex. 6B - BrowserActivity.java, L:203-206), the onCreateContextMenu() method of the Controller class (Ex. 6C - Controller.java, L:1256-1420), the getHitTestResult() and hitTestResult() methods of the WebView class (Ex. 6D - android/webkit/WebView.java, L2497-2556), and the setIntent() method of the MenuItemImpl class (Ex. 6E - com/android/internal/view/menu/MenuItemImpl.java, L:211-214).

---

[2] Text nodes are free text within a webpage.

71. The Browser application links actions to the detected structure through the onCreateContextMenu() method of the BrowserActivity class, the onCreateContextMenu() method of the Controller class, and the getHitTestResult() method of the WebView class. The onCreateContextMenu() method in the BrowswerActivity class is called after the user has selected a detected structure through a long-press, which in turn calls the onCreateContextMenu method in the Controller class. (Ex. 6B - BrowserActivity.java, L:203-206; Ex. 6C - Controller.java, L:1256-1420) The latter onCreateContextMenu method first calls the getHitTestResult() method of the WebView class, which returns the structure selected by the user. (Ex. 6C - Controller.java, L:1265) The onCreateContextMenu() method then adds Intent objects, based on the type of structure selected, to a context menu via a call to the setIntent() method on a MenuItem object. (Ex. 6C - Controller.java, L:1311-1346) For example, when the user selects a phone number, the onCreateContextMenu() method adds Intent objects that will, upon selection, open the phone dialer with the number, add it to the user's contacts, or copy the phone number. (Ex. 6C - Controller.java, L:1312-1325)

72. An Intent object is a "description of an operation to be performed." (Ex. 7 - Android Developers Reference – Intent, *available at* http://developer.android.com/reference/android/content/Intent.html) The two primary components of an Intent object are (1) the "action to be performed" and (2) the "data to operate on." (*Id.*) For example, in the case where the Browser application detects a phone number, the Intent object that it creates in order to launch the phone dialer with the detected phone number contains the "action" component set to `ACTION_VIEW` and the "data" component set to the detected phone number prepended with the prefix "tel:". (Ex. 6C - Controller.java, L:1314-1316) The "VIEW action does what ... is considered the most reasonable thing for a particular" piece of data. (Ex. 7 - Android Developers Reference - Intent) This action "[d]isplay[s] the phone dialer with the given number filled

in." (Ex. 7 - Android Developers Reference – Intent, *available at* http://developer.android.com/reference/android/content/Intent.html#ACTION_VIEW)

73. The onCreateContextMenu() method in the Brower application's Controller class adds the Intent object to a menu by calling the setIntent() method of the MenuItemImpl[3] class. (Ex. 6C - Controller.java, L:1314-1331) The effect of calling the setIntent() method is to "[c]hange the Intent associated with this item. By default there is no Intent associated with a menu item. If you set one, and nothing else handles the item, then the default behavior will be to call [the startActivity() method of the Context class] with the given Intent." (Ex. 6F - MenuItem.java, L:228-231) Therefore, the onCreateContextMenu() method links actions to the detected structure by extracting the detected structure, creating an Intent object that specifies an action to perform, placing the detected structure within that Intent object, and adding the Intent object to a menu (via setIntent()) that is later displayed. Once selected, the action processor executes the action specified by the selected Intent object.

6. a user interface enabling the selection of a detected structure and a linked action

74. The Samsung Galaxy Nexus phone contains program routines that are a user interface that enables the selection of a detected structure and a linked action.

75. For the Browser application, once the FindPartialAddress(), FindPartialEmail(), and/or FindPartialNumber() methods of the CacheBuilder class detect structures in the webpage, the BuildFrame() method of the CacheBuilder class enables the selection of those structures by creating a variable of type `HTMLAnchorElement` for each such structure. (Ex. 6A - CacheBuilder.cpp,

[3] MenuItemImpl implements the MenuItem interface.

L:1274-1285) Every **HTMLAnchorElement** is selectable. (Ex. 6G - external/webkit/Source/WebCore/html/HTMLAnchorElement.cpp, L:140-173)

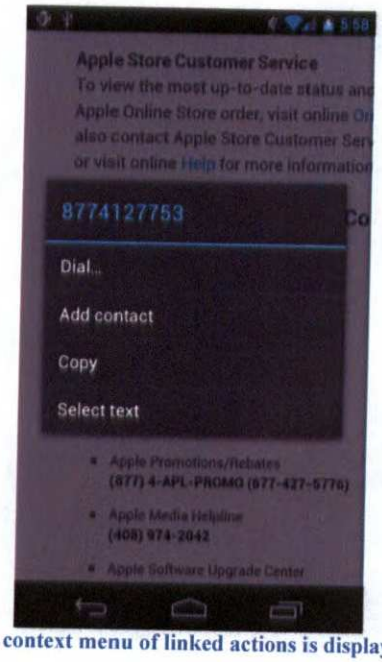
76. The Browser application further contains user interface program routines that enable the selection of a linked action by displaying a context menu. These program routines consist of the show() method of the MenuDialogHelper class and the AlertDialog class. After the user performs a long-press on the detected structure, the View **class's performLongClick**() method calls its showContextMenu() method. (Ex. 6H - View.java, L:3487-3501; Ex. 8 - Android Developers **Reference – View,** *available at* http://developer.android.com/reference/android/view/View.html) The showContextMenu() method calls the showContextMenuForChild() method of the PhoneWindow.DecorView class, which calls the show() method of the ContextMenuBuilder class. (Ex. 6H - View.java, L:3531-3533; Ex. 6I - frameworks/base/policy/src/com/android/internal/policy/impl/PhoneWindow.java, L:2115-2131) After the onCreateContextMenu() method of the BrowserActivity class links actions to the detected structure, the show() method of the ContextMenuBuilder class calls the show() method of the MenuDialogHelper class. (Ex. 6J - ContextMenuBuilder.java, L:77-94) The show() method of the MenuDialogHelper class begins the process of displaying the context menu to the user by first creating an instance of the AlertDialog class and then calling the show() method of the Dialog class. (Ex. 6K- MenuDialogHelper.java, L:51-90) The show() method of the **Dialog class then "[s]tart[s] the dialog and display[s] it on [the] screen." (Ex.** 9 - Android Developers Reference - Dialog class, *available at* http://developer.android.com/reference/android/app/Dialog.html)

77. The below figures consist of three images of the Browser application on a Samsung Galaxy Nexus running Android 4.0. Figure (a) depicts a detected phone number that the user has begun to select through a long-press **on the phone's touchscreen; (b) de**picts a context menu of linked

actions displayed to the user; and (c) depicts the phone application after the user has selected the **linked "Dial" action.**

Browser on Samsung Galaxy Nexus running Android 4.0:

(a) user begins a long-press on a detected phone number in a webpage;



(b) a context menu of linked actions is displayed;

(c) selected action is performed

7. an action processor for performing the selected action linked to the selected structure

78. The Samsung Galaxy Nexus phone includes program routines that constitute an action processor for performing the selected action linked to the selected structure. The action processor is the startActivity() method of the Context[4] class (Ex. 6L - Context.java, L:813-833), the resolveActivity() method of the Intent class (Ex. 6M - Intent.java, L:4180-4194), and the resolveActivity() method of the PackageManager class (Ex. 6N - PackageManager.java, L:1617-1645).

79. Upon selection of a linked action, the Browser application invokes the startActivity() method and passes it an Intent object that contains a description of the action to perform. (Ex. 10 - Android Application Fundamentals, *available* at http://developer.android.com/guide/topics/fundamentals.html) The startActivity() method then launches the appropriate activity, which **"is a single, focused thing that the user can do."** (Ex. 11 -

[4] These methods are implemented in the ApplicationContext.java file.

Android Developers Reference – Activity, *available at* http://developer.android.com/reference/android/app/Activity.html) **Activities are one of the "three … core components of an application" and are "activated through messages[] called** *intents*." **(Ex. 12 –** Intents and Intent Filters, *available at* http://developer.android.com/guide/topics/intents/intents-filters.html) Some Intent objects specify the Activity to launch while others specify an action to perform and the data on which to perform that action. (Ex. 7 - Android Developers Reference - Intent) In the case of the latter, the resolveActivity() method of the Intent class determines what activity should be launched (Ex. 7 - Android Developers Reference - Intent) via a call to the resolveActivity() method of the PackageManager class (Ex. 6M - Intent.java, L:4180-4194; Ex. 6N - PackageManager.java, L:1617-1645). For example, an Intent object containing the `ACTION_VIEW` action and a telephone number will cause startActivity() **to "[d]isplay the phone dialer with the given number filled in." (Ex. 7 - Android Developers Reference – Intent).**

8. and a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines

80. It is my opinion that the Samsung Galaxy Nexus phone contains a processing unit coupled to the input device, the output device, and the memory for controlling the execution of the program routines.

81. For example, the Samsung Galaxy Nexus contains a dual-core 1.2 GHz processor coupled to the input device, the output device, and the memory for controlling the execution of the **program routines. (Ex. 3 – Galaxy Nexus Tech Specs).** The processor is necessarily coupled to the output device, the touchscreen, as it must process th**e user's touches and proc**esses instructions for displaying data. The processor is also coupled to the memory of the Samsung Galaxy Nexus phone as the processor receives and acts on instructions from program routines loaded into memory. The

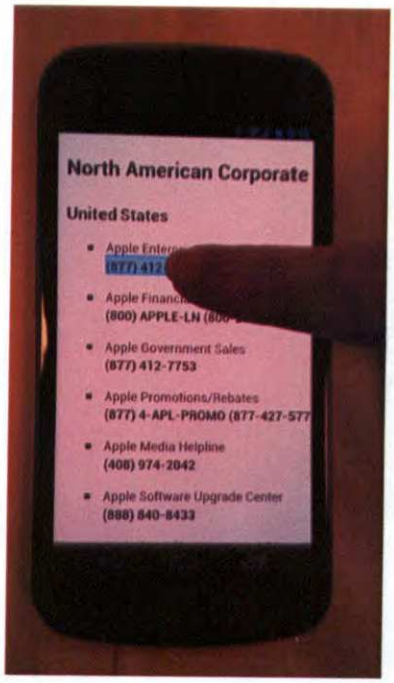
processor of the Samsung Galaxy Nexus phone also controls the execution of the program routines. Finally, the processor is coupled to the input devices such as the touchscreen, radio, and memory containing the WebKit framework.

C. Infringement Of Claim 8 Of The '647 Patent

1. The system recited in Claim 1, wherein the user interface highlights detected structures.

82. The user interface of the Browser application on the Samsung Galaxy Nexus phone highlights detected structures.

83. The Browser application highlights detected structures while the structure is being selected by the user:



Browser on Samsung Galaxy Nexus – phone number is highlighted when a user begins a long-press on a detected phone number in a webpage

X. USE BY APPLE OF THE '647 PATENTED TECHNOLOGY

84. I am aware that the ITC has already found that Apple practices at least claim 1 of the '647 patent. Specifically, during the ITC investigation *In the matter of Certain Personal Data and Mobile Communications Devices and Related Software*, Investigation No. 337-TA-710 the ITC found that the Apple iPhone 3GS running iOS version 3.1.3 met the technical prong of the domestic industry requirement (*See* Ex. 15 - Initial Determination at 157-164). I understand that this was affirmed by the Commission in the Final Determination (*See* Ex. 16 – Commission Opinion at 6).

85. For the reasons set forth in my expert report submitted in that investigation it is my opinion that the Apple iPhone 3GS running iOS 3.1.3 practices at least claim 1 of '647 patent. The relevant portions of that report are attached as Exhibit 13.

86. It is my understanding that the Apple devices running iOS 5.0 are identical with regards to the relevant functionality to the devices that Apple was found to be practicing in the above-referenced ITC action against HTC.

87. It is thus my opinion that Apple iOS devices incorporate features claimed in the '647 patent. For example, Apple iOS devices have an analyzer server for detecting structures in the data and for linking actions to the detected structures and a user interface enabling the selection of a detected structure and a linked action. Specifically, the those devices detect phone numbers and email addresses in web pages and provides actions linked to detected phone numbers or email addresses.[5]

[5] The iPad, unlike the various iPhone models, does not have telephone capability and therefore does not detect telephone number structures.

XI. VALIDITY

88. It is my opinion that the at least claims **1 and 8 of the '647 patent are valid. My** opinion that those claims are valid was confirmed by the ITC in *In the matter of Certain Personal Data and Mobile Communications Devices and Related Software*, Investigation No. 337-TA-710 where both the ITC ALJ and the Commission found that **claims 1 and 8 of the '647 patent were valid.** (*See* Ex. 15 - Initial Determination at 165 - 191); (*See* **Ex. 16 – Commission Opinion at 28-29).**

89. For the reasons set forth in my expert report submitted in that investigation it is my **opinion that the '647 patent is va**lid. A public copy of that report is attached as Exhibit I4.

90. Further, I understand that the PTO has confirmed claims 1 and 8 during reexamination **of the '647 patent.** (*See* 90/011,287 - Office Action of June 27, 2011 at 1, 27-28).

I declare under penalty of perjury that the foregoing is true and correct.

Dated: February 6 , 2012

Todd C. Mowry

Todd C. Mowry

CURRICULUM VITAE

February, 2012

Contact Information

Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsbnrgh, PA 15213

Phone: (412) 268-3725
Fax: (412) 268-5576
Email: tcm@cs.cmu.edu
URL: http://www.cs.cmu.edu/~tcm

Education

Ph.D. in Electrical Engineering, Stanford University, March 1994.

- Thesis: "Tolerating Latency Through Software-Controlled Data Prefetching."
- Supervisors: Anoop Gupta and Monica Lam.

M.S. in Electrical Engineering, Stanford University, June 1989.

B.S. in Electrical Engineering with Highest Distinction, University of Virginia, May 1988.

Academic Appointments

7/08–Present: Professor, Computer Science Department (with a courtesy appointment in the Department of Electrical and Computer Engineering), Carnegie Mellon University.

8/09–6/10: Associate Department Head for Faculty, Computer Science Department

7/97–6/08: Associate Professor, Computer Science Department (with a courtesy appointment in the Department of Electrical and Computer Engineering), Carnegie Mellon University.

12/93–6/97: Assistaut Professor, Department of Electrical and Computer Engineering (with a courtesy appointment in the Department of Computer Science), University of Torouto.

9/89–11/93: Graduate Research Assistant, Stanford University.

Honors and Awards

- Second Place, Audience Choice Award for Best Demo (for Claytronics) at the Intel Developers Forum, 2006. (Chosen among several hundred demos of Intel's latest research and technology.)
- SCS Doctoral Dissertation Award co-won by Angela Demke Brown (Ph.D. advisee), 2005.
- Most Thought-Provoking Idea Award (for Claytronics), the Wild and Crazy Idea Session IV held at ASPLOS-XI, 2004.
- Best Paper Award, the 20th International Conference on Data Engineering (ICDE), 2004.
- SCS Doctoral Dissertation Award co-won by J. Gregory Steffan (Ph.D. advisee), 2003.
- Runner-Up for Best Paper Award, the ACM SIGMOD Conference, 2001.
- Alfred P. Sloan Research Fellow, 1999-2001.
- TR100 Award (awarded by MIT's Technology Review magazine to the top 100 most promising young innovators in science and technology), 1999.
- Best Paper Award, the Second Symposium on Operating Systems Design and Implementation, 1996.
- IBM Faculty Development Awards, 1996, 1997, 1998, 2000, 2001, 2002, 2003.
- Arthur Samuel Thesis Award (awarded by the Stanford Computer Science department to the top two Ph.D. theses in a given year), 1994.

Doctoral Thesis Supervision

Student	Thesis Title (or Topic)	Graduation Date
Chi-Keung Luk	Optimizing the Cache Performance of Non-Numeric Applications. *(Nominated for the ACM Thesis Award by the University of Toronto Department of Computer Science.)*	January, 2000
J. Gregory Steffan	Hardware Support for Thread-Level Speculation. *(Co-winner of the SCS Doctoral Dissertation Award.)*	April, 2003
Antonia Zhai	Compiler Optimization of Value Communication for Thread-Level Speculation.	January, 2005
Angela Demke Brown	Explicit Compiler-based Memory Management for Out-of-core Applications. *(Co-winner of the SCS Doctoral Dissertation Award.)*	May, 2005
Christopher Colohan	Applying Thread-Level Speculation to Database Transactions.	November, 2005
Shimin Chen	Redesigning Database Systems in Light of CPU Cache Prefetching.	December, 2005
Amit Manjhi	Increasing the Scalability of Dynamic Web Applications	March, 2008
Olatunji Ruwase	Compiler Optimization for Log-Based Architectures	May, 2012 *(expected)*
Michelle Goodstein	Algorithms for Log-Based Architectures	May, 2012 *(expected)*
Vivek Seshadri	Multicore Cache Hierchies	May, 2014 *(expected)*
Gennady Pekhimenko	Monitoring to Confirm Software Patch Safety	May, 2015 *(expected)*

Masters Thesis Supervision

NOTE: these students were all at the University of Toronto.

Student	Thesis Title (or Topic)	Graduation Date
Matthew Lam	Multicast Support in Mesh Networks	July, 1996
Robert Ho	Doacross Parallelism in Single-Chip Multiprocessors	September, 1996
Angela Demke	Compiler-Inserted I/O Prefetching for Out-of-Core Applications	January, 1997
Gregory Steffan	The Potential for Thread-Level Data Speculation	January, 1997
Charles Chan	Prefetching in Networks of Workstations	February, 1998
Adley Lo	Multithreading in Networks of Workstations	February, 1998
Antonia Zhai	A Compiler Algorithm for Scheduling Non-Numeric Code with Explicitly Forwarded Data on a Speculative Multiprocessor	September, 1998
Daniel Sladic	Exploiting Fast Comunication in Single-Chip Multiprocessing	September, 1998
Sherwyn Ramkissoon	Software-Controlled Multithreading	January, 1999

Teaching Experience

Graduate Courses Taught

Course Number	Title	Year	Enrollment	Instructor Rating
15-740	Computer Architecture	2009	46	N/A
		2007	26	4.33/5.00
		2001	31	4.48/5.00
		2000	28	4.46/5.00
		1999	26	4.58/5.00
15-745	Optimizing Compilers for Modern Architectures	2011	24	N/A
		2003	23	4.74/5.00
		2003	23	4.74/5.00
		2001	14	4.56/5.00
15-740	Basic Computer Systems	1998	27	4.47/5.00
ECE1755S	Parallel Computer Architecture	1997	6	N/A
		1996	12	N/A
		1995	15	N/A
		1994	9	N/A
ECE1760S	HW/SW Tradeoffs in High-Performance Computing	1996	8	N/A
		1995	12	N/A
ECE1753F	Optimizing Compilers	1994	20	N/A

(NOTE: The University of Toronto does not collect course or instructor ratings for graduate courses.)

Undergraduate Courses Taught

Course Number	Title	Year	Enrollment	Instructor Rating
15-213	Intro to Computer Systems	2007	147	4.06/5.00
		2000	140	3.91/5.00
		1999	144	4.34/5.00
15-418/ 15-495	Parallel Computer Architecture and Programming	2011	30	N/A
		2008	14	4.20/5.00
		2004	16	3.92/5.00
		2002	23	4.80/5.00
		2002	14	4.56/5.00
15-347	Computer Architecture	1998	116	3.85/5.00
CSC372S	Microprocessor Software	1997	54	6.0/7.0
		1996	13	6.2/7.0
		1995	32	5.5/7.0
ECE540F	Optimizing Compilers	1996	34	5.68/7.0
		1995	35	5.44/7.0

Publications

Refereed Journal Articles

- Shimin Chen, Phillip B. Gibbons, Michael Kozuch, and Todd C. Mowry. Log-Based Architectures: Using Multicore to Help Software Behave Correctly. In *Operating Systems Review*, 45(1), January 2011.
- Shimin Chen, Michael Kozuch, Phillip B. Gibbons, Michael Ryan, Theodoros Strigkos, Todd C. Mowry, Olatunji Ruwase, Evangelos Vlachos, Babak Falsafi, and Vijaya Ramachandran. Flexible Hardware Acceleration for Instruction-Grain Lifeguards. In *IEEE Micro (Top Picks from 2008 Computer Architecture Conferences)*, 29(1), Jannary 2009.
- Seth C. Goldstein, Todd C. Mowry, Jason D. Campbell, Michael P. Ashley-Rollman, Michael De Rosa, Stanislav Funiak, James F. Hoburg, Mustafa E. Karagozler, Brian Kirby, Peter Lee, Padmananabhan Pillai, J. Robert Reid, Daniel D. Stancil, and Michael P. Weller. Beyond Audio and Video: Using Claytronics to Enable Pario. In *AI Magazine*, 30(2), March 2009.
- Antonia Zhai, J. Gregory Steffan, Christopher B. Colohan, and Todd C. Mowry. Compiler and Hardware Support for Reducing the Synchronization of Speculative Threads. In *ACM Transactions on Architecture and Code Optimization (TACO)*, 5(1), May 2008.
- Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Incrementally parallelizing database transactions with thread-level speculation. In ACM Transactions on Computer Systems (TOCS), 26(1), February 2008.
- Shimin Chen, Anastassia Ailamaki, Phillip B. Gibbons, and Todd C. Mowry. Improving Hash Join Performance through Prefetching. In *ACM Transactions on Database Systems*, 32(3):1-32, September 2007.
- Christopher B. Colohan, Anastasia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. CMP Support for Large and Dependent Speculative Threads. In *IEEE Transactions on Parallel and Distributed Systems*, 18(8):1041-1054, August 2007.
- J. Gregory Steffan, Christopher B. Colohan, Antonia Zhai and Todd C. Mowry. The STAMPede Approach to Thread-Level Speculation. In *ACM Transactions on Computer Systems*, 23(3):253-300, August 2005.
- Seth Copen Goldstein, Jason Campbell and Todd C. Mowry. Programmable Matter. In *IEEE Computer*, 38(6):99-101, June 2005.
- Angela Demke Brown, Todd C. Mowry and Orran Krieger. Compiler-Based I/O Prefetching for Out-of-Core Applications. In *ACM Transactions on Computer Systems*, 19(2):111-170, May 2001.
- Chi-Keung Luk and Todd C. Mowry. Architectural and Compiler Support for Effective Instruction Prefetching: A Cooperative Approach. In *ACM Transactions on Computer Systems*, 19(1):71-109, February 2001.
- Todd C. Mowry and Chi-Keung Luk. Understanding Why Correlation Profiling Improves the Predictability of Data Cache Misses in Nonnumeric Applications. In *IEEE Transactions on Computers*, 49(4), April 2000.
- Chi-Keung Luk and Todd C. Mowry. Automatic Compiler-Inserted Prefetching for Pointer-Based Applications. In *IEEE Transactions on Computers*, 48(2), February 1999.
- Mark Horowitz, Margaret Martonosi, Todd C. Mowry, and Michael D. Smith. Informing Memory Operations: Memory Performance Feedback Mechanisms and their Applications. In *ACM Transactions on Computer Systems*, 16(2):170-205, May 1998.
- Todd C. Mowry. Tolerating Latency in Multiprocessors through Compiler-Inserted Prefetching. In *ACM Transactions on Computer Systems*, 16(1):55-92, February 1998.

- Todd Mowry and Anoop Gupta. Tolerating Latency Through Software-Controlled Prefetching in Shared-Memory Multiprocessors. In *Journal of Parallel and Distributed Computing*, 12(2):87-106, 1991.

Refereed Conference Papers

- Olatunji Ruwase, Shimin Chen, Phillip Gibbons, and Todd C. Mowry. Decoupled Lifeguards: Enabling Path Optimizations for Dynamic Correctness Checking Tools. In *Proceedings of the ACM SIGPLAN 2010 Conference on Programming Language Design and Implementation (PLDI)*, June 2010.
- F. Ryan Johnson, Radu Stoica, Anastasia Ailamaki, and Todd C. Mowry. Decoupling Contention Management from Scheduling. In *Proceedings of the Fifteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2010)*, March 2010.
- Michelle Goodstein, Evangelos Vlachos, Shimin Chen, Phillip B. Gibbons, Michael Kozuch, and Todd C. Mowry. Butterfly Analysis: Adapting Dataflow Analysis to Dynamic Parallel Monitoring. In *Proceedings of the Fifteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2010)*, March 2010.
- Evangelos Vlachos, Michelle Goodstein, Michael Kozuch, Shimin Chen, Babak Falsafi, Phillip B. Gibbons, and Todd C. Mowry. ParaLog: Enabling and Accelerating Online Parallel Monitoring of Multithreaded Applications. In *Proceedings of the Fifteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2010)*, March 2010.
- Amit Manjhi, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Anthony Tomasic. Holistic Query Transformations for Dynamic Web Applications. In *Proceedings of the 2009 IEEE 25th International Conference on Data Engineering (ICDE)*, March-April 2009.
- Daniel J. Dewey, Michael P. Ashley-Rollman, Michael DeRosa, Seth Copen Goldstein, Todd C. Mowry, Siddhartha S. Srinivasa, Padmanabhan Pillai, and Jason Campbell. Generalizing metamodules to simplify planning in modular robotic systems. In *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems (IROS)*, September 2008.
- Lei Li, Wenjie Fu, Fan Guo, Todd C. Mowry, and Christos Faloutsos. Cut-and-stitch: efficient parallel learning of linear dynamical systems on SMPs. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, August 2008.
- Shimin Chen, Michael Kozuch, Theodoros Strigkos, Babak Falsafi, Phillip B. Gibbons, Todd C. Mowry, Michael Ryan, Olatunji Ruwase, and Evangelos Vlachos. Flexible Hardware Acceleration for Instruction-Grain Program Monitoring. In *Proceedings of the 35th Annual International Symposium on Computer Architecture (ISCA)*, June 2008.
- Olatunji Ruwase, Phillip B. Gibbons, Todd C. Mowry, Vijaya Ramachandran, Shimin Chen, Michael Kozuch, and Michael Ryan. Parallelizing dynamic information flow tracking. In *Proceedings of the 20th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, June 2008.
- Brian Kirby, Burak Aksak, James Hoburg, Todd C. Mowry, and Padmanabhan Pillai. A Modular Robotic System Using Magnetic Force Effectors. In *Proceedings of the IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems (IROS)*, October 2007.
- Michael Ashley-Rollman, Seth Goldstein, Peter Lee, Todd C. Mowry, and Padmanabhan Pillai. Meld: A Declarative Approach to Programming Ensembles. In *Proceedings of the IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems (IROS)*, October 2007.
- Amit Manjhi, Phillip B. Gibbons, Anastassia Ailamaki, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, Anthony Tomasic, and Haifeng Yu. Invalidation Clues for Database Scalability Services. In *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering (ICDE)*, pages 316-325, April 2007.
- Michael De Rosa, Peter Lee, Seth Goldstein, Jason Campbell, Padmanabhan Pillai, and Todd C. Mowry. Distributed Watchpoints: Debugging Large Multi-Robot Systems. In *Proceedings*

of the 2007 IEEE International Conference on Robotics and Automation (ICRA), pages 3723-3729, April 2007.

- Beujamin D. Rister, Jason Campbell, Padmanabhan Pillai, and Todd C. Mowry. Integrated Debugging of Large Modular Robot Ensembles. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2227-2234, April 2007.
- Shimin Chen, Phillip Gibbons, Michael Kozuch, Vasileios Liaskovitis, Anastassia Ailamaki, Guy Blelloch, Babak Falsafi, Limor Fix, Nikos Hardavellas, Todd C. Mowry and Chris Wilkerson. Scheduling Threads for Constructive Cache Sharing on CMPs. In *Proceedings of 18th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 105-115, June 2007.
- Shimin Chen, Babak Falsafi, Phillip B. Gibbons, Michael Kozuch, Todd C. Mowry, Radu Teodorescu, Anastassia Ailamaki, Limor Fix, Gregory R. Ganger, Bin Lin, and Steven W. Schlosser. Log-Based Architectures for General-Purpose Monitoring of Deployed Code. In *Proceedings of the Workshop on Architectural and System Support for Improving Software Dependability (ASID), held with ASPLOS XII*, October 2006.
- Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Tolerating Depeudences Between Large Speculative Threads Via Sub-Threads. In *Proceedings of the 33rd Annual International Symposium on Computer Architecture (ISCA)*, pages 216-226, June 2006.
- Amit Manjhi, Anastassia Ailamaki, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, and Anthony Tomasic. Simultaneous Scalability and Security for Data-Intensive Web Applications. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, pages 241-252, June 2006.
- Vasileios Liaskovitis, Shimin Chen, Phillip B. Gibbons, Anastassia Ailamaki, Guy E. Blelloch, Babak Falsafi, Limor Fix, Nikos Hardavellas, Michael Kozuch, Todd C. Mowry, and Chris Wilkerson. Parallel Depth First vs. Work Stealing Schedulers on CMP Architectures. In *Proceedings of 18th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, August 2006.
- Brian Kirby, Jason Campbell, Burak Aksak, Padmanabhan Pillai, James F. Hoburg, Todd C. Mowry, and Seth Copen Goldstein. Catoms: Moving Robots without Moving Parts. In *Proceedings of the Twentieth National Conference qon Artificial Intelligence (AAAI)*, pages 1730-1731, July 2005.
- Christopher Olston, Amit Manjhi, Charles Garrod, Anastassia Ailamaki, Bruce M. Maggs, and Todd C. Mowry. A Scalability Service for Dynamic Web Applications. In *Proceedings of the Second Biennial Conference on Innovative Data Systems Research (CIDR)*, pages 56-69, January 2005.
- Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Optimistic Intra-Transaction Parallelism on Chip Multiprocessors. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 73-84, September 2005.
- Shimin Chen, Anastassia Ailamaki, Phillip B. Gibbons, and Todd C. Mowry. Inspector Joins. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 817-828, September 2005.
- Antonia Zhai, Christopher B. Colohan, J. Gregory Steffan, and Todd C. Mowry. Compiler Optimization of Memory-Resident Value Communication Between Speculative Threads. In *Proceedings of the 2007 International Symposium on Code Generation and Optimization (CGO)*, pages 39-52, March 2004.
- Shimin Chen, Anastassia Ailamaki, Phillip B. Gibbons and Todd C. Mowry. Improving Hash Join Performance throngh Prefetching. In *Proceedings of the 20th International Conference on Data Engineering (ICDE)*, pages 116-127, April 2004. *(This paper received the Best Paper Award.)*

- Antonia Zhai, Christopher B. Colohan, J. Gregory Steffan and Todd C. Mowry. Compiler Optimization of Scalar Value Communication Between Speculative Threads. In *Proceedings of the Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X)*, October 2002.
- Shimin Chen, Philip B. Gibbons, Todd C. Mowry and Gary Valentin. Fractal Prefetching B+-Trees: Optimizing Both Cache and Disk Performance. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, June 2002.
- J. Gregory Steffan, Christopher B. Colohan, Antonia Zhai and Todd C. Mowry. Improving Value Communication for Thread-Level Speculation. In *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture (HPCA)*, February 2002.
- Shimin Chen, Philip B. Gibbons and Todd C. Mowry. Improving Index Performance through Prefetching. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 235-246, May 2001. *(This paper received the Runner-Up for Best Paper Award.)*
- Angela Demke Brown and Todd C. Mowry. Taming the Memory Hogs: Using Compiler-Inserted Releases to Manage Physical Memory Intelligently. In *Proceedings of the Fourth Symposium on Operating Systems Design and Implementation (OSDI 2000)*, pages 31-44, October 2000.
- J. Gregory Steffan, Christopher B. Colohan, Antonia Zhai and Todd C. Mowry. A Scalable Approach to Thread- Level Speculation. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, pages 1-12, June 2000.
- Todd C. Mowry and Sherwyn R. Ramkissoon. Software-Controlled Multithreading Using Informing Memory Operations. In *Proceedings of the Sixth International Symposium on High-Performance Computer Architecture*, January, 2000.
- Chi-Keung Luk and Todd C. Mowry. Memory Forwarding: Enabling Aggressive Layout Optimizations by Guaranteeing the Safety of Data Relocation. In *Proceedings of the 26th Annual International Symposium on Computer Architecture (ISCA)*, pages 88-99, May 1999.
- Chi-Keung Luk and Todd C. Mowry. Cooperative Prefetching: Compiler and Hardware Support for Effective Instruction Prefetching in Modern Microprocessors. In *Proceedings of the 31st Annual International Symposium on Microarchitecture*, pages 182-193, December 1998.
- J. Gregory Steffan and Todd C. Mowry. The Potential for Using Thread-Level Data Speculation to Facilitate Automatic Parallelization. In *Proceedings of the Fourth International Symposium on High-Performance Computer Architecture*, pages 2-13, February, 1998.
- Charles Chan, Adley Lo, and Todd C. Mowry. Comparative Evaluation of Latency Tolerance Techniques for Software Distributed Shared Memory. In *Proceedings of the Fourth International Symposium on High-Performance Computer Architecture*, pages 300-311, February, 1998.
- Todd C. Mowry and Chi-Keung Luk. Predicting Data Cache Misses in Non-Numeric Applications Through Correlation Profiling. In *Proceedings of the 30th Annual International Symposium on Microarchitecture*, pages 314-320, December 1997.
- Todd C. Mowry, Angela K. Demke and Orran Krieger. Automatic Compiler-Inserted I/O Prefetching for Out-of-Core Applications. In *Proceedings of the Second Symposium on Operating Systems Design and Implementation (OSDI '96)*, pages 3-17, October 1996. *(This paper received the Best Paper Award.)*
- Chi-Keung Luk and Todd C. Mowry. Compiler-Based Prefetching for Recursive Data Structnres. In *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 222-233, October 1996.

- Edouard Bugnion, Jennifer M. Anderson, Todd C. Mowry, Mendel Rosenblum and Monica S. Lam. Compiler- Directed Page Coloring for Multiprocessors. In *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 244-255, October 1996.
- Mark Horowitz, Margaret Martonosi, Todd C. Mowry, and Michael D. Smith. Informing Memory Operations: Providing Memory Performance Feedback in Modern Processors. In *Proceedings of the 23rd Annual International Symposium on Computer Architecture*, pages 260-270, May 1996.
- Todd C. Mowry, Monica S. Lam and Auoop Gupta. Design and Evaluation of a Compiler Algorithm for Prefetching. In *Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 62-73, October 1992.
- Anoop Gupta, John Hennessy, Kourosh Gharachorloo, Todd Mowry, and Wolf-Dietrich Weber. Comparative Evaluation of Latency Reducing and Tolerating Techniques. In *Proceedings of the 18th Annual International Symposium on Computer Architecture*, pages 254-263, May 1991.
- Anoop Gupta, Wolf-Dietrich Weber, and Todd Mowry. Reducing Memory and Traffic Requirements for Scalable Directory-Based Cache Coherence Schemes. In *Proceedings of International Conference on Parallel Processing*, pages 312-321, August 1990.

Technical Reports

- Evangelos Vlachos, Michelle Goodstein, Michael Kozuch, Shimin Chen, Babak Falsafi, Phillip B. Gibbons, Todd C. Mowry, and Olatunji Ruwase. Parallel LBA: Conherence-based Parallel Monitoring of Multithreaded Applications. Carnegie Mellon University Technical Report CMU-CS-09-108, March 2009.
- Michelle Goodstein, Evangelos Vlachos, Shimin Chen, Phillip Gibbons, Michael Kozuch, and Todd C. Mowry. The Butterfly Model: Theoretical Foundations. Carnegie Mellon University Technical Report CMU-CS-08-170, February 2009.
- Amit Manjhi, Anastassia Ailamaki, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, and Anthony Tomasic. Simultaneous Scalability and Security for Data-Intensive Web Applications. Carnegie Mellon University Technical Report CMU-CS-06-116, March 2006.
- Charles Garrod, Amit Manjhi, Anastassia Ailamakiii, Phil Gibbons, Bruce Maggs, Todd Mowry, Christopher Olston, and Anthony Tomasic. Scalable Consistency Management for Web Database Caches. Carnegie Mellon University Technical Report CMU-CS-06-128, July 2006.
- Amit Manjhi, Phillip B. Gibbons, Anastassia Ailamaki, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, Anthony Tomasic, and Haifeng Yu. Invalidation Clues for Database Scalability Services. Carnegie Mellon University Technical Report CMU-CS-06-139, July 2006.
- Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Supporting Large Speculative Threads for Databases and Beyond. Carnegie Mellon University Technical Report CMU-CS-05-109, July 2005.
- Christopher B. Colohan, Anastassia Ailamaki, J. Gregory Steffan, and Todd C. Mowry. Optimistic Intra-Transaction Parallelism on Chip Multiprocessors. Carnegie Mellon University Technical Report CMU-CS-05-118, March 2005.
- Shimin Chen, Anastassia Ailamaki, Phillip B. Gibbons, and Todd C. Mowry. Improving Hash Join Performance through Prefetching. Carnegie Mellon University Technical Report CMU-CS-03-157, October 2003.
- Shimin Chen, Phillip B. Gibbons, Todd C. Mowry, and Gary Valentin. Fractal Prefetching B+-Trees: Optimizing Both Cache and Disk Performance. Carnegie Mellon University Technical Report CMU-CS-02-115, March 2002.

- Spiros Papadimitriou and Todd C. Mowry. Exploring Thread-Level Speculation in Software: The Effects of Memory Access Tracking Granularity. Carnegie Mellon University Technical Report CMU-CS-01-145, July 2001.
- Shimin Chen, Phillip B. Gibbons and Todd C. Mowry. Improving Index Performance through Prefetching. Carnegie Mellon University Technical Report CMU-CS-00-177, December 2000.
- J. Gregory Steffan, Christopher B. Colohan and Todd C. Mowry. Extending Cache Coherence to Support Thread-Level Data Speculation on a Single Chip and Beyond. Carnegie Mellon University Technical Report CMU-CS-98-171, December 1998.
- Todd C. Mowry and Sherwyn R. Ramkissoon. Software-Controlled Multithreading Using Informing Memory Operations. Carnegie Mellon University Technical Report CMU-CS-98-169, October 1998.
- Chi-Keung Luk and Todd C. Mowry. Compiler and Hardware Support for Automatic Instruction Prefetching: A Cooperative Approach. Carnegie Mellon University Technical Report CMU-CS-98-140, June 1998.
- J. Gregory Steffan, Christopher B. Colohan and Todd C. Mowry. Architectural Support for Thread-Level Data Speculation. Carnegie Mellon University Technical Report CMU-CS-97-188, November 1997.
- Todd C. Mowry and Chi-Keung Luk. Predicting Data Cache Misses in Non-Numeric Applications Through Correlation Profiling. Carnegie Mellon University Technical Report CMU-CS-97-175, September 1997.
- J. Gregory Steffan and Todd C. Mowry. The Potential for Thread-Level Data Speculation in Tightly-Coupled Multiprocessors. University of Toronto Technical Report CSRI-TR-350, Febrnary 1997.
- Chi-Keung Luk and Todd C. Mowry. Predicting Data Cache Misses in Non-Numeric Applications Through Correlation Profiling. University of Toronto Technical Report CSRI-TR-359, February, 1997.
- Mark Horowitz, Margaret Martonosi, Todd C. Mowry, and Michael D. Smith. Informing Loads: Enabling Software to Observe and React to Memory Behavior. Stanford University Technical Report CSL-TR-95-673, July 1995.
- Todd C. Mowry. Tolerating Latency Through Software-Controlled Data Prefetching. Technical Report CSL-TR-94-628, Stanford University, June 1994.

Patents Held

- Todd C. Mowry. U.S. Patent 7,127,586: Prefetching hints. Issued October, 2006.
- Shimin Chen, Phillip B. Gibbons, and Todd C. Mowry. U.S. Patent 6,772,179: System and method for improving index performance through prefetching. Issued August, 2004.
- Todd C. Mowry. U.S. Patent 6,240,488: Prefetching hints. Issued May, 2001.
- Todd C. Mowry. U.S. Patent 5,732,242: Consistently specifying way destinations through prefetching hints. Issued March, 1998.
- Todd C. Mowry and Earl A. Killian. U.S. Patent 5,696,958: Method and apparatus for reducing delays following the execution of a branch instruction in an instruction pipeline. Issued December, 1997.

Distinguished Lectures

2/7/08: "Parlo: the Next Step Beyond Audio and Video", University of Toronto, Department of Electrical and Computer Engineering Distinguished Lecture Series.

Professional Activities

Professional Societies

- Member of the Institute of Electrical and Electronics Engineers (IEEE)
- Member of the Association of Computing Machinery (ACM)

Industrial Employment

5/04–6/07: Intel Corporation, Pittsburgh, Pennsylvania: Director of the Intel Research Pittsburgh Lab.

6/89–11/93: Silicon Graphics, Inc., Mountain View, California (formerly MIPS Computer Systems, Sunnyvale California): Computer Architect (part-time).

Consulting

7/07–2/11: Intel Corporation, Pittsburgh, Pennsylvania: Research Advisor.

8/96–4/04: SandCraft, Inc., Santa Clara, California: Member of the Technical Advisory Board.

1/96–4/04: IBM, Toronto: Visiting Scientist.

12/93–12/96: Silicon Graphics, Inc., Mountain View, California: Computer Architecture Consultant.

University Committee Work and Other Service Activities

Committee Work

- Faculty Hiring Committees:
  - Computer Science Department: 2008, 2010.
  - Computer Systems Area: 2000, 2001, 2002, 2003, 2004.
- Chair of the Computer Science Department Head Selection Committee, 2007.
- CSD Ph.D. Admissions Committee, 2001, 2002, 2003.
- School of Computer Science Council, 2001-2004.
- Carnegie Mellon University Faculty Senate (Presidential Appointeee), 2000-2001.
- Computer Systems Area Advocate, 2000-2004.
- Doctoral Review Committee, 1998-2004.
- ACM Thesis Award Nomination Committee, 1999.

Journal, Conference and Workshop Organization

- Associate Editor, ACM Transactions on Computer Systems, 2001-present.
- Program Chair of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2011.
- Sponsorship Chair, International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2010.
- Member of the ISCA program committee, 1998, 2000, and 2011.
- Member of the ASPLOS program committee, 2000, 2004.
- Co-Program Chair (with John Shen) of the International Conference on Parallel Architectures and Compilation Techniqnes (PACT), 2001.
- Member of the International Symposuim on Microarchitecture program committee, 1998.
- Member of the Workshop on Architectural and System Support for Improving Software Dependability (ASID) program committee, 2005.
- Member of the Workshop on Memory System Performance (MSP) program committee (in conjunction with PLDI), 2002 and 2004.
- Member of the IBM CASCON program committee, 1995, 1996, 1997.
- Member of the First SUIF Workshop Program Committee, 1996.
- Organized the CASCON 97 workshop on Compiler Optimization, 1997.
- Initiated and organized the 1st Toronto / Rochester Workshop on Shared-Memory Multiprocessors, 1994.

Miscellaneous

- Organizing the Computer Systems faculty lunches, 1999-2008, 2009-present.
- Organized the Computer Systems seminar series in the CMU CS department, 1998-2001.

List of Materials Considered

- U.S. Patent No. 5,946,647
- Prosecution History for U.S. Patent No. 5,946,647 and references
- Public document from ITC Investigation- *In the matter of Certain Personal Data and Mobile Communications Devices and Related Software*, ITC Investigation No. 337-TA-710, including:
  - Initial Determination
  - Commission Opinion
  - Expert Report of Dr. Todd C. Mowry Regarding Infringement and Domestic Industry for U.S. Patent No. 5,946,647
  - Rebuttal Expert Report of Dr. Todd C. Mowry Regarding Validity of U.S. Patent No. 5,946,647
- Samsung Galaxy Nexus
- Photographs of Samsung Galaxy Nexus
- Specifications for Samsung Galaxy Nexus, available at:
  - http://www.google.com/nexus/#/tech-specs
- Samsung Website, including pages available at:
  - http://www.samsung.com/us/mobile/cell-phones/SCH-I515MSAVZW-specs
- Android Developer Docs, including:
  - http://www.kandroid.org/online-pdk/guide/telephony.html
  - http://developer.android.com/reference/android/content/Intent.html
  - http://developer.android.com/reference/android/view/View.html
  - http://developer.android.com/reference/android/app/Dialog.html
  - http://developer.android.com/guide/topics/fundamentals.html
  - http://developer.android.com/reference/android/app/Activity.html
  - http://developer.android.com/guide/topics/intents/intents-filters.html

- Relevant portions of Android 4.0 source code, available at http://www.google.com/nexus/#/tech-specs and http://source.android.com/source/downloading.html

- Apple iPhone 4S

- Photographs of Apple iPhone 4S

Galaxy Nexus Android Smartphone

SCH-I515 ★★★★★ 322 REVIEWS

Print This

Carrier

Type	Verizon

Form Factor

Form Factor	Bar, Touchscreen

Size

Product Dimensions (inches)	5.33" x 2.67" x 0.37"
Weight (ounces)	5.1 ounces

Color

Color	Metallic Silver

Battery

Battery, Standby	150 hours (6.25 days)
Battery, Talk Time (hours)	Up to 12 hours
Battery Type and Size	3.7 Volt, Lithium Ion, 1850mAh

Network

Frequencies and Data Type	CDMA/PCS/1xEVDO Rev. A: 800/1900 MHz,LTE: 700 MHz
Data Speed	LTE, EVDO Rev. A

Platform

Platform	Android™ 4.0, Ice Cream Sandwich

CPU

Processor Speed, Type	1.2GHz Dual-Core Processor

Display

Main Display Resolution	1280x720 pixels
Main Display Size	4.65" Display

Main Display Technology	HD Super AMOLED™ contoured display

User Interface

Features	Widgets; Accelerometer; Supported Languages: Spanish, French

Camera

Camera Resolution	5.0 MP
Front-facing Camera Resolution	1.3 MP
Digital Optical Zoom	Digital/Optical Zoom
Features	Auto Focus; Shot Modes, Action, Night, Sunset, Party, Single Motion Panorama; Editing Modes; Camcorder; HD Recording; HD Playback

Audio

Features	Music Player; Compatible Music Files, WAV, MP3, AAC, AAC+, eAAC+, AMR-NB, AMR-WB, MIDI; Audio, Streaming; MP3/Music Tones

Video

Features	Video Player; Compatible Video Files, H.263, H.264, MPEG4, VC-1; Video, Streaming

Fun and Entertainment

Features	Wallpapers, Animated

Messaging Options

Features	Email; Corporate Email; Picture Messaging; Text Messaging; Instant Messaging; Threaded / Chat-style Messages; Video Messaging; Predictive Text

Connectivity

Features	Bluetooth®; Bluetooth® Profiles, A2DP, AVCTP, AVRCP, GAVDP, HFP, HSP, OPP, PAN, PBAP, SPP ; Wi-Fi® ; Wi-Fi® Hotspot; HTML Browser

Memory

Internal Memory	32GB

Calling Functions

Features	Speakerphone; Voicemail; Speech-to-Text; Text-to-Speech; Music ID; Picture Caller ID; Multitasking; Hearing Aid Compatible (HAC); TTY; Airline Mode

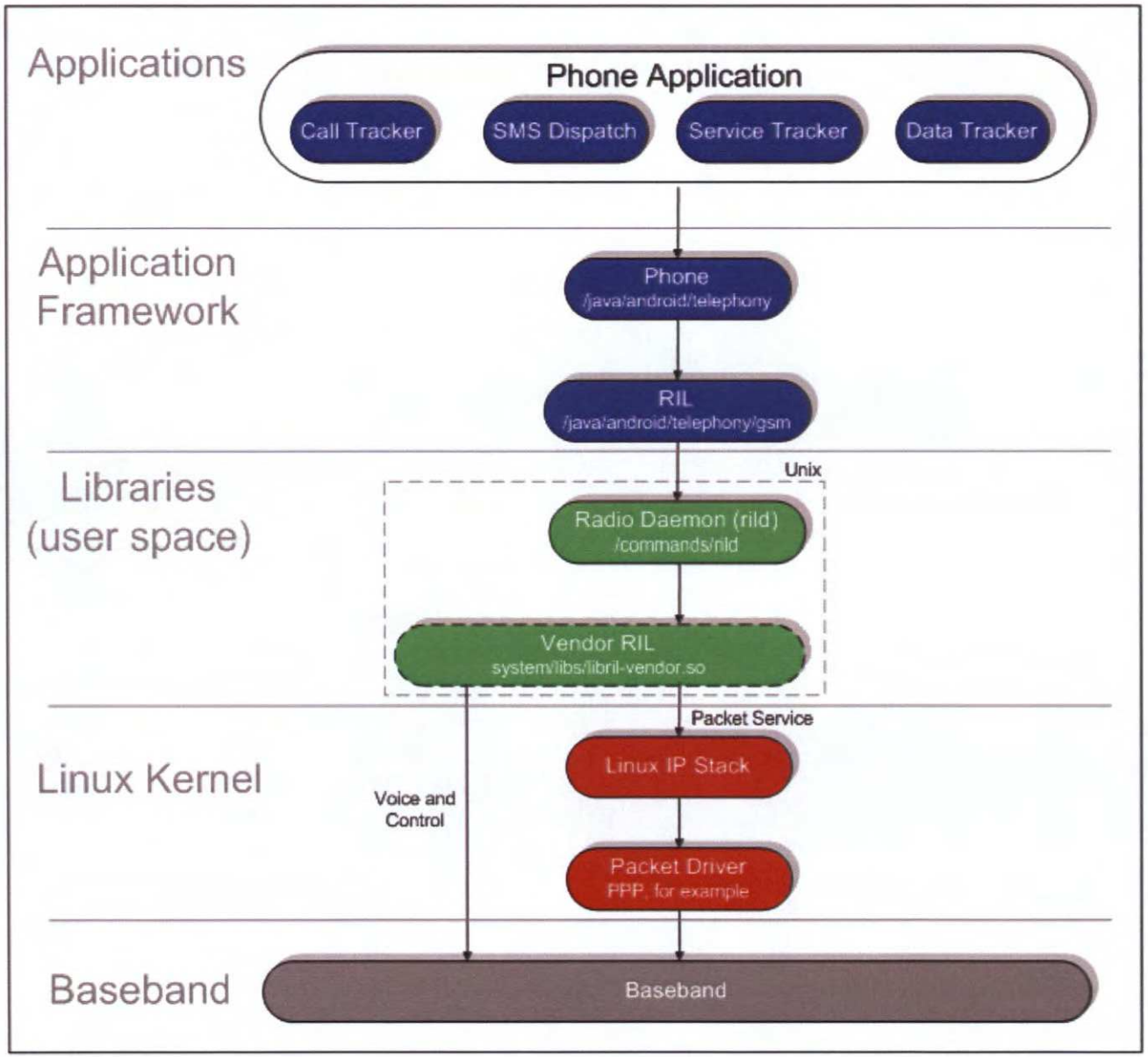
android
open source project

Radio Layer Interface 한글(korean)

Android's Radio Interface Layer (RIL) provides an abstraction layer between Android telephony services (android.telephony) and radio hardware. The RIL is radio agnostic, and includes support for Global System for Mobile communication (GSM)-based radios.

The diagram below illustrates the RIL in the context of Android's Telephony system architecture.

In this document



Solid elements represent Android blocks and dashed elements represent partner-specific blocks.

The RIL consists of two primary components:

- **RIL Daemon**: The RIL daemon initializes the Vendor RIL, processes all communication from Android telephony services, and dispatches calls to the Vendor RIL as solicited commands.
- **Vendor RIL**: The radio-specific Vendor RIL of `ril.h` that processes all communication with radio hardware and dispatches calls to the RIL Daemon (`rild`) through unsolicited commands.

RIL Initialization

Android initializes the telephony stack and the Vendor RIL at startup as described in the sequence below:

1. RIL daemon reads `rild.lib` path and `rild.libargs` system properties to determine the Vendor RIL library to use and any initialization arguments to provide to the Vendor RIL
2. RIL daemon loads the Vendor RIL library and calls `RIL_Init` to initialize the RIL and obtain a reference to RIL functions
3. RIL daemon calls `RIL_register` on the Android telephony stack, providing a reference to the Vendor RIL functions

See the RIL Daemon source code at `//device/commands/rild/rild.c` for details.

System Properties

The following RIL-related system properties are set by the RIL library:

- `ro.ril.ecclist`: list of valid Emergency Call Codes, for example, 911. Values are read from `EF_ECC` on the SIM and possibly supplmented by tables based on operator, network, or manufacturing code.

The following RIL_related system properties are available to the RIL library:

- `ro.ril.hsxpa`: inidcates `hsxpa` support of target network.
- `ro.ril.gprsclass`: inidcates GPRS class of target network.
- `ro.ril.enable.3g.prefix=1`: adds the 3G prefix to the operator name.

RIL Interaction

There are two forms of communication that the RIL handles:

- Solicited commands: Solicited commands originated by RIL lib, such as `DIAL` and `HANGUP`.
- Unsolicited responses: Unsolicited responses that originate from the baseband, such as `CALL_STATE_CHANGED` and `NEW_SMS`.

Solicited

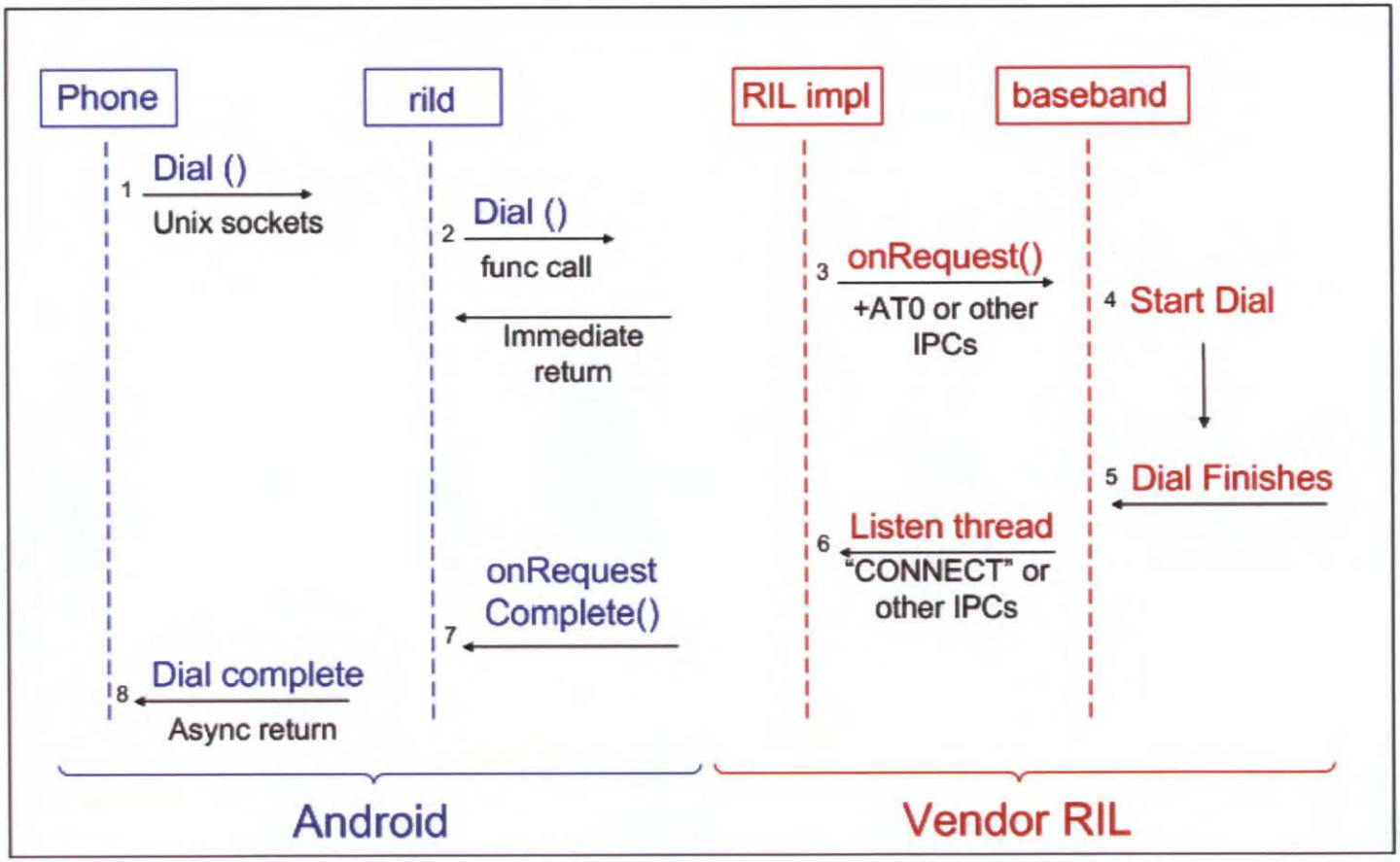
The following snippet illustrates the interface for solicited commands:

```
void OnRequest (int request_id, void *data, size_t datalen, RIL_Token t);
void OnRequestComplete (RIL_Token t, RIL_Error e, void *response, size_t responselen);
```

There are over sixty solicited commands grouped by the following families:

- SIM PIN, IO, and IMSI/IMEI (11)
- Call status and handling (dial, answer, mute…) (16)
- Network status query (4)
- Network setting (barring, forwarding, selection…) (12)
- SMS (3)
- PDP connection (4)
- Power and reset (2)
- Supplementary Services (5)
- Vendor defined and support (4)

The following diagram illustrates a solicited call in Android.



Unsolicited

The following snippet illustrates the interface for unsolicited commands:

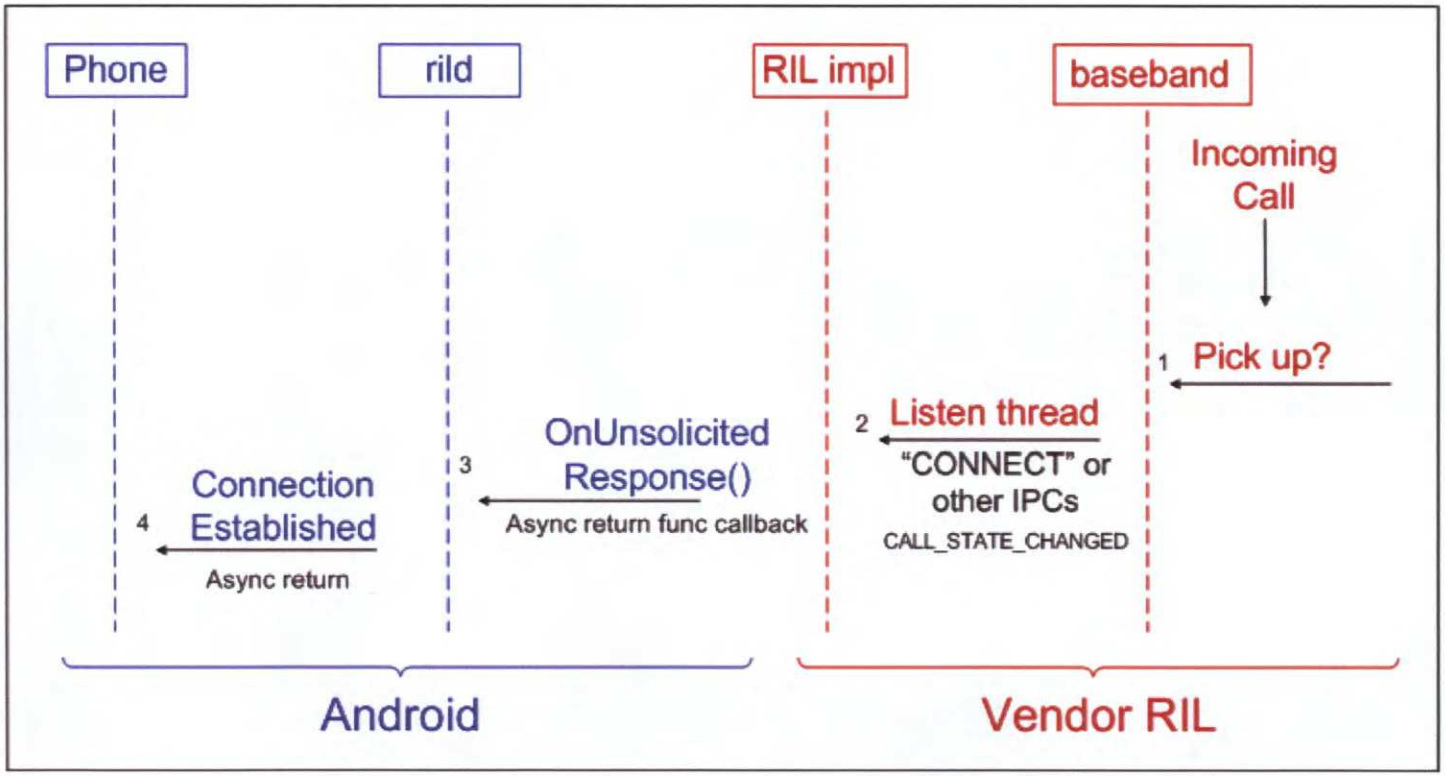
```
void OnUnsolicitedResponse (int unsolResponse, void *data, size_t datalen);
```

There are over ten unsolicited commands grouped by the following families:

- Network status changed (4)
- New SMS notify (3)

- New USSD notify (2)
- Signal strength or time changed (2)

The following diagram illustrates an unsolicited call in Android.



Implementing the RIL

To implement a radio-specific RIL, create a shared library that implements a set of functions required by Android to process radio requests. The required functions are defined in the RIL header (`/include/telephony/ril.h`).

The Android radio interface is radio-agnostic and the Vendor RIL can use any protocol to communicate with the radio. Android provides a reference Vendor RIL, using the Hayes AT command set, that you can use as a quick start for telephony testing and a guide for commercial vendor RILs. The source code for the reference RIL is found at `/commands/reference-ril/`.

Compile your Vendor RIL as a shared library using the convention `libril-<companyname>-<RIL version>.so`, for example, libril-acme-124.so, where:

- **libril**: all vendor RIL implementations start with 'libril'
- **<companyname>**: a company-specific abbreviation
- **<RIL version>**: RIL version number
- **so**: file extension

RIL_Init

Your Vendor RIL must define a RIL_Init function that provides a handle to the functions which will process all radio requests. RIL_Init will be called by the Android RIL Daemon at boot time to initialize the RIL.

```
RIL_RadioFunctions *RIL_Init (RIL_Env* env, int argc, char **argv);
```

RIL_Init should return a RIL_RadioFunctions structure containing the handles to the radio functions:

```
type structure {
        int RIL_version;
        RIL_RequestFunc onRequest;
        RIL_RadioStateRequest onStateRequest;
        RIL_Supports supports;
        RIL_Cancel onCancel;
        RIL_GetVersion getVersion;
}
RIL_RadioFunctions;
```

RIL Functions

`ril.h` defines RIL states and variables, such as `RIL_UNSOL_STK_CALL_SETUP`, `RIL_SIM_READY`, `RIL_SIM_NOT_READY`, as well as the functions described in the tables below. Skim the header file (`/device/include/telephony/ril.h`) for details.

RIL Solicited Command Requests

The vendor RIL must provide the functions described in the table below to handle solicited commands. The RIL solicited command request types are defined in `ril.h` with the `RIL_REQUEST_` prefix. Check the header file for details.

Name	Description
`void (*RIL_RequestFunc) (int request, void *data, size_t datalen, RIL_Token t);`	This is the RIL entry point for solicited commands and must be able to handle the various RIL solicited request types defined in `ril.h` with the `RIL_REQUEST_` prefix. • `request` is one of `RIL_REQUEST_*` • `data` is pointer to data defined for that `RIL_REQUEST_*` • `t` should be used in subsequent call to `RIL_onResponse` • `datalen` is owned by caller, and should not be modified or freed by callee Must be completed with a call to `RIL_onRequestComplete()`. `RIL_onRequestComplete()` may be called from any thread before or after this function returns. This will always be called from the same thread, so returning here implies that the radio is ready to process another command (whether or not the previous command has completed).
`RIL_RadioState (*RIL_RadioStateRequest)();`	This function should return the current radio state synchronously.
`int (*RIL_Supports)(int requestCode);`	This function returns "1" if the specified `RIL_REQUEST` code is supported and 0 if it is not.
`void (*RIL_Cancel)(RIL_Token t);`	This function is used to indicate that a pending request should be canceled. This function is called from a separate thread--not the thread that calls `RIL_RequestFunc`. On cancel, the callee should do its best to abandon the request and call `RIL_onRequestComplete` with `RIL_Errno CANCELLED` at some later point. Subsequent calls to `RIL_onRequestComplete` for this request with other results will be tolerated but ignored (that is, it is valid to ignore the cancellation request). `RIL_Cancel` calls should return immediately and not wait for cancellation.
`const char * (*RIL_GetVersion) (void);`	Return a version string for your Vendor RIL

The vendor RIL uses the following callback methods to communicate back to the Android RIL daemon.

Name	Description
`void RIL_onRequestComplete(RIL_Token t, RIL_Errno e, void *response, size_t responselen);`	• `t` is parameter passed in on previous call to `RIL_Notification` routine. • If `e` != SUCCESS, then response can be null and is ignored • `response` is owned by caller, and should not be modified or freed by callee • `RIL_onRequestComplete` will return as soon as possible
`void RIL_requestTimedCallback (RIL_TimedCallback callback, void *param, const struct timeval *relativeTime);`	Call user-specified callback function on the same thread that `RIL_RequestFunc` is called. If `relativeTime` is specified, then it specifies a relative time value at which the callback is invoked. If `relativeTime` is NULL or points to a 0-filled structure, the callback will be invoked as soon as possible.

RIL Unsolicited Commands

The functions listed in the table below are call-back functions used by the Vendor RIL to invoke unsolicited commands on the Android platform. See `ril.h` for details.

Name	Description
`void RIL_onUnsolicitedResponse(int unsolResponse, const void *data, size_t datalen);`	• `unsolResponse` is one of `RIL_UNSOL_RESPONSE_*` • `data` is pointer to data defined for that `RIL_UNSOL_RESPONSE_*` • `data` is owned by caller, and should not be modified or freed by callee

↑ Go to top

android developers

package
android.webkit

Provides tools for browsing the web.
The only classes or interfaces in this package intended for use by SDK developers are WebView, BroswerCallbackAdapter, BrowserCallback, and CookieManager.

Interfaces

DownloadListener	
GeolocationPermissions.Callback	Callback interface used by the browser to report a Geolocation permission state set by the user in response to a permissions prompt.
PluginStub	This interface is used to implement plugins in a WebView.
ValueCallback<T>	A callback interface used to returns values asynchronously
WebChromeClient.CustomViewCallback	A callback interface used by the host application to notify the current page that its custom view has been dismissed.
WebIconDatabase.IconListener	Interface for receiving icons from the database.
WebStorage.QuotaUpdater	Encapsulates a callback function to be executed when a new quota is made available.
WebView.PictureListener	*This interface is deprecated. This interface is now obsolete.*

Classes

CacheManager	*This class is deprecated. Access to the HTTP cache will be removed in a future release.*
CacheManager.CacheResult	*This class is deprecated. Access to the HTTP cache will be removed in a future release.*
ConsoleMessage	Public class representing a JavaScript console message from WebCore.
CookieManager	CookieManager manages cookies according to RFC2109 spec.
CookieSyncManager	The CookieSyncManager is used to synchronize the browser cookie store between RAM and permanent storage.
DateSorter	Sorts dates into the following groups: Today Yesterday seven days ago one month ago older than a month ago
GeolocationPermissions	This class is used to get Geolocation permissions from, and set them on the WebView.
HttpAuthHandler	HTTP authentication request that must be handled by the user interface.
JsPromptResult	Public class for handling javascript prompt requests.
JsResult	
MimeTypeMap	Two-way map that maps MIME-types to file extensions and vice versa.
SslErrorHandler	SslErrorHandler: class responsible for handling SSL errors.
URLUtil	
WebBackForwardList	This class contains the back/forward list for a WebView.
WebChromeClient	
WebHistoryItem	A convenience class for accessing fields in an entry in the back/forward list of a WebView.
WebIconDatabase	Functions for manipulating the icon database used by WebView.
WebResourceResponse	A WebResourceResponse is return by `shouldInterceptRequest(WebView, String)` and contains the response information for a particular resource.
WebSettings	Manages settings state for a WebView.
WebStorage	Functionality for manipulating the webstorage databases.
WebStorage.Origin	Class containing the HTML5 database quota and usage for an origin.
WebView	A View that displays web pages.
WebView.HitTestResult	
WebView.WebViewTransport	Transportation object for returning WebView across thread boundaries.
WebViewClient	
WebViewDatabase	
WebViewFragment	A fragment that displays a WebView.

Enums

ConsoleMessage.MessageLevel	
WebSettings.LayoutAlgorithm	Enum for controlling the layout of html.
WebSettings.PluginState	The plugin state effects how plugins are treated on a page.
WebSettings.RenderPriority	
WebSettings.TextSize	*This enum is deprecated. Use `setTextZoom(int)` and `getTextZoom()` instead.*
WebSettings.ZoomDensity	Enum for specifying the WebView's desired density.