

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD., and
SAMSUNG ELECTRONICS, AMERICA, INC.,
Petitioners,

v.

CERENCE OPERATING COMPANY,
Patent Owner.

Case No. IPR2024-01267
U.S. Patent No. 9,026,428

**DECLARATION OF ERIC KOSKINEN IN SUPPORT OF
PATENT OWNER'S RESPONSE TO THE PETITION**

I, Eric Koskinen, hereby declare as follows:

I. Introduction

1. I am over the age of eighteen (18) years and otherwise competent to make this declaration.

2. I have been retained as an expert witness on behalf of Patent Owner for the above-captioned *inter partes* review (“IPR”). I understand that the petition for *inter partes* review involves U.S. Patent No. 9,026,428 (“the ’428 Patent”).

3. I make this declaration based on my personal knowledge, educational background and training, consideration of the materials I discuss herein, and my expert opinions.

4. My work in this case is being billed at my normal hourly consulting rate of \$700, with reimbursement for actual expenses. My compensation is not related to the outcome of any proceeding involving these Asserted Patents. I have no personal interest in the outcome of the case.

5. In preparing this Declaration, I have reviewed and considered the ’428 Patent, the Petition, the declaration submitted by Mr. Ward in this proceeding, and each document cited in my declaration.

II. Background and Qualifications

6. My qualifications for forming the opinions given in this declaration are summarized here and are addressed more fully in my curriculum vitae, which is submitted as Exhibit 2008.

7. I currently serve as a Tenured Associate Professor at Stevens Institute of Technology and, for several years, have served as a consultant, providing software-related services to law firms and corporations. Previously, I was a Lecturer/Researcher at Yale and a Professor at New York University. I completed an undergraduate degree (with Highest Honors) in Computer Science and Physics from The College of William and Mary, received a Master's degree in Computer Science from Brown University, and completed a Ph.D. in Computer Science from the University of Cambridge.

8. I have been working in the fields of computer science and software engineering since 2001, including the years 2005-2013 when I completed the Master's degree and the Doctorate mentioned previously.

9. I have work experience in all phases of commercial software design and operation, including requirements engineering, software architecture, software engineering, software deployment, and software project management. This work has included writing and developing software and browser-based web applications for Amazon.com (IMDb), Microsoft, and IBM. I have developed User Experiences

(UXs) and User Interfaces (UIs) for a variety of web applications including public- and internal- facing web applications. My experience also includes the development and deployment of countless multi-tier and distributed interacting systems.

10. I am a computer scientist, with expertise in the areas of programming languages, formal methods, concurrency, and systems. I have authored over 40 top-tier academic publications over the past 16 years and been cited over 2,000 times. I have served in international academic leadership positions including top-tier Steering Committees (e.g., ACM/IEEE Logic in Computer Science) and Program Committees (e.g., ACM Principles of Programming Languages (POPL), ACM/SIGPLAN Programming Language Design & Implementation (PLDI), and the International Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA)).

11. I have been awarded funding in excess of \$6M from a variety of federal funding agencies, including three National Science Foundation (NSF) Awards, one DARPA Award, and two Awards from the Office of Naval Research (ONR).

12. I have authored top-tier conference publications in the field of programming languages, compilers, and automatic software verification, including articles in ACM Principles of Programming Languages (e.g., two papers in POPL 2023), ACM/SIGPLAN Programming Language Design & Implementation (PLDI 2017), and recent articles published in Object-Oriented Programming, Systems, Languages,

and Applications (OOPSLA 2019, 2020, 2022, and 2024) and Computer Aided Verification (CAV 2021). My publications have also received awards, e.g., a Distinguished Paper award at POPL 2023.

13. I have also authored top-tier conference publications in the field of systems, concurrency, and distributed systems (e.g., EuroSys 2008, PPOPP 2008, SPAA 2008a, SPAA 2008b, PODC 2017, and the Journal of Distributed Computing 2020), including recent works on multicore parallelization (OOPSLA 2022) and distributed ledgers, i.e., blockchains (PODC 2017 and Journal of Distributed Computing 2020). I have also served on program committees in these areas, including the International Conference On Principles Of Distributed Systems (OPODIS 2012), the ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2012), the ACM Symposium on Principles of Distributed Computing (PODC 2018), Principles and Practice of Parallel Programming (PPOPP 2018), and the International Conference on Networked Systems (NETYS 2023).

14. I have taught computer science courses on Data-Structures, Algorithms, Object-Oriented Programming, Compilers, and Societal Impacts of Information Technology. I have supervised PhD, Master's, and undergraduate students who have gone on to jobs in top companies or academic positions.

15. I also have an extensive background in software systems. I have developed a wide variety of software applications, including desktop applications, web-based

applications, network applications, and mobile applications. I have worked on a wide variety of computer infrastructure systems such as web servers, database servers, transactional systems, network systems, and APIs.

16. From 2003 to 2005, I was a Software Development Engineer at Amazon.com, where I developed industrial software applications including custom web servers, internal and external services, web-based applications, mobile applications and e-commerce applications, interfacing with internal payment systems.

17. As a Software Engineer at Amazon, I worked on the Internet Movie Database (IMDb). At IMDb I was involved in developing applications that constructed and queried high-performance databases, including Berkeley DBs (key/value Hash Tables), CDBs, MySQL database, PostgreSQL databases, high-performance Web Servers, and custom software abstractions for internal representation.

18. I have delivered dozens of invited lectures at universities and institutions around the world, including Oxford University, Massachusetts Institute of Technology, Cornell University, Princeton University, Yale University, Columbia University, Rice University, Tokyo University, Université Cité de Paris, ETH Zurich, Ecole Polytechnique Federale de Lausanne, University of Johannesburg South Africa, IBM, Microsoft, NEC SRI, and Oracle.

III. Legal Principles

19. I am not an attorney and do not offer opinions on the law in this declaration. For the purposes of this declaration, I have been informed about certain aspects of the law that are relevant to forming my opinions. My understanding of the law is as follows.

A. Person of Ordinary Skill

20. I have been informed that a person having ordinary skill in the art (i.e., a POSITA) is a hypothetical person who is presumed to have known the relevant art at the time the invention was made. I understand that the condition “at the time the invention was made” is imposed to rule out impermissible hindsight. I also understand that an expert is to analyze the prior art from the perspective of a person of ordinary skill in the art and not simply to provide his/her own personal conclusions. I further understand that the factors that may be considered in determining the level of ordinary skill include: (1) the problems encountered in the art; (2) the prior art solutions to the problems encountered in the art; (3) the rapidity of innovation; (4) the sophistication of the technology; and (5) the education level of active workers in the field. I understand that these factors need not all be considered for the analysis and that one or more of these factors may control.

21. I understand that Petitioners and their expert have proposed that a POSITA in the context of the '428 patent “would have had at least a Bachelor’s of Science

Degree (or equivalent) in electrical engineering, computer science, or computer engineering, or a related technical field and two years of experience working with handwriting recognition, user interfaces incorporating on-line recognition technologies, or related technical fields.” *See, e.g.*, Ex. 2001 (Ward Decl.), ¶ 26. I further understand that Petitioners and their expert assert that “A greater amount of professional education could compensate for fewer years of work experience, and vice versa.” *Id.* For the purposes of this Declaration I accept and agree with this POSITA definition.

22. As of October 15, 2012, I met, and in a number of aspects exceeded, the qualifications of a POSITA as defined herein. Additionally, all of my opinions in this declaration are from the perspective of a POSITA as defined here during the relevant timeframe.

B. Claim Construction

23. I understand that a claim construction inquiry begins and ends in all cases with the actual words of the claim. Apart from the written description and the prosecution history, the claims themselves provide substantial guidance as to the meaning of particular terms. I further understand that the context in which a term is used in the asserted claim can be highly instructive. The patent specification can also shed light on the meaning of claim terms.

24. I understand that, when conducting a claim construction inquiry, courts are not required to construe every limitation present in a patent's asserted claims. I further understand that where a term is used in accordance with its plain meaning, the court should not recharacterize it using different language.

25. I understand that there is a "heavy presumption" that claim terms carry their full ordinary and customary meaning unless the accused infringer can show that the patentee expressly relinquished claim scope. The ordinary and customary meaning of a claim term is the meaning that the term would have to a POSITA at the time of the invention.

26. I understand that courts generally do not import limitations into claims from examples or embodiments appearing only in a patent's written description, even when a specification describes very specific embodiments of the invention or even only a single embodiment. Similarly, I understand that statements during patent prosecution do not limit the claims unless the statement is a clear and unambiguous disavowal of claim scope.

27. I further understand that a preamble of a patent claim is typically not limiting on the scope of the claim if the body of a claim defines a structurally complete invention. I understand that a preamble may be found to be limiting if it supplies the antecedent basis to a claim term in the claim's body, if it recites essential structure or steps, or is necessary to give life, meaning, or vitality to the claims. I understand

that language in a claim preamble which merely states a purpose or intended use of the invention is not limiting.

C. Burden of Proof

28. I understand that Petitioners bear the burden of establishing that the challenged claims are invalid by clear and convincing evidence.

D. Anticipation

29. I am informed that for a claim to be anticipated, every limitation of the claimed invention must be found in a single prior art reference, either expressly or inherently, arranged as in the claim. I understand that the words in a reference are to be construed in light of the relevant surrounding circumstances in each case, and a reference in any event is good only for that which it discloses.

30. I further understand that anticipation cannot be established by multiple, distinct teachings that the artisan might combine to achieve the claimed invention. Additionally, I understand that when a reference does not disclose that the drawings are to scale and is silent as to dimensions, an accused infringer cannot infer dimensions to demonstrate invalidity. Finally, I understand that an element may be inherently disclosed only if it is necessarily present, not merely probably or possibly present, in the prior art.

E. Obviousness

31. I understand that the grounds for institution in this case only include allegations that challenged Claims 1-3, 7-9 and 12-13 are obvious. I also understand that the scope of issues to be considered in an IPR are limited to those grounds set forth in the IPR Petition.

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

33. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be some motivation for a POSITA to modify the single prior art reference, or combine two or more references, in a manner that provides the elements of the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the

reference(s) themselves, the knowledge or “common sense” of a POSITA, or from the nature of the problem to be solved, and this reason may be explicit or implicit from the prior art as a whole. I also understand that where there is a reason to modify or combine the prior art to achieve the claimed invention, that it is also important that a POSITA would have a reasonable expectation of success. I have been informed that, under the law, the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.

34. It is my understanding that obviousness must be determined as of the date of the invention (*i.e.*, the effective filing date or priority date of the patent). Thus, in considering the reason or motivation to combine references, it is essential to avoid using hindsight. I also understand that the problem examined when considering obviousness may be a problem that would have motivated a POSITA at the time that the invention was made, not the specific problem solved by the invention. I understand that defining the problem in terms of its solution reveals improper hindsight in the selection of the prior art relevant to obviousness. Further, I understand that an overly narrow statement of the problem can represent a form of hindsight, because an inventive contribution may lie in defining the problem in a new, revelatory way. Similarly, I understand that an assertion that a person of ordinary skill could combine the references, rather than that they would have been

motivated to do so, is an impermissible form of hindsight. Moreover, I understand that knowledge of a problem and motivation to solve it are entirely different from reason to combine particular references to reach the particular claimed method.

35. I am informed that the analysis of the reason to combine prior art should be explicit. Additionally, I understand that an obviousness analysis must include articulated reasoning with rational underpinnings to support the conclusion of obviousness. I understand that providing a full explanation of the reason to combine the references and the reasonable expectation of success is a necessary component of the obviousness inquiry, as 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. However, I understand that conclusory statements fail to adequately explain why a person of ordinary skill would have a motivation to combine the potential prior art references.

36. I am informed that common sense, common wisdom, and common knowledge may be used to support a reason to combine, so long as the use of common sense in the analysis is explained with sufficient reasoning. But I understand that common sense can only be used to supply a limitation missing from the prior art where the technology is unusually simple and straightforward.

37. I am also informed that evidence suggesting reasons to combine cannot be viewed in a vacuum apart from evidence suggesting reasons not to combine. In

analyzing motivation, both advantages and disadvantages must be considered. Teaching away, for example, is a statement in the prior art that either (i) discourages one from following the path of the claimed invention, or (ii) encourages one to follow a path that diverges from the path of the claimed invention. The concept of teaching away bears directly on whether there was a reason to combine prior art. Known disadvantages in old devices which would naturally discourage a search for new inventions may also be taken into account in determining obviousness.

IV. Claim Construction

38. For the purposes of my analysis, I have given all the claim terms their plain and ordinary meaning, as would have been understood by a person having ordinary skill in the art at the time of the alleged invention, which I understand is October 15, 2012, having taken into consideration the language of the claims, the specification, and the prosecution history of record. I reserve the right to respond to any construction questions that may be raised in this matter.

V. Technology Background

A. The '428 Patent

39. The '428 patent seeks to solve problems with prior art character recognition methods which did not provide contextual analysis for character and word recognition, and which further required the user to press a button or take some other action to determine when one character or word ends, which “decreases the speed of

character/letter input by the users.” ’428 patent at 1:31-42. As the ’428 patent explains:

The recognition engine or process provides an indication or output as to when one or more characters can be committed and sent to the application, as opposed to requiring a user to input of a space gesture, select a commit button, or other user input. In other words, the system provides an automatic and quicker recognition system.

Id. 2:54-59. This analysis of when one or more characters can be committed to an application without further user input is based on innovative contextual analysis, using “elaborate linguistic knowledge” to assist in performing “identification of e.g., the end of one word and the beginning of the next without the need to detect a space gesture between two words.” *Id.* 2:59-62. Specifically, “the use of linguistic tools with an intermediary buffer allows the system to use a ‘history’ of characters input by a user to increase the speed and accuracy of handwritten characters input to the system” which allows the system to “automatically decide when to clear this intermediate buffer by analyzing the context of two or more characters or words.” *Id.* 2:59-62.¹

¹ The ’428 patent explains that “[c]ontext information is any information that may help inform the recognition engine of the users intent, and may include information such as sentence or word-pair context, language context, application-specific context (e.g. data fields/tags), user-specific context, location, time of day etc.” *Id.* 12:34-39.

40. At a high level, the '428 patent describes a system and method that enables users to enter characters, letters, texts or other related input onto mobile devices, with increased speed and accuracy, and thus reduce input errors. *Id.* 2:30-34 (“As will be explained in further detail below, the technology described herein relates to systems and methods for increasing the speed and accuracy with which a user can enter characters or text into devices, such as into mobile devices having touch screens.”).

41. More specifically, the '428 patent describes a system and method for a mobile device with a touch screen to receive character inputs from a user, and with the use of a programmed processor receiving the inputs, and disambiguating the inputs, in order to be able to present corresponding character sequence choices to the user.

42. The system and method of the patent are centered around a “linguistically aided autocommit system for handwriting recognition user interfaces (UIs).” *Id.* 2:39-41. A user can enter the character through “one or a series of strokes or input features (e.g., using a finger and/or a stylus on a touch screen interface).” *Id.* 4:36-38. The input is received through the touch screen interface of a mobile device.

43. Once this input from the user is received, the process will perform character recognition on the input. One way that character recognition can be performed is through “comparing a best match between one or more strokes/gestures input at a touch-sensitive input device to a database of recognized characters.” *Id.* 4:41-43.

Once the recognition engine has made a best match between the input characters and the database, then the system and method will store the “highest-candidate character sequence text” (*id.* 6:4) to a “temporary editable character buffer” (*id.* 4:44), which may be presented to the user.

44. Specifically describing the role of the recognition engine, it may, as an example, “receive a series of notifications from the operating system 611 of finger down events (when a user depresses a point on the touch screen), finger move events (e.g., when a user moves his finger while depressing the screen), and finger up events (e.g., when a user lifts their finger from the touch screen).” *Id.* 12:9-14. In the setting of a touchscreen mobile device the input could be a finger, and the recognition engine can also recognize a stylus, or anything that a user may use to actuate the touch screen.

45. The received notifications “may include a time series of position data, e.g., an X-Y set of coordinates (x, y) with a timestamp t (i.e., the coordinates (x, y, t), which reflect the user’s tactile input to the touch screen 603.” *See, e.g.,* ’428 patent at 12:16-20. The input data to the recognition engine may also include multitouch inputs: “(e.g., additional sets of simultaneous coordinates (x, y, t) representing a second finger simultaneously contacting the touch screen), audio inputs (e.g., voice inputs), visual inputs, accelerometer inputs, location data, sensor data, etc.” *Id.* 12:27-30.

46. There may also be more dynamic information from the OS that will provide the recognition engine more context data, such as “sentence or word-pair context, language context, application-specific context (e.g. data fields/tags), user-specific context, location, time of day.” *Id.* 12:34-39.

47. Thus, the recognition engine may “include a character selection submodule 618, a sequence list selection submodule 620, a keyboard mapping database 622, a vocabulary module 624, a user context database 626, an application context database 628 and a disambiguation parameter database 630.” *Id.* 12:59-63. An illustration of the recognition engine can be found in Fig 6, replicated below. The recognition engine is able to analyze the sequence of characters in the buffer “for a match with certain rules, a database or dictionary” *Id.* 4:53-54.

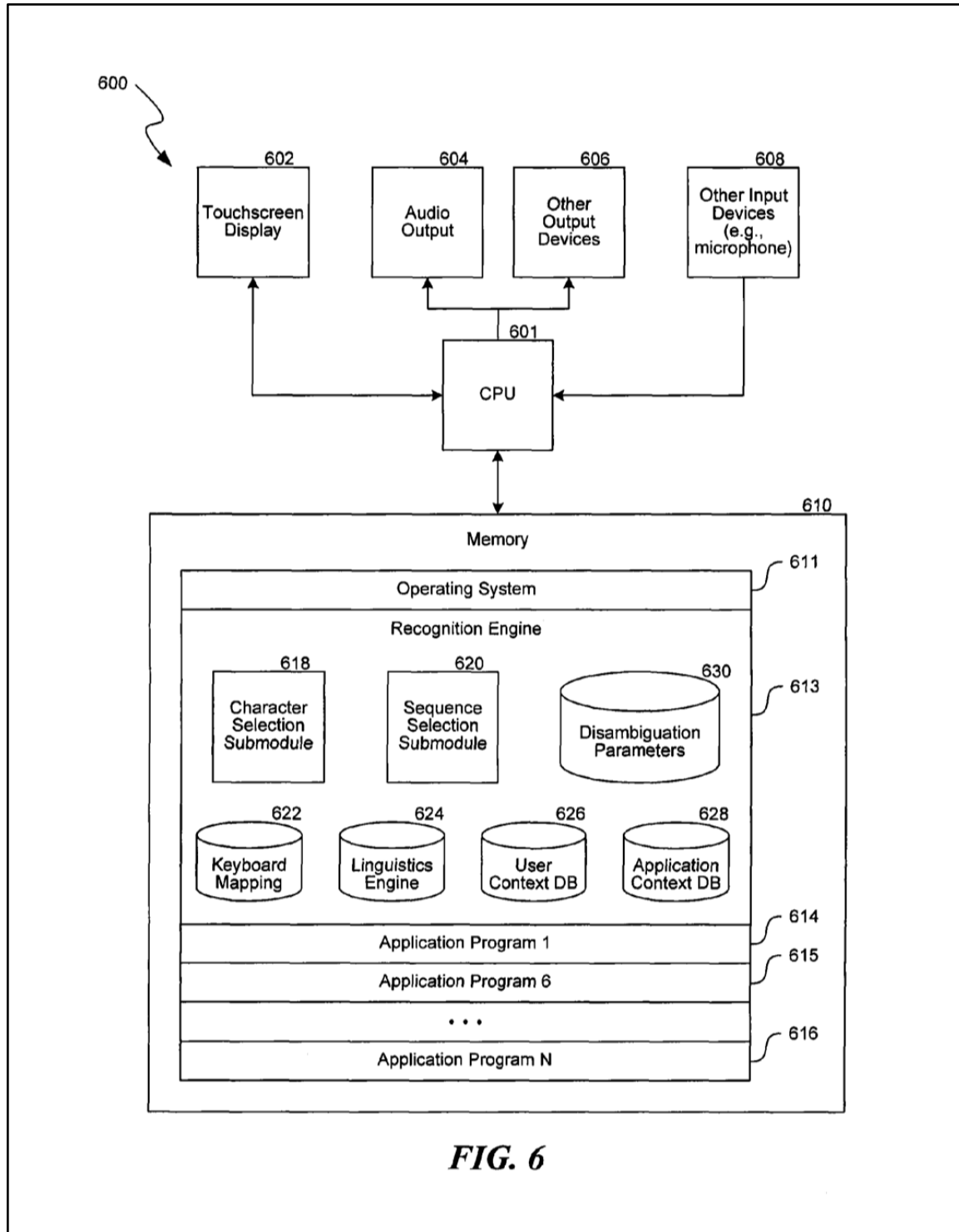


FIG. 6

48. Moreover, the recognition engine includes the editable or changing buffer, and this is where the engine sends the highest-ranking characters candidates. *Id.*

2:41-44. This candidate sequence in the buffer could be modified, as per the four different ways: 1) The user can continue to enter more characters, which would cause the recognition engine to continue updating the buffer; 2) the user is able to select a candidate from the candidate list, which might be achieved with some gesture or specialized button; 3) the user can modify the text in the application directly; and finally, 4) the system's auto-commit, where the engine uses "linguistics or other heuristic rules to identify that part of the current content can be flushed from the engine buffer." *Id.* 3:44-46.

49. This buffer might display to the user the possible characters, or a recognition candidate list, which is "a list that contains information of a hypothesis from the engine based on the content of this buffer." *Id.* 3:13-14. As indicated in Fig. 3D, the recognition candidate list is "Rat/Rar/Rather/Rate/Art/Rati." *Id.* FIG. 3D.

50. If the system is unable to identify a sequence in the buffer, e.g. if no words can be found in the dictionary to match the first sequence of characters, then this unidentifiable sequence may also be displayed to the user in the buffer. Moreover, if some of the sequence in the buffer was identified, that identified section could be committed to the application and flushed from the buffer. *Id.* FIG. 2.

51. The '428 patent's recognition engine determines that "the user has had several iterations to edit the first characters 414 and determines that the characters 414 form a recognized sequence 442 that the user is unlikely to edit. As a result, the

recognition engine removes the recognized sequence 442 from the buffer...The sequences 442 is removed from the text display and sent to the application and 444 is displayed in the text display 307.” *Id.* 8:58-66.

52. Therefore, the recognition engine can determine when a sequence of characters may be output to the application, leaving and clearing the buffer. This results in the characters no longer being editable or modifiable by the recognition engine. *Id.* 2:49-53. The editable buffer is not required to be a user interface element, but rather it is a buffer where the recognition engine can modify the content. *Id.* 3:14-16.

53. Thus, the ’428 patent recognition engine provides an indication when one or more characters are committable and are able to be sent to the application. This does not require the user to input a space, or select a commit button, or any other user input: “[a] user can continually enter characters without spaces, and the system can automatically parse words from the string of input characters” *Id.* 4:64-67.

54. Thus, once the characters are no longer being altered or edited in the buffer, then “the candidate characters and/or words are committed to an application and the buffer is cleared of these candidates characters and/or words.” *Id.* 2:44-46. This is once the system determines that “at least a portion of the sequence in the temporary editable character buffer is a committable sequence.” *Id.* 4:48-49.

55. Due to the innovative contextual analysis at each stage in the recognition process, the system provides an automatic and quicker recognition system than those disclosed in the prior art.

56. The system is able to analyze two or more words, as well as characters individually and in context with other characters, including Asian characters. The '428 patent also refers to the ability to receive characters that are not only alphabetic, but also “characters or symbols that represent a phonetic or sub-word component, including Japanese kana, Korean jamos, and Chinese zhuyin.” *Id.* 6:21-23; *see also* Fig. 5A-5D.

57. The '428 patent improves sequence selection with the “linguistics engine 624 [that] provides one or more dictionaries that define a set or words, or logical groups of words.” *Id.* 13:34-36. For example, “an American-English” or “American-English-Msg” dictionaries would refer to commonly used phrases, groups of words, even emoticons that are used by Americans over text and multimedia messaging applications. Additionally, the '428 patent could include, in some embodiments “Chinese, Japanese and/or Korean dictionaries.” *Id.* 13:42-43.

58. Moreover, the '428 patent discloses that a “user context data 626 may include any historical information relating to the user’s prior character and/or sequence selection behavior, such as previous words utilized by the user, the frequency of words entered by the user, and common phrases employed by the user.” *Id.* 14:27-

31. This user context data reflects what particular languages a user may use. “For example, if the system utilizes both Chinese and English language dictionaries for a bilingual user, the user context database 626 may track the relative frequency with which the user selects Chinese words versus English words in order to determine the users “primary language,” and then words from the dictionary may be preferred or presented first over words from the other dictionary.” *Id.* 14: 39-45.

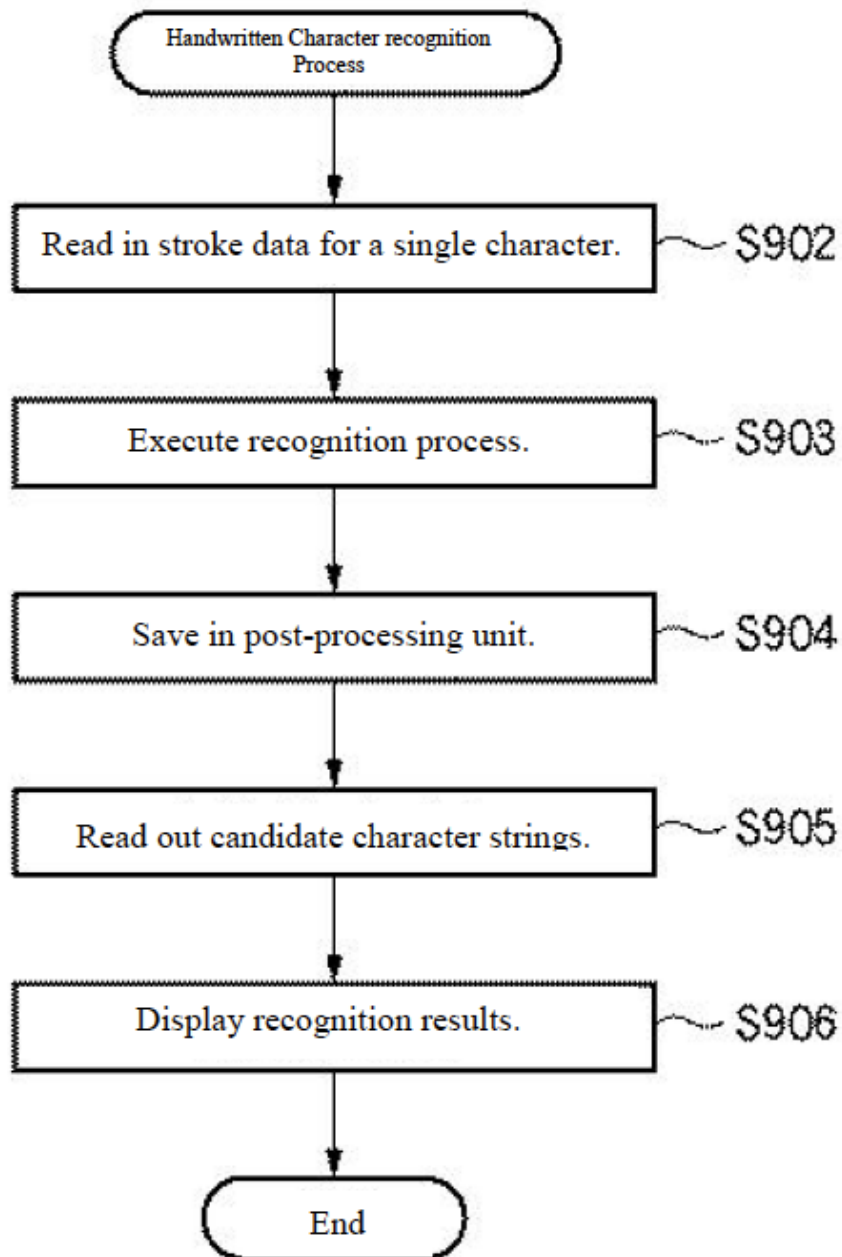
B. Arai

59. *Arai*² is a Japanese patent application which discloses a system and method for recognizing handwritten Japanese characters. As the specification explains, the “invention relates to an information processing device and controlling method and program thereof for inputting stroke data of a handwritten character and performing character recognition, and executing post-processing in relation to the character recognition result.” *See Arai* [0001]. *Arai* explains that its purpose is to “enable efficient and accurate execution of a handwritten character recognition.” *Id.* [0006]. To achieve this purpose, the systems and methods of *Arai* comprise “an information processing device for inputting handwritten character stroke data, recognizing a character, and executing post-processing on the character recognition result.” *Id.* [0007].

² JP Pat. App. JP 2006-323502 A, published Nov. 30, 2006 (Ex. 1005 (certified English translation), “*Arai*”).

60. The *Arai* character recognition steps are shown in Figure 8 of the patent (reproduced below). As this figure shows, the *Arai* character recognition process: 1) performs initial character recognition (steps S902-S903); 2) performs post-processing using a common dictionary lookup to see whether an entered string of characters matches a known word (step S904); and 3) then displays the results to the user for editing (steps S905-S906). *See* FIG. 8 (below); *see also id.* [139]-[154].

[FIG. 8]



61. A so-called post-processing continuation determination unit in *Arai* relies on user input, as the individual process types evaluate whether to stop post-processing or to append a character to the character string. Fig. 4 of *Arai* (reproduced below)

outlines the ‘process type’, which is what a user might do that would terminate the post-processing continuation:

[FIG. 4]

Process Type	Post-processing Continuation
Candidate Change process	Post-processing Character String Continuation Termination
Handwritten Character Input	Post-processing Character String Continuation
Character Cutoff Operation	Post-processing Character String Continuation
Kana-Kanji Conversion Process Instruction	Post-processing Character String Continuation Termination
Single-Character Deleting Process	Post-processing Character String Continuation
Space Input	Post-processing Character String Continuation Termination
Enter Input	Post-processing Character String Continuation Termination
Switch Application	Post-processing Character String Continuation

62. For example, as shown in Fig. 4 above, a user might select a different candidate, Kana-Kanij conversion button, or continue to input handwritten characters, input a space, or delete a character. All of these inputs would either terminate the post-processing character string continuation or continue the post-processing character strings. *Id.* [0085]-[0091].

63. *Arai* further requires the user to confirm that the character that is displayed is the character desired by the user: “[I]f the character displayed by the handwritten character input screen 53 is the character that is desired by the user, the user confirms this character to insert.” *Id.* [0075]. *Arai* further requires hitting the Enter Key to send a word (or words) to the application. *Id.* [0091].

C. *Fenwick*

64. *Fenwick*³ is a U.S. Patent directed to “[a] pattern recognition system having inter-pattern spacing correction” which allows for “correcting spacing errors” that may occur when inputting text into system. *Fenwick* at Abstract; *id.* 1:6-10. As the patent explains:

In a typical pattern recognition system, various pattern groups (which may be as small as a single individual pattern) are formed from the pattern data by measuring the inter-pattern spacing distance between adjacent individual patterns. Adjacent patterns which are spaced apart from each other by less than a threshold inter-pattern distance are typically combined together to form one pattern group. Adjacent patterns spaced apart by more than the threshold amount are assumed to correspond to next adjacent pattern groups.

Id. 1:32.

65. *Fenwick* further explains that “a significant problem” with prior pattern recognition techniques “is that a relatively large inter-pattern distance between adjacent patterns can sometimes be mistakenly identified as a space between pattern groups, resulting in a single pattern group being split into separate, and generally incomprehensible, parts.” *Id.* 1:33-38. For example, “sloppy handwriting” can create “irregular inter-character spacing” which “may result in the letters within a word being incorrectly recognized as belonging to separate words.” *Id.* 1:38-45. As *Fenwick* further explains, “Inter-character spacing problems are of particular

³ U.S. Pat. No. 5,161,245, issued Nov. 3, 1992 (Ex. 1006, “*Fenwick*”).

significance in handwriting recognition systems where writing styles and writing clarity can vary substantially.” *Id.* 1:45-48.

66. To solve these spacing problems, *Fenwick* employs a system which: 1) stores each input character in one or more buffers in memory; 2) compares the last entry of each buffer against a dictionary of known words; 3) copies the last entries which match words to new buffers, while placing a space after the last entry in the old buffer; 4) deletes a buffer when a last entry from that buffer does not at least match the beginning of a known word in the dictionary; 5) repeats steps (1) through (4) until all the buffers have been deleted or until an end character, such as a carriage return or period has been stored in each buffer; 6) again compares the last entries from each buffer to the dictionary and deletes any buffer having a last entry which is not a known word in the dictionary or which does not end with a space followed by an end character; and then 7) saves and/or outputs the remaining words. *Id.* Abstract; cl. 1.

D. *Baker*

67. *Baker* is a European Patent that “relates generally to methods and systems for word recognition, and more particularly, to methods and systems that employ context and meaning to recognize a word, or a series of words.” *Baker* [0001]. As the patent explains:

Accordingly, the invention, in one aspect, provides systems and methods that perform semantic analysis by analyzing language knowledge represented as, in one example, raw text, and employing the language knowledge for dynamically computing the semantic similarity between the words in a “sample passage” and the words in the “current passage” being analyzed.

Id. [0024].

68. Thus, “current passage” is the described by Baker as “a passage of words or phrases that are being analyzed by the system or method and that includes at least one word to be recognized” (*id.* [0030]) and “current word” is “understood to encompass the word or phrase on which the system or method is operating to estimate a probability distribution given the other words which are known, or hypothesized to be in the current passage” (*id.* [0031]).

69. *Baker* differs from ’428 patent, as it seeks to “recognize an unrecognized or ambiguous word that occurs **within a passage of words**...The system can select the best choice word by using the choice word to extract from a reference source sample passages of text that relate to the choice word.” *Id.* [0025]. Thus, *Baker* is dependent on the other words within the passage in order to determine the unknown word:

For example, the system can select the dictionary passages that defines the choice word or, if the choice word represents a phrase of two or more words, the system, can select a dictionary passage that defines one of the words in the phrase. The system then compares the selected passage to the current passage, and generates a score that indicates the likelihood that the choice word would occur within that passage of text. The system can select the choice word with the best score to substitute into the passage.

Id. [0025].

70. The choice list generator “can generate for each current word 38 a list of one or more choice words 20” (*id.* [0063]), which is significantly different from ’428 that recognizes handwritten characters and from those sequence of characters the recognition engine provides a candidate list.

71. Moreover, *Baker* is geared towards speech recognition (*see id.* [0004], [0006], [0032]; [0036]; [0046]; [0068]; [0078]; [0083]-[0085]; [0088]-[0090]; [0092]-[0093]; [0098]; [0099]; [0107] and Figure 2), as opposed to handwritten technology like ’428 patent, and, as such, *Baker* is directed to a different problem and provides solutions those problems, and does not disclose the ’428 patent.

72. In more detail, *Baker* argues that their invention “avoids many of the problems of prior art systems by, in one aspect providing systems and methods that perform semantic and linguistic analysis on the fly, as the data is needed, rather than relying on pre-computed statistical counts stored in predetermined bins.” *Id.* [0024]. In order to create this “on the fly” analysis, *Baker* states that “[t]he reference database 40 can be an on-line database that is accessed by the word list generator 16.” *Id.* [0095]. In this patent the reference database is described as being “a dictionary, thesaurus, antonym dictionary, medical dictionary, legal dictionary, scientific and engineering dictionary, or other type of reference source.” *Id.* [0095]. Moreover, in Fig. 3 the

embodiment connects to an optional reference database 40 and “optionally connects via a transmission path to the reference database 40.” *Id.* [0101].

VI. Petitioners Grounds for Requesting *Inter Partes* Review are Deficient

A. Ground 1: Petitioners Fail To Show That Claims 1-3, 7-9, and 12-13 Would Have Been Obvious Based on *Arai* In View Of *Fenwick*

73. In my opinion, Petitioners fail to show any of the ’428 patent claims are rendered obvious by the cited prior art. As explained in detail below, *Arai* and *Fenwick* are directed to entirely different character and pattern recognition methods than the ’428 patent, and thus the prior art fails to teach multiple limitations of the claims. Moreover, a POSITA would not have been motivated to combine the cited prior art in the manner Petitioners propose.

1. *Arai* And *Fenwick* Are Directed To Fundamentally Different Approaches To Character And Pattern Recognition Than The ’428 Patent

a. The ’428 Patent

74. As discussed above, *supra* Section III, the ’428 patent seeks to solve problems with prior art character recognition methods which did not provide contextual analysis for character and word recognition, and which further required the user to press a button or take some other action to determine when one character or word ends, which “decreases the speed of character/letter input by the users.” ’428 patent at 1:31-42. As the ’428 patent explains:

The recognition engine or process provides an indication or output as to when one or more characters can be committed and sent to the application, as opposed to requiring a user to input of a space gesture, select a commit button, or other user input. In other words, the system provides an automatic and quicker recognition system.

Id. 2:54-59. This analysis of when one or more characters can be committed to an application without further user input is based on innovative context analysis, using “elaborate linguistic knowledge” to assist in performing “identification of e.g., the end of one word and the beginning of the next without the need to detect a space gesture between two words.” *Id.* 2:59-62. Specifically, “the use of linguistic tools with an intermediary buffer allows the system to use a ‘history’ of characters input by a user to increase the speed and accuracy of handwritten characters input to the system” which allows the system to “automatically decide when to clear this intermediate buffer by analyzing the context of two or more characters or words.”

Id. 2:59-62.⁴

⁴ The '428 patent explains that “Context information is any information that may help inform the recognition engine of the users intent, and may include information Such as sentence or word-pair context, language context, application-specific context (e.g. data fields/tags), user-specific context, location, time of day etc.” *Id.* 12:34-39.

This novel process allows the '428 patent's recognition process to solve the problems of prior art systems which were unable to perform advanced contextual analysis throughout the character recognition process, and which lacked the ability to analyze *portions* of entered character sequences to determine whether to commit the character sequence to an application and flush the utilized memory buffer *without* requiring additional user input. Moreover, the '428 patent's novel process involving multiple stages of contextual analysis allows for the application of character recognition across *multiple* languages, a significant challenge due to the vastly different recognition algorithms between different language models. *See e.g.*, Ex.2009 (Andersson Decl.) ¶¶3-15 (discussing challenges with attempting to combine handwriting recognition and text prediction algorithms from different languages); Ex.2010 (Ward Dep. Tr.), 28:1-10, 28:24-29:1, 29:11-19. 32:1-12, 33:1-7, 34:2-6, 34:8-24, 35:3-21, 35:24-36:2, 36:10-13, 50:1-51:14, 51:24-52:3, 52:20-53:10, 53:17-18, and 57:9-15.

75. The differences between prior art recognition systems and the '428 patent's advanced recognition system is further shown in Figure 2 of the '428 patent (shown below) and the corresponding disclosure in the specification walking through the '428 patent's recognition process.

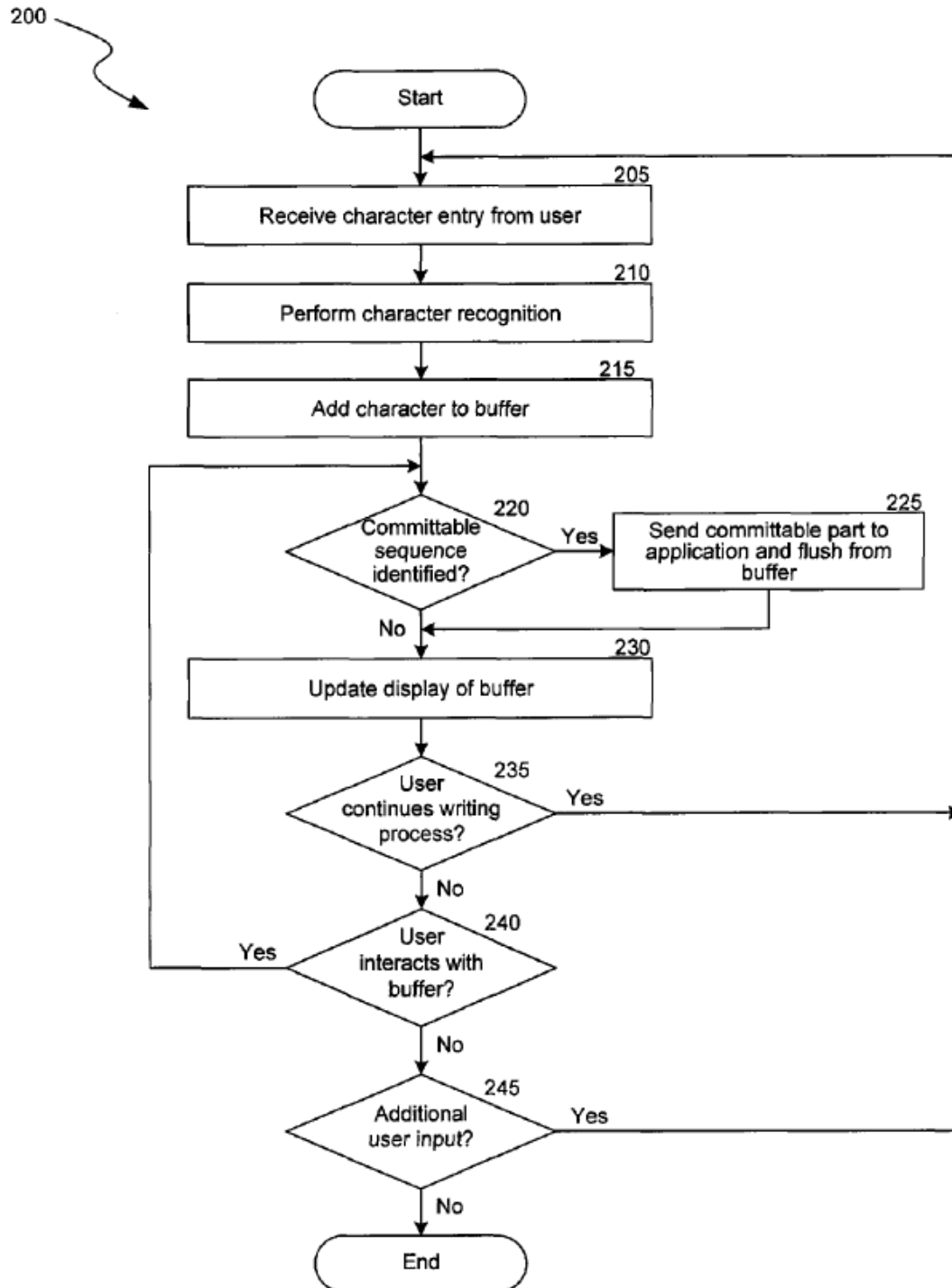


FIG. 2

76. For example, as shown above in Figure 2 (above), the character recognition process: 1) receives a character entry from the user (step 205); 2) performs character

recognition (step 210) ; 3) adds the character to a temporary memory buffer (step 215); 4) performs the process of identifying whether a committable character sequence has been identified from any portion of the identified character sequence (step 220); 5) if a committable character sequence has been identified, the process automatically sends that part to the application and automatically flushes that portion of the character sequence from memory (step 225). If a committable sequence is not identified, the process then updates the display buffer (step 230), and then monitors for additional user input (steps 235, 240, 245).

77. With respect to step 215, the specification explains that the temporary memory buffer may be displayed to the user to allow for editing or confirmation of the identified characters. '428 patent at 3:11-14. With respect to step 220, the specification explains that the identification of a committable sequence actually involves *two* steps: 1) identification of a “recognition candidate list,” which is a list of potential candidate character sequences the process determines that the user may be attempting to enter based on contextual and linguistic data (allowing the '428 patent process to predict what the user may be trying to enter) (*see id.* 3:13-21, 5:13:30); and 2) identification of whether any *sub-portion* of an identified character candidate list may be automatically committable to the user application and flushed from the temporary memory buffer (similarly using contextual analysis to render this decision) (*id.* 4:47-58, 9:42-49, Fig. 7).

78. The '428 patent's recognition process differs from other prior recognition methods in that it can perform contextual analysis at *multiple* different stages. For example, the specification explains that contextual analysis may be applied: 1) during the initial character recognition process (*id.* 12:31-34); 2) during the initial selection of the predicted character candidate sequences (*id.* 6:63-65, 13:26-34); 3) during the subsequent identification of sub-portions of the character candidate sequences which may be automatically committable to the application and flushed from the memory buffer (*id.* 4:50-58, 9:42-49); 4) when deciding whether to clear the temporary memory buffer (*id.* 3:2-4); and 5) even after words are committed to the application (*id.* 3:36-37).

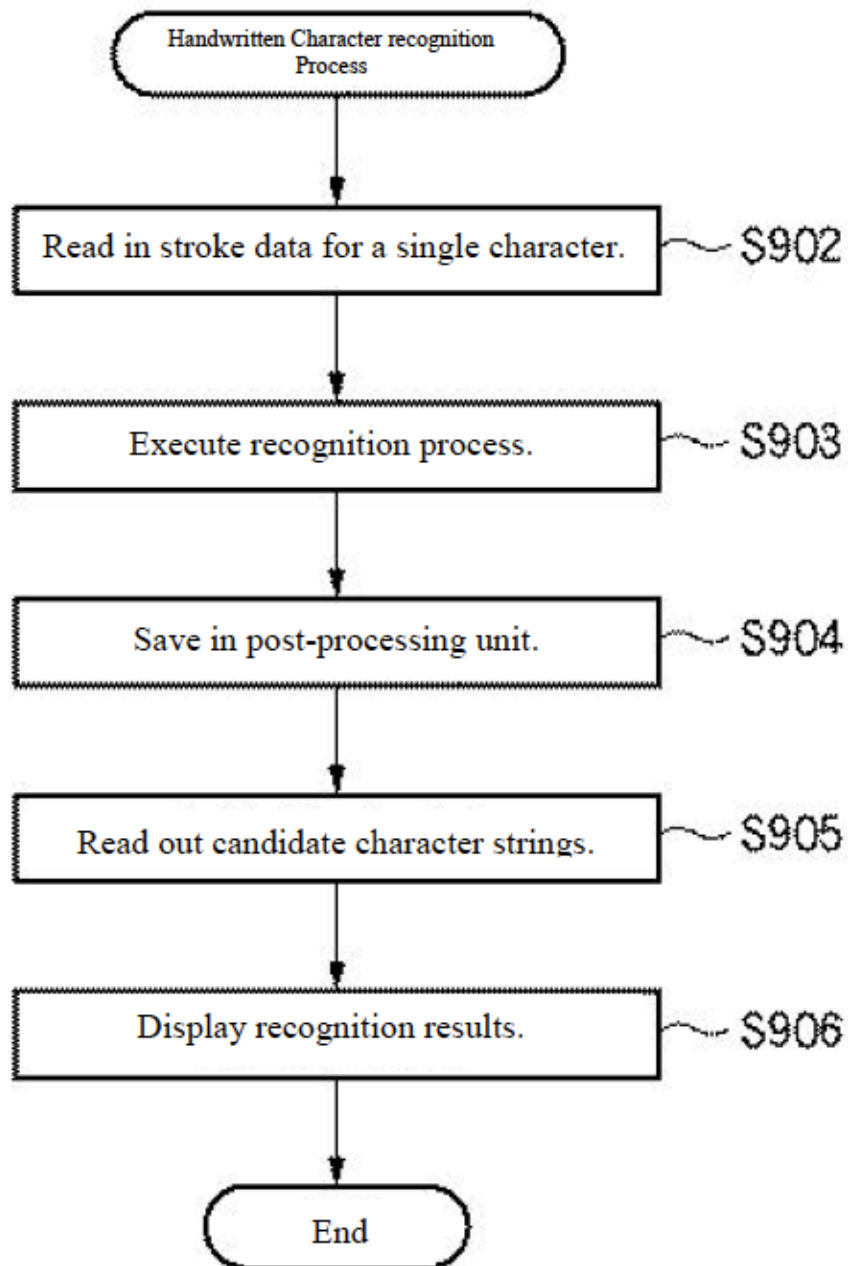
79. In short, the '428 patent process 1) performs individual character recognition on each entered character; 2) stores each character in a temporary and editable buffer displayed to the user that can be used for advanced contextual analysis; 3) identifies a list of probable character sequences based on contextual analysis and elaborate linguistic data to assist in predicting what the user may be trying to input; and 4) further identifies a *sub-portion* of any identified probable character sequences that may be able to be auto-committed to the application and removed from the temporary memory buffer (again based on contextual analysis and linguistic data).

b. Arai

80. Conversely, as shown in Fig. 8 of *Arai* (below), along with the corresponding disclosure in *Arai*'s specification (*see* [139]-[154]), *Arai*'s recognition process is directed to the exact type of prior art systems and methods the '428 patent was trying to improve upon. *Arai*'s process provides none of the contextual analysis described above, e.g. no ability to use neighboring words, no ability for predictive text generation, and no ability to automatically commit characters to an application and flush corresponding memory *without* additional user input. Instead, as shown below in Figure 8, the *Arai* recognition process simply performs initial character recognition (steps 902-903), performs basic post-processing using a common dictionary lookup to see whether an entered string of characters matches a known word (step 904), and then displays the results to the user for editing (without performing any contextual analysis described above or neighboring words or analyzing any *sub-portions* of any character sequences which may be automatically

committed to the application and removed from the memory buffer *without requiring further user input*) (steps 905-906).

[FIG. 8]



81. Moreover, as discussed above, while the '428 patent recognition process does not require user input to commit a character sequence to an application and flush the

corresponding memory buffer, the post-processing continuation determination unit in *Arai* does rely on user input. *Arai* [0085]-[0091], FIG. 4. Indeed, as the Petition admits, “*Arai* requires hitting the Enter Key to send a word (or words) to the application (*Arai* ¶[0091]).” Petition at 31.

82. This significantly differs from the ’428 patent, which requires no user input. The “recognition engine or process provides an indication or output as to when one or more characters can be committed and sent to the application, *as opposed to requiring a user to input* of a space gesture, select a commit button, or other user input.” See ’428 patent at 2:54-58 (emphasis added). As discussed above regarding the unique features of the ’428 patent, the user can continually input characters, without spaces, and the system will “automatically parse words from the string of input characters.” *Id.* 4:66-67.

83. Furthermore, the ’428 patent is predictive, as the recognition engine is able to “present the user with the one or more candidate character sequences” *Id.* 6:61-62; *see also id.* 3:12-14 (“The characters in the buffer may be displayed in the UI through a recognition candidate list, (a list that contains information of a hypothesis from the engine based on the content of this buffer).”). As seen in FIG. 3D the recognition engine shows the user some possible candidates: “Rat/Rar/Rather/Rate/Art/Rat.” *Id.* Fig 3D. Conversely, *Arai* does not offer predictive or hypothetical candidate characters, rather *Arai*’s recognition process simply performs its initial character

recognition for each entered Japanese symbol, and then checks to see if the series of symbols entered match a complete word in a dictionary lookup by calculating the chain probabilities of the recognized character sequence. *See, e.g.,* Arai [139]-[154].

84. Moreover, while the '428 patent's use of advanced contextual analysis allows for character recognition across multiple languages (*see supra* at Section III), *Arai's* lack of advanced contextual analysis (detailed above) significantly limits its ability to be effectively used with more than one language. *See, e.g.,* Ex.2009 (Andersson Decl.) ¶¶3-15 (discussing challenges with attempting to combine handwriting recognition and text prediction algorithms from different languages); Ex.2010 (Ward Dep. Tr.), 28:1-10, 28:24-29:1, 29:11-19, 32:1-12, 33:1-7, 34:2-6, 34:8-24, 35:3-21, 35:24-36:2, 36:10-13, 50:1-51:14, 51:24-52:3, 52:20-53:10, 53:17-18, and 57:9-15.

c. Fenwick

85. In contrast to the '428 patent, which is a “linguistically aided autocommit system for handwriting recognition user interfaces” (*see* '428 patent at 2:39-40), *Fenwick* merely discloses a spacing correction algorithm. *Fenwick* at 1:6-10; 1:34; and 1:42-45.

86. *Fenwick* describes a system and process of multiple memory buffers that store characters in one or more of the buffers, as discussed above, to create a word match, in order to append a space after it. *Fenwick* is only able to append a space character after a full word has been detected and then the buffer is subsequently cleared and

deleted. Beyond this, *Fenwick*, does not do any contextual analysis (as discussed above with regard to the '428 patent) of the characters.

87. This contrasts with the '428 patent, which utilizes a single temporary memory buffer to store the character sequences, as seen in Figure 2 of '428 patent, where the character is added to a temporary editable character buffer and then the process can utilize characters stored in that buffer for contextual analysis. “The recognition engine may additionally access or receive and process, dynamic context information from the operating system and/or application program (s) in order to facilitate its identification [of the characters].” *See* '428 patent at 12:31-34. This may include “information such as sentence or word-pair context, language-context, application-specific, location, time of day...As another example, [or a] probabilistic “N-best” list from another mode of input. An N-best list is a rank ordered list of suggested words.” *Id.* 2:36-50. Moreover, the '428 patent is able to determine that the first recognized sequence portion are the intended characters input by the user and therefore, commits and sends that portion to the application, without user input. '428 patent at 3:44-46.

88. The '428 patent further differs from *Fenwick*, because the '428 patent improves on speed and accuracy for the user. *See* '428 patent at 3:1; 9:2-6; 10:11-5. “Moreover, the present system provides improved speed since a user need not purposefully pause between each inputted character, or select a commit button;

instead, the user can continue to enter characters, one on top of the other, in the text entry location.” *Id.* 10:11-5. Moreover, this system and method of the ’428 patent increases speed of input and “acceptance of input using character input in alphabets that typically do not use a space between characters (e.g., Japanese, Chinese, and /or Korean alphabets). *Id.* 9:2-6.

2. Because Arai And Fenwick Are Directed To Fundamentally Different Approaches To Character And Pattern Recognition Than The ’428 Patent, They Fail To Disclose Or Render Obvious Multiple Claim Limitations

a. Claim 1

89. The steps of Claim 1 of the ’428 patent are reproduced below:

1[pre]: A method for data input on a touchscreen of a mobile device, the method comprising:

1[a]: receiving a first user input, wherein the first user input is handwritten input received on the touchscreen of the mobile device;

1[b]: determining a first recognized sequence based on the first user input; wherein the first recognized sequence is a best match of the first user input;

1[c]: writing the first recognized sequence to a memory storage buffer of the mobile device;

1[d]: generating a first candidate sequence, wherein the first candidate sequence is a best match of one or more recognized sequences;

1[e]: presenting the first candidate sequence to the user, wherein the presenting includes displaying the first candidate sequence to the user and allowing the user to edit the first candidate sequence;

1[f]: identifying at least a first sequence portion of the first candidate sequence wherein the identifying is at least partially based on one or more rules or dictionaries of words accessible to the mobile device;

1[g]: removing the first sequence portion from the memory storage buffer, wherein the removing is performed in the absence of receiving user input editing the first candidate sequence, representing a space character, or selecting a word; and

1[h]: sending the first sequence portion to an application running on the mobile device.

'428 patent, cl. 1.

90. As discussed further below, Petitioners rely on the wrong disclosure in *Arai* for limitations 1[d] and 1[e]. Moreover, *Arai* does not disclose or suggest limitation 1[f], and *Arai* in view of *Fenwick* do not disclose or suggest limitations 1[g] and 1[h].

i. Petitioners Identify The Wrong Disclosure In *Arai* For Limitations 1[d] And 1[e]

91. First, Petitioners identify the wrong disclosure in *Arai* for limitations 1[d] and 1[e].

92. Specifically, Petitioners and their expert Mr. Ward mischaracterize the

disclosure in *Arai* and identify the **wrong** steps in *Arai*'s handwriting recognition process as purportedly disclosing limitations 1[d] and 1[e] of the '428 patent, i.e. "generating a first candidate sequence, wherein the first candidate sequence is a best match of one or more recognized sequences" and "presenting the first candidate sequence to the user, wherein the presenting includes displaying the first candidate sequence to the user and allowing the user to edit the first candidate sequence." Specifically, as discussed further below, Petitioners and their expert improperly point to the same disclosure in *Arai* regarding *Arai*'s initial character recognition step for **both** limitation 1[b] (i.e., "determining a first recognized sequence...") **and** limitation 1[d] (i.e., "generating a first candidate sequence...").

a. *Arai*'s Actual Recognition Process

93. As discussed above, the *Arai* recognition process simply reads in character input, performs initial character recognition, executes post-processing using calculations of chains of probabilities and a common dictionary lookup to see whether an entered string of characters matches a known word, and then displays the results to the user for editing.

94. For example, *Arai*'s recognition process is explained in paragraphs [139] through [155] of *Arai*, as well as Figure 8 of *Arai* (which shows a flowchart of *Arai*'s recognition process consisting of sequential steps S902, S903, S904, S905, and

S906). As *Arai*'s specification explains, at the first **step S902** of Figure 8, "stroke data for a single character is read into a work area." *Arai*, [0141].

95. Next, at **step S903**, character recognition is performed on the stroke data. *Arai*, [0142]. This process involves calculating "evaluation values" relating to "the values for the degree of matching" of the stroke data to "reference character dictionary data." *Id.* The character codes and the evaluation values corresponding to the reference character dictionary data that have the "highest degrees of matching" are then stored as the "recognition results" for each individual character entered by the user. *Id.* For example, *Arai* provides an example of two individual characters entered by the user "工場", with the recognition process first determining a set of evaluation values for the first character "工" and then subsequently determining a second set of evaluation values for the second character "場". *Arai* [144]-[145]. Importantly, at this point in the *Arai* recognition process, only *individual* characters have been analyzed, i.e., *no* analysis has been performed on any combination of characters to facilitate generating any candidate character sequences.

96. Subsequently, at **step S904**, the recognition result candidates for each individual character entered by the user are sent to the post-processing unit 34, which then performs post-processing on the *combined* character sequences in memory. Specifically, *Arai* explains that "in the post-processing unit 34 the chain probabilities for the character strings *combining these candidates* are calculated using a

dictionary for post-processing.” *Arai* [0146] (emphasis added). The post-processing unit 34 then “determines the candidate character string with the highest chain probabilities among these accommodations.” *Id.*

97. Finally, at step S905 and S906, the candidate character string is read out from the post-processing unit 34 and the results are displayed for the user to edit. *Id.* [0147]-[0149].

98. Thus, while *Arai* performs an initial character recognition (i.e., steps S902-S903), and then generates a probable candidate character sequence to display to the user for editing (steps 903-906), *Arai*’s process fails to perform the ’428 patent’s steps of *further* identifying a *portion* of any probable candidate character string that may be automatically committed to an application and flushed from the memory buffer (without requiring further user input).

b. Petitioners’ Mischaracterization Of Arai’s Process

99. Faced with this deficiency in *Arai*’s disclosure, Petitioners and their expert attempt to “double dip” from *Arai*’s disclosed method, pointing to the *same* disclosure in *Arai* for both limitation 1[b] (the initial character recognition limitation), as well as limitation 1[d] (the “generating a first candidate sequence...” limitation). Limitations 1[b] and 1[d] are separate steps, and being distinct is important to how Claim 1 operates.

100. For example, for limitation 1[b] (“determining a first recognized sequence...”), Petitioners point to *Arai*’s initial character recognition process, i.e., the determination of the highest ranked “recognition results candidates” via calculation of the “evaluation values” for each individual character entered. Petition at 17-19. Indeed, Petitioners cite to and walk through the *same* paragraphs Patent Owner discusses above regarding *Arai*’s recognition process, i.e. paragraphs [0142] through [0145] (discussing *Arai*’s calculation of evaluation values and walking through an example of how these recognition results are processed for two sample characters entered by the user, i.e., the characters “工” and “場”). *Id.*

101. Then for limitation 1[c] (“writing the first recognized sequence to a memory storage buffer ...”), Petitioners point to the same paragraphs discussed above regarding the individual character recognition results being stored in RAM. Petition at 20-21. Petitioners also again rely on *Arai*’s example walk through of the process, citing to *Arai*’s example of how each individual “character’s recognition results (e.g., each of 工 and 場) are written to a memory storage buffer” *Id.*, 20-21.

102. Thus, at this point, Petitioners have *already* relied on *Arai*’s initial character recognition process (i.e., the determination of the highest “evaluation values” for each individual character entered) for limitation 1[b] and further relied on the storage of these highest-ranked “evaluation values” in memory for limitation 1[c]. Following *Arai*’s example walkthrough of its recognition process, at this point both

individual characters “工” and “場” have *already* been evaluated and stored in memory together.

103. Petitioners then improperly attempt to rely on the same disclosure and process steps in *Arai* for limitation 1[d], which requires the *additional* processing step of “generating a first candidate sequence, wherein the first candidate sequence is a best match of one or more recognized sequences.” Petition at 21-22. Specifically, Petitioners assert that the recognized characters “工” and “場” form a first candidate sequence “工場”, and that “[b]ecause *Arai* generates 工場 using an ‘evaluation value’ and ‘first-ranked recognition result, a POSITA would have understood this first candidate sequence 工場 to be a best match of the first and second recognized sequences (工 and 場).” *Id.* But, as discussed above, Petitioners *already* relied on *Arai*’s determination of “evaluation values” and “first ranked recognition results” for each individual character for limitations 1[b], and the storage of these individual characters together for limitation 1[c].

104. Moreover, even if Petitioners had not already relied on these same steps in *Arai* for earlier limitations, Petitioners cannot rely on these steps to satisfy limitation 1[d] of the ’428 patent, because limitation 1[d] requires an *additional* processing step (beyond the initial individual character recognition process step), i.e., processing of the *combination* of characters stored in memory to generate the 428 patent’s hypothetical “candidate sequence.”

105. As discussed above, the very first place that *Arai* performs post-processing on the ***combined*** character string in memory (i.e., generating a “best match of one or more recognized sequences” as required by limitation 1[d]), is not until step S904 ***after*** the initial character recognition process is performed in *Arai* (step S903), i.e., when those individual character results are sent to the post-processing unit to calculate the “chain probabilities for the character strings ***combining these candidates.***” *Arai*, ¶[0146] (discussing step S904).

106. Put another way, the Petition confuses a *candidate character* with a *candidate sequence*. For instance, Petitioners state “*Arai* receives the two inputs 工 and 場 (§VIII.A.1.b), **determines the recognized sequence for each**, namely, 工 and 場 (§VIII.A.1.c), and **from these two sequences**, *Arai* generates a first candidate sequence.” Petition at 21-22 (emphasis added, highlighting Petitioner mischaracterizing characters as sequences). But *Arai* specifically describes in Step S903 how the ‘recognition process’ is performed on the stroke data, which are the first “**candidate characters**” (see *Arai* [0142]) of the handwritten individual character. There is no recognized sequence for each character.

107. To treat these characters as sequences (as Petitioner does) is problematic and runs afoul of important distinctions between characters and sequences per *Arai*'s disclosures in those same paragraphs. As discussed above, *Arai* describes how these ‘character codes’ or candidate characters are assigned a value, which are the “values

for the degree of matching (the evaluation values)...[which correspond] to the reference characters dictionary data.” *Id.* [0142]. Those character codes with the highest degrees of matching are stored as ‘recognition results.’ Importantly, as discussed above, these evaluation values are **assigned to individual characters, not sequences.**

108. For example, “the first candidate character in the recognition result for "工", by the handwritten character recognition unit 35, is " 工," and the evaluation value is 100, where the second candidate character is " 亅", with an evaluation value of 100, and the third candidate character is " 丂", with an evaluation value of 90.” *Id.* [0144].

109. In S903, the user has input a second character and the “first candidate character for the recognition result for "場" is "揚", with an evaluation value of 98, where the second candidate character is " 場", with the evaluation value for the degree of matching being 96.” *Id.* [0145].

110. Once again, Petitioners conflate characters with sequences, arguing that *Arai* “generates 工揚 using an “evaluation value” and “first-ranked recognition result.” Petition at 21. This blurring of characters-versus-sequences is misleading and is not how *Arai* generates 工揚, the first candidate sequence. As discussed above, the evaluation value is ascribed to candidate characters, not sequences. Furthermore, in

S904, “in the post-processing unit 34 the chain of probabilities for the character strings **combining** these candidates are calculated using a dictionary for post-processing...Given this, the post-processing unit 34 determines the candidate character string with the **highest chain probability** among these accommodations.” *Id.* [0146] emphasis added.

111. Therefore, *Arai*’s candidate sequence is not comprised of the highest evaluation value (which *Arai* ascribes only to characters) but rather it is the “chain probabilities for the character strings combining these candidates...using a dictionary for post processing.” *Id.* [0146].

112. Furthermore, *Arai* makes it clear that the per-character evaluation values may actually be overruled by the sequence chain probability calculation: “when "工場", is inputted as the handwritten characters, then **even if** the recognition results by the handwritten character recognition unit 35 are "工揚", **the process in the post-processing unit 34** in Step S904 will determine that the candidate character string with the highest chain probability is "工場," so "工場" is read out as the candidate character string for the handwritten character.” *Id.* [0148] (emphasis added). N.B. the subtle difference between the kanji characters 場 (as manually written and finally determined in 34) and 揚 (result of intermediate character recognition 35).

113. Thus, the first candidate sequence for *Arai* would be calculation of “the chain

probabilities for the character strings **combining** these candidates...using a dictionary for post-processing.” *Id.* [0146], and not relying on the individual evaluation values, as Petitioners assert.

114. Indeed, Petitioners’ expert Mr. Ward confirmed during his deposition that “generating a first candidate sequence” in the ‘428 patent is a *separate* distinct step from the earlier character recognition step of claim 1, and that (contrary to Petitioners’ arguments), the candidate sequence is not just the output of the initial individual character recognition step, it is instead an additional step of determining a hypothesis of what character sequence the user is trying to enter based on the combination of characters in the memory buffer. *See, e.g.,* Ex.2010 (Ward Dep. Tr.), 106:4-10 (“**Q.** Since these are different steps in the method, the determining of first recognized sequence is a separate step from the subsequent generating a first candidate sequence; correct? **A.** *It’s listed here as a district step, yes.*”); *id.*, 109:22-110:8 (“**Q.** So to my point then, the generation of the first candidate sequence is doing something different than the initial best match determination that’s happening at the earlier step with determining a first recognized sequence ... **A.** *They are discrete steps, yes, sir.*”); *id.*, 126:10-20 (“**Q.** These candidate sequences are not just the combination of the individual characters that have been entered; right? They are instead hypotheses of what the user may be trying to enter; right? **A.** *I believe you’re paraphrasing but, yes....*”); *id.*, 12522-126:3 (“**Q.** So when the recognition engine

is creating these candidate sequences, it's trying to hypothesize what character sequences the user is trying to enter based on analyzing the current content of the characters in their memory buffer; correct? *A. In part, yes.*”).

115. Further confirmation that Petitioners and their expert are mischaracterizing *Arai*'s recognition process can be found in their arguments regarding limitation 1[e].

For example, limitation 1[e] of the '428 patent requires “presenting the first candidate sequence to the user, wherein the presenting includes displaying the first candidate sequence to the user and allowing the user to edit the first candidate sequence.” For this limitation, Petitioners rely on disclosure from *Arai* that the example characters “工場” are displayed “as the recognition result in the input character buffer area 61.” Petition at 22 (citing *Arai* [151]). Petitioners then cite to disclosures in *other* parts of *Arai* regarding the ability to edit characters in the input area. *Id.* (citing *Arai* [0051], [0088], [0089]).

116. However, when explaining how its method works “as a time series” (i.e., in linear order), *Arai* makes it clear that editing of the characters in the input area does not occur until *after* the post-processing unit performs its analysis of the full character strings in memory to calculate the chain probabilities. *See Arai* [150]-[154]. Put simply, the example characters “工場” Petitioners rely on as being displayed are *not* editable by the user as required by limitation 1[e], because (once again) they are *not* the candidate character sequence determined by the post-

processing unit in *Arai* (which has not occurred yet), they are merely the results of the initial **individual** character recognition step. *See, e.g.*, Ex.2010 (Ward Dep. Tr.), 210:18-21 (testifying that *Arai* allows editing after the post-processing step).

117. In short, *Arai* discloses or suggests limitation 1[d] requiring “generating a first candidate sequence, wherein the first candidate sequence is a best match of one or more recognized sequences,” as this occurs in *Arai* at step S904, when the post processing unit calculates “the chain probabilities for the character strings” and then “determines the candidate character string with the highest chain probabilities....” *Arai*, ¶[0146]. *Arai* further discloses or suggests limitation 1[e], requiring “presenting the first candidate sequence to the user, wherein the presenting includes displaying the first candidate sequence to the user and allowing the user to edit the first candidate sequence,” as this occurs in *Arai* at steps S905-S906, when the candidate sequence is displayed to the user for editing. But once again, Petitioners are identifying the **wrong** disclosure in *Arai* for these limitations because Petitioners’ mischaracterization allows them to stretch *Arai*’s process out into more steps than actually occur.

118. For example, by incorrectly relying on *Arai*’s initial character recognition process (i.e., step S903) for limitation 1[d] (requiring “generating a first candidate sequence...”), revisiting step S903 already used for 1[b], Petitioners are then able to misrepresent *Arai*’s sole post-processing step S904 (i.e., calculating “the chain

probabilities for the character strings”) as being the *subsequent* processing steps required by the ’428 patent, including limitations 1[f] through 1[h] requiring “identifying at least a portion of the candidate sequence” that may be automatically committed to an application and flushed from the memory buffer.

119. But as explained above, and addressed further below for each corresponding limitation, *Arai* does *not* disclose or suggest any subsequent processing steps after the generation of the candidate character sequence, and thus does not disclose or suggest limitations 1[f] through 1[h], requiring *further* processing to identify at least a portion of said candidate character sequence (or automatically committing said portion to an application and flushing said portion from the memory buffer).

- ii. **1[f]: “identifying at least a first sequence portion of the first candidate sequence wherein the identifying is at least partially based on one or more rules or dictionaries of words accessible to the mobile device;”**

120. *Arai* does not disclose or suggest this limitation.

121. As discussed above the Petitioner is trying to claim that the candidate sequence is based on the ‘evaluation values,’ so that the Petitioners can claim that *Arai* discloses the first sequence portion of the first candidate sequence. However, as discussed above, the **first candidate sequence** of the *Arai* does not depend on the individual character’s evaluation values. It depends on the ‘post-processing unit 34’ within which “the chain probabilities for the character strings **combining** these candidates are calculated using a dictionary for post-processing.” Petition at 24

(citing Ward Decl., ¶126) (emphasis added). Therefore, *Arai* does not identify first portion sequences.

122. Moreover, as the dictionary that the Petitioner refers to is located in the post processing unit 34, which as discussed above is a distinct stage **after** the individual characters have been assigned an evaluation value, it is not possible for the post-processing dictionary to be involved in assigning evaluation values.

123. Additionally, the Petitioner, again, incorrectly refers to the individual characters as ‘sequences’: “where 工 is a portion of the first candidate sequence 工揚.” Petition at 26. Due to the nature of *Arai*, one cannot identify a first sequence portion, because the post processing unit requires that candidate characters are combined in order to calculate the chain probabilities for the character strings: “the chain probabilities for the character strings **combining these candidates** are calculated using a dictionary for post-processing.” *See Arai* at [0146]. Taking the two characters above, the post-processing unit would calculate the combinations of the characters (for example, "工揚", "工場", "エ揚", "エ場", "エ揚", "エ場") and determine the “candidate characters string with the highest chain probability among these accommodations.” *Id.* [0146].

124. The candidate character string that has the highest chain probability in *Arai* is “determined through considering the probability that the characters in the candidate

character string...are linked together, and considering the degree of match with reference character dictionary data within the recognition dictionary.” *Id.* [0053].

125. In short, *Arai* is not able to identify an isolated first sequence portion of the first candidate sequence during this step because all characters in the sequence are needed for the chain probability calculation.

126. In the example provided in the '428 patent, the user writes “Iliketowrite” and the engine is able to recognize that “I like” is a first sequence portion of the first candidate sequence “I like to write.” Thus, the next portion of the sequence would be “to write.” *See* '428 patent at 3:47-60. This is also seen in the '428 patent Fig. 4F where “AB” is the first sequence portion of the first candidate sequence in the “AB CDEFG” example. *Id.* FIG. 4F.

127. While Petitioners rely on *Arai*’s post-processing step S904 (i.e., calculation of the chain probabilities for the character strings) as satisfying this limitation (Petition at 23-26), as discussed above, step S904 of *Arai*’s recognition process is where *Arai* generates the candidate character sequence. At **no** point during this post-processing step, or any time after this step, does *Arai* perform any **additional** processing to identify any “first sequence portion” of said candidate character sequence, as required by limitation 1[f].

128. Specifically, because there is no disclosure in *Arai* of identifying any “portions” of the candidate character sequence, Petitioners simply assert that step

S904's identification of the character string with the highest chain probability (i.e., “工場” in the example shown in *Arai*) **necessarily** includes an identification of a sub-portion of the candidate character string. Petition at 26. As Petitioners argue, because the example character string includes both the individual characters “工” and “場”, then “工” is a portion of the first candidate sequence 工場 (‘identifying at least a first sequence portion of the first candidate sequence’).” *Id.*

129. Petitioners’ argument is plainly wrong. If the mere generation of the candidate sequence of limitation 1[d] **automatically** satisfied the separate processing step of limitation 1[f] (requiring identification of at least a portion of the candidate character sequence), then limitation 1[f] would be superfluous to claim 1.

130. Moreover, even if *Arai*’s recognition process did perform some identification of a “portion” of the candidate character sequence—and it does not—Petitioners identify **no** disclosure in *Arai* of said portion being identified “at least partially based on one or more rules or dictionaries of words accessible to the mobile device” as required by limitation 1[f]. Specifically, Petitioners point to the fact that the chain probabilities are calculated by referencing dictionary data. Petition at 24-25. But the result of *Arai*’s dictionary comparison is merely the **full** candidate character string (i.e., “工場” in the example shown in *Arai*) (see *Arai* ¶¶[146]-[152]), and the dictionary lookup is **not** being used to identify of any “portion” of the candidate character sequence as required by limitation 1[f].

131. In short, contrary to Petitioners' assertions, *Arai* fails to disclose or suggest identifying any "first sequence portion of the first candidate sequence" and further fails to disclose or suggest said "first sequence portion" being identified "at least partially based on one or more rules or dictionaries..."

iii. 1[g]: "removing the first sequence portion from the memory storage buffer, wherein the removing is performed in the absence of receiving user input editing the first candidate sequence, representing a space character, or selecting a word; and"

132. *Arai* in view of *Fenwick* fail to disclose or suggest this limitation.

133. As discussed above, *Arai* does not disclose or suggest identification of any "first sequence portion of the first candidate sequence," as required by limitation 1[f]. Thus, *Arai* necessarily cannot disclose or suggest the subsequent limitations requiring action on that first sequence portion (i.e., limitations 1[g] and 1[h]). Additionally, because Petitioners do not contend that *Fenwick* discloses or suggests identification or use of any first sequence portion of the first candidate sequence, *Arai* and *Fenwick* together also fail to disclose or suggest this limitation.

134. *Arai* in view of *Fenwick* further fail to disclose or suggest the remainder of this limitation, i.e., removing the first sequence portion from the memory buffer, "wherein the removing is performed in the absence of receiving user input editing the first candidate sequence, representing a space character, or selecting a word." As Petitioners admit, "*Arai* is silent on when and under what conditions the results are remove from RAM." Petition at 26. And while Petitioners point to disclosure

in *Fenwick* regarding automatically deleting identified words from memory buffers, as discussed below, a POSITA would not have been motivated to combine *Arai* and *Fenwick*, and would further not have a reasonable expectation of success in combining these references, for the reasons discussed below.

135. **First**, a POSITA would not have been motivated to combine *Arai* and *Fenwick*, because, as discussed above, *Arai* is a Japanese patent application disclosing a handwriting recognition algorithm for the Japanese language (a language which does not utilize spaces between words), while *Fenwick* is an English patent disclosing an algorithm for correcting spacing errors which commonly occur when reading in English characters. Simply put, a POSITA would not have been motivated to combine an algorithm for fixing spacing errors between words with an algorithm designed to recognize characters and words in a language which does not utilize spaces between words.

136. **Second**, a POSITA would not have been motivated to combine *Arai*'s Japanese handwriting recognition algorithm with *Fenwick*'s English spacing correcting algorithm, because both the initial recognition process, as well as the post-processing that occurs on any recognized characters, differs substantially between different languages, and even differs substantially between different dialects between the same language. *See, e.g.*, Ex.2009 (Andersson Decl.), ¶¶3-15; Ex.2010 (Ward Dep Tr.), 28:1-10, 28:24-29:1, 29:11-19. 32:1-12, 33:1-7, 34:2-6, 34:8-24,

35:3-21, 35:24-36:2, 36:10-13, 50:1-51:14, 51:24-52:3, 52:20-53:10, 53:17-18, and 57:9-15.

137. For example, the differences between different language recognition models and challenges with combining different models was frequently discussed in academic publications prior to the priority date of the '428 patent. For example, as explained in *Handwriting: Recognition, Development and Analysis*:

There are several factors that make character recognition a more complex task than digit recognition. In this scenario, one takes into account the existence of a greater number of classes to be recognized (this set can be composed of digits, uppercase, lowercase or accented characters, punctuation and symbols). The variation of the calligraphy of the individuals during the act of writing over time, are characteristics that also make the process of recognition of handwritten characters a complex activity. It is, therefore, noticeable that there is a high level of variability of the instances that can be correctly assigned to the same class. On the other hand, the high similarity between some distinct characters also increases the occurrence of false positives. The task still becomes more challenging, as it is usually specific to each domain and application. ***In this way, the techniques established and used for the recognition of a certain family of characters, such as Latin/Roman, can not be applied in the same way for characters of different origin, such as Indus, Chinese or Arab.***

Ex.2011, Lacerda, E.B., de Souza, T. V., Zanchettin. C., Rabelo, J. C., & Coutinho, L.D. (2017). *Handwriting Recognition: Overview, Challenges and Future Trends*. In Bezerra, B.L., Zanchettin, C., Toselli, A.H., & Pirlo, G. (Eds.), *Handwriting: Recognition, Development and Analysis* (pp.3-32). Nova Science Publisher.

138. Similarly, as explained in *Multi-Language Online Handwriting Recognition*:

There is strong variability in writing style between different groups of people and between individuals, for example preference for printed style versus cursive writing or the age and location where writing was taught for the same language. There is strong variability even for the same person, depending on the input speed, sloppiness, or context in which the text is input, e.g., while walking and on a mobile phone. Many ambiguities are inherent in the input, e.g., a small stroke could indicate an apostrophe or could just be accidental 'noise'. Also, many characters are virtually indistinguishable without context. e.g., a single round stroke could be any of the Unicode characters shown in Fig. 2. Similar ambiguities exist for other basic shapes such as horizontal or vertical strokes. ***For scripts with thousands of characters, machine learning is still a challenge, as is the gathering of proper training and evaluation data and the coverage of all common input symbols.*** Handwriting input often has a non-monotonic relationship to the output which is caused by delayed strokes (e.g., t-strokes and i-dots in English) or corrections that are entered later (e.g., a forgotten letter is inserted into a word.)

Ex.2012, Keyzers, D., Deselaers, T., Rowley, H., Wang, L., & Car-bune, V. (2017, June). *Multi-Language Online Handwriting Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1180- 1194.

139. The complexity of handwriting recognition directly impacts the difficulty in designing software for handwriting recognition. Mr. Jonas Andersson, one of the named inventors of the '428 patent, testified to this extent. He is a software engineer who has worked in the handwriting recognition field for more than two decades, handwriting recognition systems. He describes how the handwriting recognition systems “are typically divided into two separate parts,” the lower-level “single character recognition” (“SCR”) part, which “is the process for solving the initial character recognition problem,” and then the higher-level “multiple character

recognition” (“MCR”) part, which “attempts to recognize a sequence of strokes that represents one or more characters.” Ex.2009 (Andersson Decl.), ¶¶4-5.

140. Furthermore, Mr. Andersson explains, “both the lower-level SCR, as well as the higher-level MCR processes differ substantially depending on the target language being analyzed.” *Id.*, ¶7. Moreover, Mr. Andersson details how Alphabetic languages differ from Chinese, Japanese and Korean languages: “[w]hile Cerence and its predecessors have handwriting recognition algorithms for the Alphabetic languages (i.e., for recognition of letters in scripts like Latin, to be used for example for the English language) as well as handwriting recognition algorithms for CJK languages (Chinese, Japanese, Korean),” “Cerence utilizes different engineering teams for each, with each developing different handwriting recognition technologies.” *Id.*

141. Mr. Andersson provides an example:

For example, the SCR process, where an input stroke sequence is matched against character templates in a database, requires much more strokes to be matched for CJK languages, because there are a lot more characters in the CJK languages than in the English language and each CJK character contains on average a lot more strokes than an English letter. This means that each SCR operation takes a lot more time for the CJK languages than for English.

Additionally, because there are so many strokes in each CJK character, there are a lot more possible segmentations in each CJK writing than in English writing. So, in total, an MCR technology that is based on the robust segmentation by recognition strategy would have to do many more and much heavier SCR operations to solve the segmentation

problem, i.e. the MCR operation would be much slower for each CJK language than for English language.

Id., ¶¶8-9.

142. At the lower-level SCR stage, Mr. Andersson further explains that “there can be variances in handwriting style, multiple possible recognition results for a particular input shape, and misidentifications of shapes due to certain characters having similar stroke patterns.” *Id.*, ¶11. Mr. Andersson confirms that “[t]hese challenges are exacerbated if attempting to combine multiple language models, which necessarily increases the potential recognition results, as well as the potential for misidentification of characters.” *Id.*

143. Mr. Andersson explained how the added complexities of a higher-level MCR process for CJK languages meant that “the engineering team working on the CJK handwriting recognition technologies did not fully develop a higher-level MCR process for CJK languages until more than a decade after a higher-level MCR process was developed for Alphabetic type languages, like English.” *Id.*, ¶10.

144. Furthermore, at the higher-level MCR stage, Mr. Andersson further highlighted “one problem to be solved in the segmentation problem, i.e., identifying when one character ends and another begins.” *Id.*, ¶12. Moreover, “the content, order, and shape of characters entered by the user may vary significantly even for a single language, depending on the user’s education, nationality, handwriting style,

and the context of the input being received.” *Id.* Thus, “MCR models can vary significantly depending on the application, even for a single language.” *Id.*

145. Moreover, “as with the SCR stage, these challenges in the MCR stage are exacerbated if attempting to combine multiple language models.” *Id.*, ¶13. For example, “different language models would utilize different dictionary or database lookups, different contextual analysis, different formatting assumptions, and different processes for determining whether one recognized character is associated with another.” *Id.*

146. In short, “handwriting recognition models differ substantially depending on the target language being recognized, including differences in both the lower-level SCR stage (i.e., the initial character recognition stage) as well as the higher-level MCR stage (i.e., the segmentation post-processing stage).” *Id.*, ¶15. These differences “make[] combining recognition models for different languages challenging.” *Id.* Moreover, “[d]ue to these differences, Cerence and its predecessors needed to use and in fact used different engineering teams in different countries across the world to develop different handwriting recognition.” *Id.*

147. Mr. Andersson also confirmed that, based on his experience, “these challenges existed as of October 15, 2012,” the priority date of the ’428 patent. *Id.*, ¶14.

148. Indeed, Petitioners *own* expert, Mr. Ward, testified *exhaustively* during his deposition regarding the challenges faced in performing individual character

recognition and post-processing, as well as the additional challenges that may be present for different languages. For example, Mr. Ward testified that “recognizing handwritten input like recognizing speech input is a complex and difficult problem.” Ex.2010 (Ward Dep Tr.), 28:24-29:1; *id.*, 29:19 (“It’s a complex problem.”). Indeed, to solve this problem, Mr. Ward testified that “you need to use everything you can to try and come up with a successful solution,” including using “context information which might be incorporated in the recognition itself,” as well as “application-specific information,” and “post processing for things done after some initial character recognition.” *Id.*, 28:1-10.

149. Mr. Ward further testified that post-processing in recognition algorithms “can be many different forms,” such as “comparing against a list of acceptable inputs or patterns for acceptable inputs,” or “going back and revisiting the recognition results based on further information that was not available at the time of the input.” *Id.*, 29:11-18.

150. Mr. Ward further testified about the challenges that may occur when attempting to perform character recognition and post-processing on even a *single* language. For example, Mr. Ward testified that: 1) “there could be more than one possible recognition result for the particular input shape or particular user input that can vary depending on handwriting style or handwriting variability” (*id.*, 32:1-6); 2) recognized characters may be confused because they have nearly identical shapes,

but are intended to be different characters depending on the context, “such as the exact same symbol may represent a 1 or an I or a lower case L” (*id.*, 32:7-12); and 3) “Unusually spelled proper names or incomplete words make it hard to perform simple dictionary or database checks” (*id.*, 33:1-7). Mr. Ward further confirmed that these types of problems would exist as of the priority date of the ’428 patent, as these are problems that would “need to be dealt with in general.” *Id.*, 34:2-6.

151. Indeed, when asked what additional challenges a POSITA would face when attempting to do handwriting, character, or word recognition for multiple languages, Mr. Ward testified that algorithms may differ *even for the same language*, due to “different writing styles that depend on where someone went to school or when someone went to school,” as well as “dialect differences or national differences in what words are or how certain things are written such as dates.” *Id.*, 34:8-19; *id.*, 35:24-36:2 (“*you could have very different algorithms applied even to the same language or different combinations of algorithms applied to the same language.*”). Mr. Ward further testified that there may be differences in “various abbreviations,” and that “formatting differences in punctuation” “might be different in different countries or in different places even within the same language.” *Id.*, 34:20-24.

152. With respect to the challenges of combining multiple different language models, Mr. Ward testified that “[i]f you’re dealing with different languages, you

may need to deal with additional characters or different characters or characters written in a different form.” *Id.*, 34:1-35:3. Indeed, Mr. Ward testified that:

Depending on the language, some languages, the way writing is taught is more strict about stroke order and stroke direction than in the educational systems commonly used in other countries or other languages. If you're dealing with multiple languages, you may need to deal with multiple languages or multiple character sets at the same time which might even have very similar input or handwriting forms or characters that can be in one language or another language.

Id., 35:3-13.

153. As Mr. Ward explained, “[y]ou may have a language that incorporates multiple algorithms even for what you might consider the same steps such as recognizing shapes or user input.” *Id.*, 36:10-13. Mr. Ward further confirmed that even the process of a basic dictionary lookup may differ between different languages, explaining that:

If you had a word list or dictionaries, those may be organized in different fashions in different languages. The alphabetization or dictionary sorting order of a language which nominally uses the same character set or a similar character set or another could be different. Dictionaries can be arranged around strokes in the characters rather than letters.

Id., 35:14-21. Mr. Ward further explained that “[i]f you're dealing with multiple languages, there are also things that tend to be associated with a particular language, not just a particular education for diacritical marks like umlauts in German, accents in French.” *Id.*, 51:24-52:3.

154. Mr. Ward even specifically testified about challenges that may occur when attempting to design an algorithm which could process both English and Japanese characters, explaining that “you could have the same handwritten input that would ... just as plausibly [be] a Roman – we’ll say Roman letter character or a Katakana character.” *Id.*, 50:11-21. And further that there may be “differences in the behavior of people writing the connected strokes,” including “which stroke order or stroke direction was more consistent in Katakana and also Kanji characters versus what we experienced in actual use in the U.S. with Roman alphabet characters.” *Id.*, 50:1-51:5. As Mr. Ward testified, “some of the problems would be that depending on the language ... you may have wider set of shapes including a number of just ambiguous inputs” noting that “Katakana had something like 44 characters and was a syllabary.” *Id.*, 51:10-14.

155. Mr. Ward further testified about differences in the post-processing portions of any recognition algorithm used with between different languages such as English and Japanese, explaining that “you would probably have a different dictionary or different set of context phrases used for account Katakana and Japanese and for English” (*id.*, 52:20-53:10) and “there may be other context differences as well” (*id.*, 53:17-18).

156. Mr. Ward further testified that different language recognition models may utilize system memory differently, as such that “ROM space or table space” would be different. *Id.*, 57:9-15.

157. Given all of the challenges discussed above, including those identified by both Mr. Andersson, and Petitioners’ own expert Mr. Ward, a POSITA would not have been motivated to combine *Arai* (a Japanese patent application directed to a handwriting recognition algorithm for the Japanese language), with *Fenwick* (a spacing correction algorithm for the English language). Nor would a POSITA have had a reasonable expectation of success in attempting to combine these very different methods, particularly in view of the significant differences between these types of algorithms discussed above.

158. Importantly, neither Petitioners, nor their expert Mr. Ward, address these challenges, particularly the challenges in attempting to combine a Japanese handwriting recognition and post-processing algorithm with an English spacing correction algorithm. Nor do Petitioners or their expert Mr. Ward explain how or why a POSITA would have a reasonable expectation of success in doing so, particularly in view of the challenges discussed above.

159. Instead, Petitioners and their expert Mr. Ward merely conclude (without support), that combining *Arai* and *Fenwick* would be obvious as simply “applying a known technique” (i.e. *Fenwick*) to “a known method” (i.e., *Arai*), “to yield

predictable results.” Petition at 32 (citing Ward Decl., ¶140). But Petitioners assertions are contradicted by the record evidence, including Mr. Ward’s *own* deposition testimony.

160. **Third**, Petitioners arguments regarding motivation to combine lack merit. Specifically, Petitioners assert that “[a] POSITA would have been motivated to combine Arai and Fenwick, because Arai seeks to solve the problem of a user being ‘forced to perform other input operations in addition to the actual intended character input, adversely affecting a natural use experience.’” Petition at 31 (citing Arai [0004]). As Petitioners explain, “[a] POSITA would have been motivated to incorporated Fenwick’s teachings regarding automatically sending recognized words to an application into Arai because this would have furthered Arai’s purpose of having a natural, uninterrupted flow in entering characters without having to press extra keys.” *Id.*, (citing Ward Decl., ¶139).

161. But Petitioners’ argument that automatically sending words to an application would further *Arai’s* purported goal of improving natural use experience is *irrelevant* to limitation [g], which relates to “removing the first sequence portion from the memory storage buffer,” *not* automatically sending character strings to an application. Indeed, the requirement of automatically sending sequence portions to an application is found in the next claim limitation (i.e. limitation 1[h]). Importantly, neither Petitioners, nor their expert Mr. Ward explain why a POSITA would be

motivated to combine *Arai*'s recognition algorithm with *Fenwick*'s feature of automatically deleting memory buffers.

162. Automatically flushing the associated memory buffer is not a pre-requisite to automatically sending a sequence portion to an application, as an algorithm could choose to keep identified sequences in memory for additional contextual analysis, or simply flush the memory buffer based on some other criteria, such as an automatic timeout, or after a certain number of characters have been entered, or based on some other contextual information.

163. In short, as discussed above, a POSITA would not have been motivated to combine *Arai* and *Fenwick* to achieve the '428 patent's method, and would not have had a reasonable expectation of success in doing so. Thus, once again, *Arai* in view of *Fenwick* fail to disclose or suggest limitation [g].

b. Claim 2

164. Claim 2 depends on claim 1, and thus Petitioners have failed to demonstrate that *Arai* in view of *Fenwick* disclose or render obvious the limitations of this claim for the same reasons discussed with respect to claim 1.

c. Claim 3

165. Claim 3 depends on claim 1, and thus Petitioners have failed to demonstrate that *Arai* in view of *Fenwick* disclose or render obvious the limitations of this claim for the same reasons discussed with respect to claim 1.

166. Moreover, *Arai* in view of *Fenwick* further do not disclose or render obvious the additional limitations of claim 3. *Id.* Specifically, claim 3 requires “The method of claim 1, further comprising: determining at least a second candidate sequence based on the first user input; and presenting at least the second candidate sequence to the user.” ’428 patent, cl. 3. However, neither *Arai*, nor *Fenwick*, disclose presenting any “second candidate sequence” to the user.

167. While *Arai* discloses calculating the chain probabilities of character sequences, and even gives an example of multiple different candidate combinations being generated (*see, e.g., Arai* [146]), *Arai*’s specification explains that only the “the candidate character string with the highest chain probability” is displayed to the user for editing (*see Arai* [146], [152]). Thus, *Arai* does not disclose presenting any “second candidate sequence” to the user. Nor does *Fenwick* disclose any disclosure of multiple candidate sequences to the user.

168. Moreover, Petitioners arguments for why *Arai* purportedly discloses this limitation lack merit. Specifically, as with claim 1, Petitioners cite to the **wrong** disclosure in *Arai*, once again attempting to pass off *Arai*’s initial character

recognition process as somehow generating “candidate sequences.” Petition at 36-37. For example, Petitioners cite to an example in *Arai* where initial evaluation values are calculated for the individual character entered ‘ㄱ’, and then note that *Arai* discloses allowing a user to select individual characters displayed in the “character buffer area 61,” which allows “a group of the first through fourth candidates for that character” to be displayed. *Id.*, 36 (citing *Arai* [0086]).

169. But as evident from the language Petitioners cite, this is not a list of hypothetical candidate character sequences (i.e., potential words or strings of characters the user may be trying to enter), this is simply a display of the rank-ordered recognition results for the *individual character* ‘ㄱ’ which were determined during the initial individual character recognition step in *Arai*. *Arai* [0086] (“a group of the first through fourth candidates *for that character* will be displayed”).

170. Because *Arai* does not disclose or suggest this additional limitation, and Petitioners do not point to any disclosure in *Fenwick* for this additional limitation, *Arai* in view of *Fenwick* do not disclose or render obvious the additional limitations of claim 3.

d. Claim 7

171. Claim 7 depends on claim 1, and thus Petitioners have failed to demonstrate that *Arai* in view of *Fenwick* disclose or render obvious the limitations of this claim for the same reasons discussed with respect to claim 1.

e. Claim 8

172. Claim 8 depends on claim 1, and thus Petitioners have failed to demonstrate that *Arai* in view of *Fenwick* disclose or render obvious the limitations of this claim for the same reasons discussed with respect to claim 1.

f. Claim 9

173. *Arai* in view of *Fenwick* do not disclose or render obvious the limitations of claim 9. Claim 9 is reproduced below:

9[pre]: At least one non-transitory computer-readable storage medium storing instructions for data input, wherein the data input is performed at a wireless device having a processor and a memory, comprising:

9[a]: receiving a first user input and at least a second user input;

9[b]: wherein the first and second user input are handwritten input received on a handwriting input area of the wireless device;

9[c]: determining a first sequence based on a best match of the first and second user inputs, wherein the first sequence includes at least the first input;

9[d]: storing the first sequence to a memory of the mobile device;

9[e]: automatically determining that at least a portion of the sequence matches a word or phrase in a database of acceptable words and/or phrases without the user inputting a space or selecting a word; and

9[f]: sending the first sequence to an application or service running on, or accessible by, the wireless device, wherein the sending comprises removing the first sequence from the memory storage buffer, and wherein the sending is performed in the absence of user editing of the first sequence.

174. As shown above, independent claim 9 of the '428 patent is the system version of the '428 patent's method claim 1, and thus *Arai* in view of *Fenwick* do not disclose or render obvious claim 9 for similar reasons as discussed above with respect to claim 1.

175. For example, with respect to limitation 9[c], “determining a first sequence based on a best match of the first and second user inputs, wherein the first sequence includes at least the first input,” this limitation parallels limitation 1[d] of claim 1 (i.e., “generating a first candidate sequence, wherein the first candidate sequence is a best match of one or more recognized sequences”). And as Petitioners did with limitation 1[d] of claim 1, Petitioners once again mischaracterize *Arai*'s disclosure to identify the **wrong** disclosure in *Arai* for limitation 9[c] of claim 9, once again pointing to *Arai*'s initial character recognition process for this limitation, rather than appropriately pointing to *Arai*'s post-processing unit's later calculation of the chain probabilities for the entered character string. *See* Petition at 41-42.

176. With respect to limitation 9[e], which requires “automatically determining that at least a portion of the sequence matches a word or phrase in a database of acceptable words and/or phrases without the user inputting a space or selecting a

word,” this limitation parallels limitation 1[f] of claim 1 (i.e., “identifying at least a first sequence portion of the first candidate sequence wherein the identifying is at least partially based on one or more rules or dictionaries of words accessible to the mobile device.”). And as Petitioners did with limitation 1[f] of claim 1, Petitioners once again mischaracterize the disclosure in *Arai* to point to *Arai*’s sole post-processing step of calculating the chain probabilities as satisfying this limitation. Petition at 42-45. But as with limitation 1[f], Petitioners fail to point to *any* disclosure in *Arai* of determining whether any “portion” of an entered character sequence matches anything.

177. Specifically, with respect to limitation 9[e], Petitioners assert that “[b]ecause *Arai* continues post-processing while the user continues handwritten character input, a POSITA would have understood that *Arai* determines that at least a portion of the sequence matches a word or phrase without the user entering a space or selecting a word.” Petition at 45 (citing Ward Decl., ¶181). But once again, Petitioners are wrong. *Arai* clearly discloses only determining whether a **full character string** appears in the dictionary lookup, not whether any sub-portions of any character string matches any database or dictionary. *Arai* [146] (showing example outputs of *Arai*’s post-processing unit 34). Thus, as with limitation 1[f] of claim 1, *Arai* fails to disclose or suggest limitation 9[e] of claim 9.

178. Finally, with respect to limitation 9[f], which requires “sending the first sequence to an application or service running on, or accessible by, the wireless device, wherein the sending comprises removing the first sequence from the memory storage buffer, and wherein the sending is performed in the absence of user editing of the first sequence,” this limitation parallels limitations 1[g] and 1[h] of claim 1 (i.e., “removing the first sequence portion from the memory storage buffer, wherein the removing is performed in the absence of receiving user input editing the first candidate sequence, representing a space character, or selecting a word” and “sending the first sequence portion to an application running on the mobile device.”). And as Petitioners did with limitations 1[g] and 1[h] of claim 1, Petitioners rely on the combination of *Arai* and *Fenwick* to disclose or suggest limitation 9[f]. Petition at 45-47. But as with limitations 1[g] and 1[h] of claim 1 discussed above, *Arai* and *Fenwick* fail to disclose or suggest limitation 9[f].

179. Moreover, as discussed above, a POSITA would not have been motivated to combine *Arai* and *Fenwick*, and would not have had a reasonable expectation of success in doing so.

g. Claim 12

180. Claim 12 depends on claim 9, and thus Petitioners have failed to demonstrate that *Arai* in view of *Fenwick* disclose or render obvious the limitations of this claim for the same reasons discussed with respect to claim 9.

h. Claim 13

181. Claim 13 depends on claim 9, and thus Petitioners have failed to demonstrate that *Arai* in view of *Fenwick* disclose or render obvious the limitations of this claim for the same reasons discussed with respect to claim 9.

B. Ground 2: Petitioners Fail To Show That Claims 9 and 12-13 Would Have Been Obvious Based on Arai In View Of Fenwick, And In Further View Of Baker

182. In their second ground for invalidity, Petitioners assert that *Arai* in combination with *Fenwick* and *Baker* renders obvious claims 9 and 12-13 of the '428 patent. Petition at 48- 51. As Petitioners explain, their second ground for invalidity is identical to their arguments for claims 9 and 12-13 discussed above for Petitioners first ground for invalidity, except that Petitioners' second ground further adds *Baker* for the disclosure of the limitation "a database of acceptable words and/or phrases" found in claim 9. *Id.*

183. Because Petitioners do not rely on the addition of *Baker* to disclose or suggest any other limitations beyond "a database of acceptable words and/or phrases," Petitioners have not shown that *Arai* in combination with *Fenwick* and *Baker* renders obvious claims 9 and 12-13 of the '428 patent for the same reasons discussed above with respect to Petitioners' first invalidity ground.

184. Additionally, as with the combination of *Arai* and *Fenwick*, a POSITA would not have been motivated to combine the Japanese patent application *Arai* (disclosing

a Japanese handwriting recognition algorithm) with a European patent specification *Baker* (disclosing word recognition algorithms for Latin or Cyrillic-based words), because both the initial recognition process, as well as the post-processing that occurs on any recognized characters, differs substantially between different languages, and even differs substantially between different dialects between the same language. *See, e.g.,* Ex.2009 (Andersson Decl.) ¶¶3-15; Ex.2010 (Ward Dep Tr.), 28:1-10, 28:24-29:1, 29:11-19. 32:1-12, 33:1-7, 34:2-6, 34:8-24, 35:3-21, 35:24-36:2, 36:10-13, 50:1-51:14, 51:24-52:3, 52:20-53:10, 53:17-18, and 57:9-15.

185. Once again, neither Petitioners, nor their expert Mr. Ward, address these challenges, particularly the challenges in attempting to combine a Japanese handwriting recognition and post-processing algorithm with an algorithm for Latin or Cyrillic-based words. Nor do Petitioners or their expert Mr. Ward explain how or why a POSITA would have a reasonable expectation of success in doing so, particularly in view of the challenges discussed above.

186. Instead, once again, Petitioners and their expert Mr. Ward merely conclude (without support), that combining *Arai* and *Baker* would be obvious as simply “applying a known technique” (i.e. *Baker*) to “a known device” (i.e., *Arai-Fenwick*), “to yield predictable results.” Petition at 50 (citing Ward Decl., ¶197). But as discussed above, Petitioners assertions are contradicted by the record evidence, including Mr. Ward’s *own* deposition testimony.

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true.

April 25, 2025



Eric Koskinen