

**UNITED STATES PATENT AND TRADEMARK OFFICE**

---

**BEFORE THE PATENT TRIAL AND APPEAL BOARD**

---

PALO ALTO NETWORKS, INC.,  
Petitioner,

v.

TAASERA LICENSING LLC,  
Patent Owner.

---

Case IPR2023-00443  
Patent 6,842,796 B2

---

**DECLARATION OF JOHN R. BLACK, JR., PH.D.**

# TABLE OF CONTENTS

|                                                                                                                                                                                                                                                                                                                  | Page |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
| I. Introduction .....                                                                                                                                                                                                                                                                                            | 1    |
| II. Background and Qualifications.....                                                                                                                                                                                                                                                                           | 2    |
| III. Legal Standards Used In My Analysis.....                                                                                                                                                                                                                                                                    | 5    |
| A. Prior Art .....                                                                                                                                                                                                                                                                                               | 6    |
| B. Person of Ordinary Skill in the Art.....                                                                                                                                                                                                                                                                      | 6    |
| C. Obviousness .....                                                                                                                                                                                                                                                                                             | 8    |
| IV. The '796 Patent .....                                                                                                                                                                                                                                                                                        | 10   |
| A. Overview of the Challenged '796 Patent.....                                                                                                                                                                                                                                                                   | 10   |
| B. Relevant Prosecution History .....                                                                                                                                                                                                                                                                            | 13   |
| V. Claim Construction .....                                                                                                                                                                                                                                                                                      | 14   |
| VI. Summary of Cited Prior Art References .....                                                                                                                                                                                                                                                                  | 17   |
| A. Sheth (EX1004) .....                                                                                                                                                                                                                                                                                          | 17   |
| B. Richardson (EX1005) .....                                                                                                                                                                                                                                                                                     | 19   |
| C. Madnick (EX1006) .....                                                                                                                                                                                                                                                                                        | 23   |
| D. Groner (EX1007) .....                                                                                                                                                                                                                                                                                         | 24   |
| VII. Opinions Regarding Obviousness.....                                                                                                                                                                                                                                                                         | 26   |
| A. <u>Ground 1</u> : Claims 1-10, 12-21, 23, and 25 Would Have Been Obvious Over Sheth (EX1004).....                                                                                                                                                                                                             | 26   |
| 1. Independent Claims 1, 12, 23, and 25 .....                                                                                                                                                                                                                                                                    | 26   |
| a. Claim 1 .....                                                                                                                                                                                                                                                                                                 | 26   |
| i. “A method of automatically processing an input sequence of data symbols, the method comprising the steps of” .....                                                                                                                                                                                            | 26   |
| ii. “identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression” ..... | 28   |
| iii. “identifying at least a portion of information associated with the at least one regularly identifiable expression; and” .....                                                                                                                                                                               | 29   |

|                                                                                                                                                                                                                                                                                                                  |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| iv. “extracting the portion of information.” .....                                                                                                                                                                                                                                                               | 30 |
| b. Claim 12 .....                                                                                                                                                                                                                                                                                                | 31 |
| c. Claim 23 .....                                                                                                                                                                                                                                                                                                | 33 |
| d. Claim 25 .....                                                                                                                                                                                                                                                                                                | 35 |
| 2. Dependent Claims 2-10 and 13-21 .....                                                                                                                                                                                                                                                                         | 37 |
| a. Claims 2 and 13.....                                                                                                                                                                                                                                                                                          | 37 |
| b. Claims 3 and 14.....                                                                                                                                                                                                                                                                                          | 37 |
| c. Claims 4 and 15.....                                                                                                                                                                                                                                                                                          | 40 |
| d. Claims 5 and 16.....                                                                                                                                                                                                                                                                                          | 41 |
| e. Claims 6 and 17.....                                                                                                                                                                                                                                                                                          | 43 |
| f. Claims 7 and 18.....                                                                                                                                                                                                                                                                                          | 43 |
| g. Claims 8 and 19.....                                                                                                                                                                                                                                                                                          | 44 |
| h. Claims 9 and 20.....                                                                                                                                                                                                                                                                                          | 45 |
| i. Claims 10 and 21.....                                                                                                                                                                                                                                                                                         | 45 |
| B. <u>Ground 2</u> : Claims 1-10, 12-21, 23, and 25 Would Have Been Obvious Over Richardson (EX1005) and Madnick (EX1006) .....                                                                                                                                                                                  | 46 |
| 1. Richardson-Madnick Combination .....                                                                                                                                                                                                                                                                          | 46 |
| 2. Independent Claims 1, 12, 23, and 25 .....                                                                                                                                                                                                                                                                    | 49 |
| a. Claim 1 .....                                                                                                                                                                                                                                                                                                 | 49 |
| i. “A method of automatically processing an input sequence of data symbols, the method comprising the steps of” .....                                                                                                                                                                                            | 49 |
| ii. “identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression” ..... | 50 |
| iii. “identifying at least a portion of information associated with the at least one regularly identifiable expression; and” .....                                                                                                                                                                               | 54 |
| iv. “extracting the portion of information.” .....                                                                                                                                                                                                                                                               | 55 |
| b. Claim 12 .....                                                                                                                                                                                                                                                                                                | 59 |
| c. Claim 23 .....                                                                                                                                                                                                                                                                                                | 63 |

|                                                                                                                  |    |
|------------------------------------------------------------------------------------------------------------------|----|
| d. Claim 25 .....                                                                                                | 65 |
| 3. Dependent Claims 2-10 and 13-21 .....                                                                         | 68 |
| a. Claims 2 and 13.....                                                                                          | 68 |
| b. Claims 3 and 14.....                                                                                          | 69 |
| c. Claims 4 and 15.....                                                                                          | 70 |
| d. Claims 5 and 16.....                                                                                          | 71 |
| e. Claims 6 and 17.....                                                                                          | 72 |
| f. Claims 7 and 18.....                                                                                          | 73 |
| g. Claims 8 and 19.....                                                                                          | 74 |
| h. Claims 9 and 20.....                                                                                          | 76 |
| i. Claims 10 and 21.....                                                                                         | 77 |
| C. <u>Ground 3</u> : Claims 11, 22, and 24 Would Have Been Obvious Over<br>Richardson, Madnick, and Groner ..... | 78 |
| 1. Richardson-Madnick-Groner Combination.....                                                                    | 78 |
| 2. Claims 11 and 22 .....                                                                                        | 81 |
| 3. Claim 24 .....                                                                                                | 82 |
| D. No Secondary Considerations Exist .....                                                                       | 85 |
| VIII. Conclusion .....                                                                                           | 86 |
| Appendix                                                                                                         |    |

## CITED EXHIBITS

| Ex. No. | Brief Description                                                                                            |
|---------|--------------------------------------------------------------------------------------------------------------|
| 1001    | U.S. Patent No. 6,842,796                                                                                    |
| 1002    | U.S. Patent No. 6,842,796 File History (U.S. Appl. No. 09/898,289)                                           |
| 1004    | U.S. Patent No. 6,311,194 B1 to Sheth et al. (filed Aug. 21, 2000; issued Oct. 30, 2001) (“Sheth”)           |
| 1005    | U.S. Patent No. 6,278,996 B1 to Richardson et al. (filed Mar. 30, 1998; issued Aug. 21, 2001) (“Richardson”) |
| 1006    | U.S. Patent No. 5,913,214 to Madnick et al. (filed Aug. 8, 1996; issued Jun. 15, 1999) (“Madnick”)           |
| 1007    | U.S. Patent No. 6,507,643 B1 to Groner (filed Mar 16, 2000; issued Jan. 14, 2003) (“Groner”)                 |

I, John R. Black, Jr., declare as follows:

**I. Introduction**

1. I have been retained by Palo Alto Networks, Inc. (“Petitioner”) in connection with its request for *inter partes* review of U.S. Patent No. 6,842,796 (“the ’796 patent”). A copy of the ’796 patent has been marked EX1001. I have reviewed and am familiar with the ’796 patent, which generally relates to extracting information from documents by matching regular expressions.

2. I have been asked to provide my opinions regarding the patentability of all claims (1-25) of the ’796 patent (the “challenged claims”). This declaration discusses my background and qualifications, the legal standards used in my analysis, an overview of the ’796 patent from the perspective of a person of ordinary skill in the art as of the patent’s alleged priority date (a “POSITA”), and my opinions regarding the patentability of the challenged claims.

3. My work in this proceeding is being charged to Petitioner at a standard rate of \$735 per hour. My compensation is in no way contingent on the substance of my opinions or the outcome of this proceeding.

4. As set forth more fully below, it is my opinion that the challenged claims of the ’796 patent are unpatentable as obvious based on the following grounds, which set forth the prior art references that a POSITA would have

considered, either alone or in combination, that are cited in the table of cited exhibits above:

| Ground | Claims              | References                  |
|--------|---------------------|-----------------------------|
| 1      | 1-10, 12-21, 23, 25 | Sheth                       |
| 2      | 1-10, 12-21, 23, 25 | Richardson, Madnick         |
| 3      | 11, 22, 24          | Richardson, Madnick, Groner |

## II. Background and Qualifications

5. A current copy of my CV is attached as an Appendix.

6. I am an Associate Professor of Computer Science at the University of Colorado, Boulder. I received a B.S. in Mathematics and Computer Science from the California State University at Hayward (now "California State University, East Bay") in 1988. I received an M.S. in Computer Science in 1997, and a Ph.D. in Computer Science in 2000, both from the University of California at Davis.

7. I have taught more than 60 classes in computer science, on subjects including data structures, algorithms, networking, operating systems, software engineering, security, cryptography, discrete mathematics, and quantum computing. I have authored or coauthored more than 20 publications, primarily on issues relating to computer security. I have been involved with computers for over 40 years in both commercial and academic capacities.

8. My earliest interest was in networks and security. My first memories in this regard were around 1975 when a group of friends and I learned about the telephone network and its security. A few years later, personal computers became available and I spent most of my free time studying, programming, and modifying them. I worked extensively with various networking products throughout the 1980s, and developed an interest in cryptography soon thereafter. Although my undergraduate institution had no courses in cryptography or security in the 1980s, I decided to pursue self-study at the time, and opted to double major in Computer Science and Mathematics because cryptography is a blend of these two subject areas.

9. After earning my B.S. degree in 1988, I worked for six years at Ingres Corp as a software developer, writing and reviewing code written in C. My work primarily was directed at transaction logging, data type support, and security. The data type facility had support for wildcard matching for database queries, which used regular expressions and I was responsible for maintaining and extending this code.

10. In 1995, I began my Ph.D. at UC Davis under cryptographer Phillip Rogaway. My area of focus was cryptography and security and my research involved encryption, authentication, hash functions, and network security.

11. After graduation I took a position as an Assistant Professor at the University of Nevada at Reno. In the Fall of 2001, I taught the networking class there, which included coverage of Ethernet, interior gateway protocols, exterior gateway protocols, ARP, DHCP/BOOTP, IP, UDP, TCP, HTTP, SMTP and other protocols. In 2001, a graduate student and I looked at the security of ARP and invented a new protocol "AuthARP" to add security to the protocol.

12. In 2002, I moved to the University of Colorado at Boulder where I am currently employed. In the Fall of 2002, I co-designed and co-taught a new course called "Foundations of Computer and Network Security," which included descriptions of security issues around both wired and wireless security challenges, mostly for public-facing network services including the world-wide web. I have taught this class seven more times since then, including modern topics such as wireless networking, the Internet of Things, and so forth.

13. In my career at the University of Colorado, I have published several more papers in the area of cryptography and network security. I have taught more than 30 courses in network security and cryptography, and have graduated several PhD students in these areas.

14. I also have worked for a consulting company at times, writing software on contract basis. Although most projects are covered by NDAs, many involved networks and/or aspects of computer security.

15. In 2011, I began technical consulting for a local company called Cardinal Peak, which focuses primarily on video encoding and delivery systems. My work for Cardinal Peak has largely been directed to video encoding, transcoding, compression, encryption, and DRM, but also has included networking projects and embedded device work. For example, I designed the security system for the Pro1 smart thermostat, implemented the DRM for the Yonder Music App, worked on 802.1X code for smart dog collars, and helped design the cryptography used in Fitbit devices for wireless transfer of a Fitbit watch to a phone or laptop. I also wrote a Command Line Interpreter for a milling machine control module that had full support for POSIX regular expressions.

16. In 2016, I took a leave of absence from the University of Colorado to start a company named "SecureSet" in Denver, Colorado. The objective of SecureSet is to take reasonably proficient technical people and turn them into computer and network security specialists via five months of intensive training. SecureSet was sold to WeWork in 2019 and continues to offer computer security classes today.

### **III. Legal Standards Used In My Analysis**

17. I am not a patent attorney, nor have I independently researched patent law. Counsel for Petitioner have explained certain legal standards to me that I have relied upon in forming my opinions set forth in this Declaration. I set forth

some of these legal standards below and may set forth additional legal standards in my analysis in later sections of this Declaration.

**A. Prior Art**

18. I have been informed that the law provides certain categories of information, known as prior art, that may be used to render patent claims obvious. I understand that the reference materials I discuss in this Declaration are prior art at least because they would have been available to members of the public as of March 14, 2001, and are relevant to the subject matter of the '796 patent.

19. The references I discuss in this Declaration are from the same field of endeavor as the claimed invention (even if they address a different problem) and/or are reasonably pertinent to the problem faced by the named inventors (even if they are not in the same field of endeavor as the claimed invention).

**B. Person of Ordinary Skill in the Art**

20. I understand that the '796 patent issued from Application No. 09/898,289, filed July 3, 2001. I understand that, during prosecution, the applicants alleged the “conception of an invention falling within independent claims 1, 12, 23-25 and at least one or more dependent claims, at least as early as March 14, 2001.” EX1002, 116. I have not been asked to analyze whether the applicants properly demonstrated conception as of March 14, 2001. For purposes

of my analysis in this Declaration, however, I have used this date as the relevant priority date because all prior art cited herein was available by then.

21. I understand that the assessment of the patentability of the claims of the '796 patent must be undertaken from the perspective of a hypothetical POSITA as of the priority date of the '796 patent. I understand that the POSITA is a hypothetical person who is presumed to have known all the relevant art as of the priority date. I understand factors that may be considered in determining the level of ordinary skill of a POSITA include (i) the types of problems encountered in the art, (ii) prior art solutions to those problems, (iii) rapidity with which innovations are made, (iv) sophistication of the technology, and (v) educational level of active workers in the field. I understand that in a given case, every factor may not be present, and one or more factors may predominate.

22. I understand that the hypothetical POSITA to which the claimed subject matter pertains would have had the capability of understanding the scientific and technological principles applicable to the pertinent art. I further understand that a POSITA is also a person of ordinary creativity, not an automaton. Further, I understand that the hypothetical POSITA may comprise a team.

23. Based on these factors, my knowledge and expertise, and the prior art to the '796 patent, it is my opinion that a POSITA at the time of the relevant priority date would have had a bachelor's degree in computer science, computer

engineering or an equivalent, as well as four years of industry experience and would have had a working knowledge of existing data extraction methods and programming languages including perl and flex. Additional graduate education could substitute for professional experience, or significant experience in the field could substitute for formal education.

24. I meet this definition of a POSITA today and met this definition by March 14, 2001.

**C. Obviousness**

25. I have been informed that, even if every element of a claim is not found explicitly or implicitly in a single prior art reference, the claim may still be unpatentable if the differences between the claim and the prior art are such that the claim as a whole would have been obvious to a POSITA at the time the invention was made. For purposes of obviousness, I understand that a POSITA may rely on a single prior art reference or on multiple references in combination.

26. I have been informed that the following four factors are considered when determining whether a patent claim would have been obvious to a POSITA: (a) the level of ordinary skill in the art; (b) the scope and content of the prior art; (c) the differences between the prior art and the claim; and (d) any “secondary considerations” tending to prove nonobviousness. These secondary considerations, which I understand are also called “objective indicia” or “objective evidence,” may

include factors such as: (i) the invention's satisfaction of a long-felt unmet need in the art; (ii) unexpected results of the invention; (iii) skepticism of the invention by experts; (iv) commercial success of an embodiment of the invention; and (vii) praise by others for the invention. I have also been informed that there must be an adequate nexus or connection between the evidence that is the basis for an asserted secondary consideration and the scope of the invention claimed in the patent.

27. I understand that when every limitation of a claim is disclosed in the cited prior art references, the question of obviousness turns on whether a hypothetical person of ordinary skill in the art would have been motivated to combine those teachings to derive the claimed subject matter with a reasonable expectation of success. Further, I understand that obviousness does not require absolute predictability. Only a reasonable expectation that the beneficial result will be achieved is necessary to show obviousness.

28. I have been informed that a claimed invention can be rendered obvious by the combination of teachings in the prior art even if there is no explicit teaching to combine them. Instead, any problem known in the field at the time of the alleged invention can provide a sufficient rationale to combine the elements of the prior art in the manner claimed in the patent.

29. I have been informed that examples of sufficient rationales for establishing obviousness include the following:

- combining prior art elements according to known methods to yield predictable results;
- substituting known elements for other known elements to obtain predictable results;
- using a known technique to improve similar devices, methods, or products in the same way;
- choosing from a finite number of identified, predictable solutions that would be obvious to try; and
- providing some teaching, suggestion, or motivation to modify the prior art reference or to combine teachings in prior art references to arrive at the claimed invention.

#### **IV. The '796 Patent**

##### **A. Overview of the Challenged '796 Patent**

30. The '796 patent relates to “information extraction from data streams such as text and spoken documents.” EX1001, 7-10. Notably, the specification’s “Background of the Invention” admits that previous data-extraction techniques could “identify all occurrences of certain classes of words in a document,” but alleges that there remained “a need for data processing techniques which explicitly identify portions of data that are sought to be identified, rather than only implicitly identifying, or tagging, such portions of data.” *Id.*, 1:24-27, 1:46-49.

31. The '796 purports to solve this alleged problem by “using stereotypical phrases or ‘regular expressions’ to identify the information-bearing portions of a document, and thereby explicitly identify the information found therein.” *Id.*, 2:65-3:2. The patent states that regular expressions “comprise symbolic strings, typically word sequences, that can be constructed by the basic operations of ‘and’ and ‘or’ acting on an atomic alphabet of symbols.” *Id.*, 3:2-8. For example, a message could include “the phrase ‘call me back at 555-1234.’” *Id.*, 11:27-29. The alleged invention “determines that the phrase ‘call me back at’ matches one of the enumerated expressions,” and thus “the information associated with the phrase ‘call me back at,’ namely, ‘555-1234,’ is identified as the information-bearing portion of the expression, and extracted.” *Id.*, 11:29-34.

32. The '796 patent admits that regular expressions were known in the prior art and “have been discussed in Aho et al., ‘Compilers: Principles, Techniques, and Tools,’ Addison-Wesley, 1985.” *Id.*, 3:2-4. This was the standard textbook on compilers in the mid-1980’s and was my textbook when I first learned about regular languages and regular expressions around the same time. This book is also where I first learned the programming language “lex,” the forebear of “flex” (which simply stands for “fast lex”).

33. The '796 patent further admits that regular expressions may “be stored in one of several common programming languages, e.g., as a ‘flex’ or ‘perl’

program,” both of which “have built-in support for regular expressions” (*id.*, 2:23-27) and were described in prior-art references published in 1992 and 1996, respectively (*id.*, 3:26-31). A POSITA as of March 2001 would have been closely familiar with these programming languages and with the use of regular expressions.

34. The ’796 patent has five independent claims. Claim 1 recites:

1. A method of automatically processing an input sequence of data symbols, the method comprising the steps of:  
identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression;  
identifying at least a portion of information associated with the at least one regularly identifiable expression; and  
extracting the portion of information.

35. Claims 12 and 23 restate the same alleged invention as an “[a]pparatus” or “article of manufacture,” respectively. Claim 24 recites the same apparatus as claim 12 but adds a generic “data capture device for obtaining the input sequence of data symbols.” Claim 25 recites the same method as claim 1 without the final extraction step and adding that regular expressions are identified “by comparing the input document to one or more previously-stored regular

expressions to determine if there is a match between a portion of the input document and at least one of the previously-stored regular expressions.”

36. As discussed in greater detail below, the dependent claims generally recite this same step of “comparing” data to “match” (claims 2, 13); “normalizing” data (claims 3, 14); “classes of data symbols” (claims 4, 15); “a characteristic phrase” that precedes or follows information (claims 5, 6, 16, 17); taking any “specified action” (claims 7, 18); “present[ing]” information to a user (claims 8, 19); using the flex, lex, or perl programming languages (claims 9, 20); limiting data to categories including text, audio, or video (claims 10, 21); and limiting data to voice mail (claims 11, 22).

## **B. Relevant Prosecution History**

37. I have reviewed and considered the prosecution history of the ’796 patent, which is marked EX1002. I provide a brief summary of certain key events during the prosecution of the ’796 patent below.

38. The Examiner initially rejected all claims as anticipated or obvious over Adler (U.S. Pat. No. 6,342,901). EX1002, 86-90. In response, the applicants added the clause, “wherein [the at least one regularly identifiable expression / a regular expression] represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression,” to all independent claims. *Id.*, 96-100. The applicants argued that Adler lacks this

limitation because it only teaches the use of “cue words,” which “do not represent patterns that are matchable in accordance with a programming language that supports such patterns, as in the claimed invention.” *Id.*, 101.

39. The Examiner then rejected the claims over Klein (U.S. Pat. App. Pub. No. US 2002/0194026), which teaches “[a] PERL language regular expression parser ... along with a rule set that includes a library of regular expressions that define how information is delimited.” *Id.*, 108-09. In response to the Examiner’s rejection, the applicants did not dispute that Klein reads on the claims. Instead, they submitted a declaration under 37 C.F.R. §1.131 that allegedly shows “conception of an invention falling within independent claims 1, 12, 23-25 and at least one or more dependent claims, at least as early as March 14, 2001, and thus prior to the June 13, 2001 effective date of Klein.” *Id.*, 116. The Examiner then allowed the claims following this declaration. *Id.*, 139.

40. The prior art cited in this Declaration was not considered during prosecution. As discussed below, it is my opinion that the prior art in this Declaration teaches all limitations of the ’796 patent’s claims, including the “wherein” clause that was added during prosecution to distinguish Adler.

## **V. Claim Construction**

41. Counsel for Petitioner has informed me that language in a patent claim should be read in accordance with the ordinary and customary meaning of

such claim as understood by a POSITA in light of the specification and prosecution history of the patent.

42. Counsel for Petitioner has also informed me that only those terms need be construed that are in controversy, and only to the extent necessary to resolve the controversy.

43. In my opinion, the reference to the “pen programming language” in claim 20 of the ’796 patent is an obvious typographical error that should be construed as “perl programming language.”

44. I have been informed that obvious minor typographical and clerical errors in patent claims may be corrected so long as (1) the correction is not subject to reasonable debate based on consideration of the claim language and the specification; and (2) the prosecution history does not suggest a different interpretation of the claims.

45. In this case, the reference to “pen” in claim 20 is clearly a typographical error that should read “perl.” Notably, “pen” is not referenced in any other claim of the ’796, the specification, or the prosecution history.

46. The prosecution history of the ’796 patent confirms that “pen” is a typographical error in the issued claims because the last submitted version of claim 20 before allowance correctly recites “perl.” EX1002, 99. The visual similarity between the letter “n” in “pen” and the contiguous letters “rl” in “perl” suggests

that the inclusion of “pen” in the issued version of the ’796 patent may have resulted from an error in optical character recognition (“OCR”).

47. Moreover, dependent claims 13-22 (which depend from claim 12) recite materially identical limitations as claims 2-11 (which depend from claim 1), and claim 9—the counterpart to claim 20—correctly recites “one of the flex, lex, and perl programming language.” This mirrors the specification’s correct recitation of these same three programming languages. EX1001, 3:22-25 (“In accordance with an illustrative embodiment of the invention, the term ‘regular expression’ is taken to mean any form of pattern that is matchable in the flex, lex, or perl programming languages....”).

48. Thus, in my opinion, correcting “pen” to “perl” would not be subject to reasonable debate to a POSITA, and nothing in the prosecution history suggests a different interpretation.

49. For purposes of my opinions of unpatentability in this Declaration, no other claim construction is necessary because, in my opinion, the challenged claims would have been obvious over the cited references under any reasonable construction and based on their plain and ordinary meaning to a POSITA.

## **VI. Summary of Cited Prior Art References**

50. The prior art references relevant to my analysis are briefly summarized below. I discuss additional elements of each reference in my analysis of how a POSITA would have combined them to arrive at the challenged claims.

### **A. Sheth (EX1004)**

51. I understand that Sheth is prior art to the '796 patent at least because it issued from an application filed on August 21, 2000. EX1004.

52. Sheth states that it “is directed to software, a system and a method for creating a database of metadata (metabase) of a variety of digital media content,” particularly media content on the Internet. *Id.*, 4:54-59. Sheth explains that, “[i]n order to keep the metabase up-to-date with the latest news or movie trailers, metadata extraction from various sources” is provided. *Id.*, 9:66-10:1.

53. Sheth describes an “extraction process during which metadata for digital media is discovered, copied from a Web site, enhanced, and inserted into the present invention’s database.” *Id.*, 10:7-9. Sheth’s extraction process utilizes “WebCrawlers 3 that scan certain Web sites” and “Extractors 7 that do the actual metadata extraction.” *Id.*, 10:10-12. Sheth explains: “Crawling and extraction rules are input to WebCrawler and Extractor programs that traverse Web sites and retrieve digital media metadata from selected pages.” *Id.*, 10:40-42.

54. In Sheth, once a webpage is identified for extraction, “[t]he extractor scans the Web page content for pieces of text that match the pattern specified by the extraction rules.” *Id.*, 11:50-52. Furthermore, “[a]fter the extractor has attempted to find every attribute in its list of extraction rules, it creates an XML document containing attribute-value pairs.” *Id.*, 11:52-54, Fig. 6. Sheth explains that “[t]he ability to retrieve text that matches a particular pattern is provided by the PERL regular expression language.” *Id.*, 11:64-66; *see also id.*, 6:45-47. Examples of extraction rules written in perl are provided in Figures 8 and 9. *Id.*, 11:60-64, 12:23-33, Figs. 8, 9. Figure 8 is reproduced below:

```
# Name : CNNUS
#
producer          [c]CNN
description        [ ](?i)<img src[^>]*alt="Video"[^>]*>([<]*)*<
descriptionReplac1 s/\\[([^\[])*\]\[^\[]*\\//
parentURL          [u](.*)
invalidated        [c]0
surrogate          [ ](?i)([<]*)*</h.>
productionSource   [c]tv
location           [S](?i)/date.*?<p>(.*)\\(.*?\\)
locationReplac1    s/. *?<I>.*?<\\//I>(.*)/$1/
locationReplac2    s/(.*?)--.*?/$1/
locationReplac3    s/. *?<em>.*?<\\//em>(.*)/$1/
mediaType          [c]video
source             [c]http://www.cnn.com
```

*Id.*, Fig. 8.

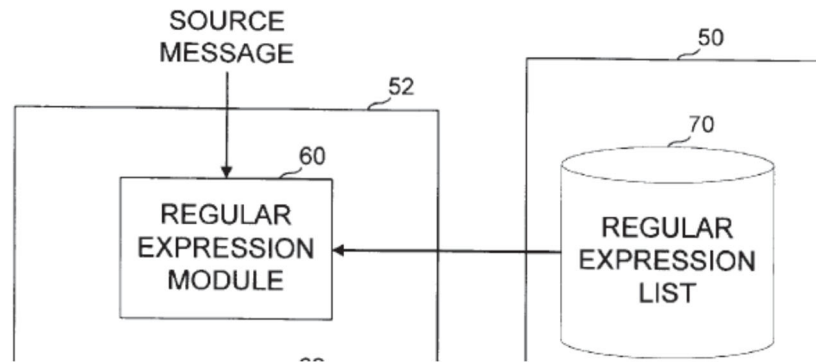
**B. Richardson (EX1005)**

55. I understand that Richardson is prior art to the '796 patent at least because it issued from an application filed in 1998. EX1005.

56. Richardson states that it “is directed to a method and apparatus for automatically recognizing and responding to intent in message text, particularly as used in a large-scale automatic message response system to examine input text and respond appropriately with output text relevant to the writer’s intent.” *Id.*, 1:21-26. To determine the writer’s intent, Richardson’s “system isolates the writer’s critical words which relate to prototypical statements stored in the system.” *Id.*, 2:56-57. Richard explains that “[t]he input message text can be electronic mail and may include text,” among other computer-readable data. *Id.*, 3:19-20.

57. Figure 1 depicts apparatuses for implementing Richardson’s method and system, whereby “[c]ustomer system 12 sends input message 14 to server 16 for processing.” *Id.*, 4:40-41, Fig. 1. “Server 16 is adapted to take an action based on an internal knowledge base and the text of query message 14,” and “[i]f the action is a response message 18, that response message 18 is sent to customer system 12.” *Id.*, 4:44-47. Customer system 12 and server 16 “are general purpose computer systems programmed according to their use and comprising one or more central processing unit” and “memory,” among other components. *Id.*, 5:10-19.

58. Richardson's server 16 comprises, among other components, "knowledge base 36" and "message processing module 38." *Id.*, 4:60-61, 5:4-6, 5:23-24, Figs. 2, 3. In turn, knowledge base 36 comprises "extraction rule set 50," and message processing module 38 comprises "natural language extraction (NLE) module 52," which are partly depicted below in Figure 4. *Id.*, 5:25-29.



*Id.*, Fig. 4 (detail).

59. As Richardson explains, "FIG. 4 shows NLE module 52 comprising a regular expression module 60," and "[e]xtraction rule set 50 is shown comprising a regular expression lexicon 70," among others. *Id.*, 5:29-34. In Richardson, "understanding and responding to text messages begins with accepting the input text message, which is first processed by NLE module 52," whereupon "[r]egular expression module 60 identifies instances of the regular expressions represented in regular expression lexicon 70." *Id.*, 5:34-37, 5:51-57-59. The message and its extracted data are then sent to an intent matching module (IMM). *Id.*, 6:34-36.

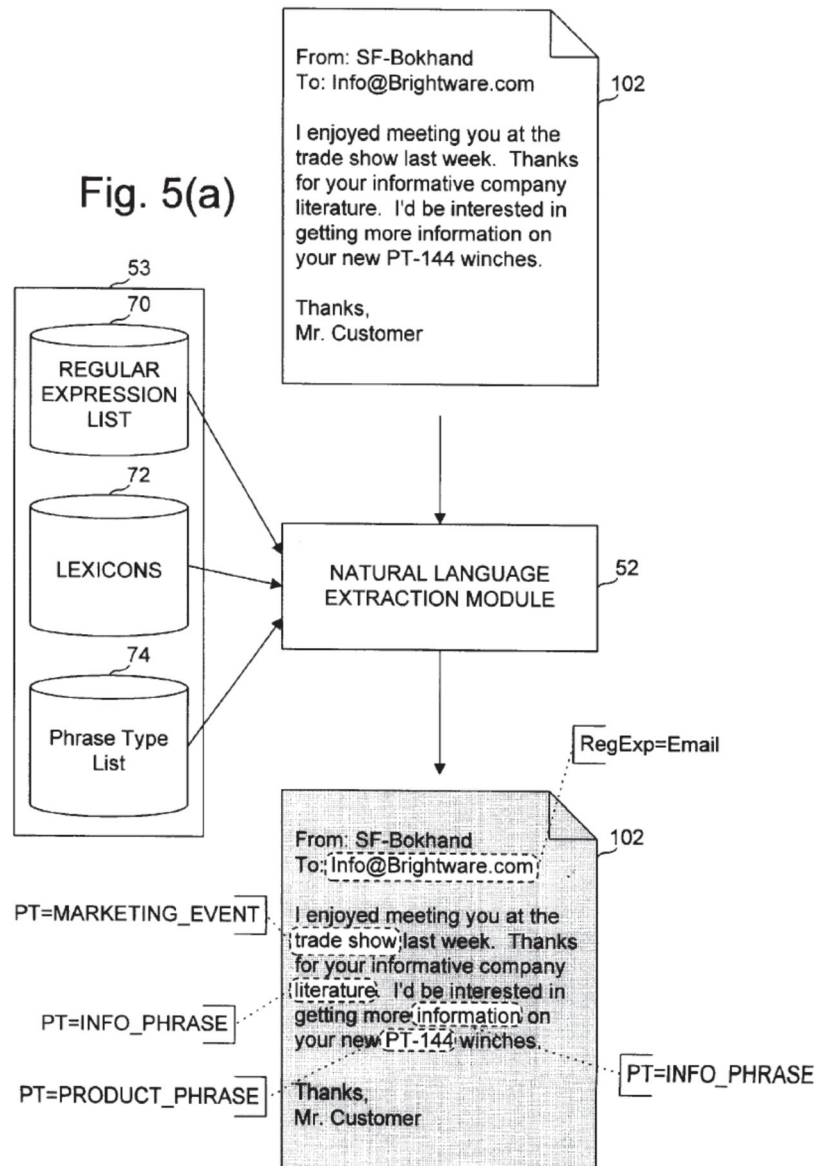
60. Richardson provides a more detailed embodiment in Figs. 5(a) and 5(b), which depict how an “input message is processed by NLE module 52 to generate tags associated with portions of the input message.” *Id.*, 8:42-45. “NLE module 52 uses a regular expression list 104 and a phrase types list 106 in assigning tags to a message,” and “[a]n example of regular expression list 104 ... is presented in Table 1,” reproduced below. *Id.*, 8:50-55.

| TABLE 1                 |       |
|-------------------------|-------|
| Regular Expression List |       |
| Expression              | Type  |
| http://*{ /* }>         | URL   |
| ### ### ####            | Phone |
| [a-zA-Z]@*,*            | Email |
| ...                     | ...   |

*Id.*, 8:59-66. Richardson explains as follows:

Regular expression list 104 is used to isolate and identify character strings in the message which match the expressions in the list.... In the example shown, an e-mail address is matched and tagged with a “RegExp=Email” tag. Of course, in an actual application, more complex regular expressions could be listed, such as a regular expression that identifies not only that it is an e-mail address, but that it is the “To:” address. Other regular expressions will locate telephone numbers, or even the signature lines of an e-mail message.

*Id.*, 9:17-27. Figure 5(a) depicts this process:



*Id.*, Fig. 5(a).

61. After regular expressions are identified, Richardson’s “analysis process continues, as shown in FIG. 5(b),” which involves “database lookups” and other actions based on extracted data. *Id.*, 10:24-30. Richardson explains that “tagged elements of message 102 are passed to action resolution module (‘ARM’) 56” and ultimately to “action execution module (‘AEM’) 57, which performs the

actions listed on list 122, such as creating a response 130.” *Id.*, 10:64-65, 11:14-17. Richardson further explains that “[a] preferred response 130 includes all or part of the original message,” along with “tags and the values for tagged regular expressions and tagged phrase types.” *Id.*, 11:17-23.

**C. Madnick (EX1006)**

62. I understand that Madnick is prior art to the ’796 patent at least because it issued in 1999. EX1006.

63. Madnick states that it “relates to retrieving data from heterogeneous data sources including structured sources and semi-structured sources and, more particularly, extracting data from World Wide Web pages in response to a query phrased in a structured query language.” *Id.*, 1:15-20.

64. In Madnick, data is retrieved from data sources 104 (e.g., webpages) using data receivers 102. *Id.*, 3:60-4:18. To extract data from webpages, Madnick discloses “wrapper generator 614,” which comprises “a command transmitter 618; and a data retriever 620.” *Id.*, 9:45-49. Madnick further discloses “[s]pecification file 706 for a data source 104 [that] describes the commands that must be transmitted to the data source 104 by the command transmitter 618 in order to interact with the data source 104.” *Id.*, 11:11-12:2. “The data retriever 620 then extracts the data requested by the data receiver 102 from the semi structured data source 104 using the specification file.” *Id.*, 13:24-26.

65. An example of Madnick's specification file 706 is in Table 2:

```
REGULAR EXPRESSIONS:
network.Company  \<FONT\ssize=d\>(.*?|s+\W\w+\W#
network.Ticker   \s\((.*?)\</FONT>A#
network.Last     Last\</A\>\s+(.*?)\s+\<A#
network.Last__Trade \{last\strade:\s(.*?)\sEST\W#
network.Low      Day\sRange\</A\>\s\<ID\>\s*(.*&)\s*\s*d#
network.High     Day\sRANGE\</A\>\s\<TD\>.*?\s*(.*?S\s+#
network.Change   Change\</A\>\s+(.*?)\n#
network.Prev__Close Prev.\sClose\<A\>\s(.*?)\n#
network.Tick-Trend Tick\sTrend\</A\>\s+\W+TD\>(.*?\</TD\>#
network.Volume   Volume\</A\>\s+(.*?)\s+\<A#
network.Market   Market\<A\>\s+(.*?)\s+n#
network.Open     Open:\<A\>\s+(.*?)\s+\<A#
network.Year__Low 52\sweeks\Range\</A\>\s+(.*?)\s-#
network.Year__High 52\sweek\sRange\</A\>\s+.*\s(.*?)\n#
network.PE__Ratio P/E\sRatio\</A\>\s+(.*?)\s+\<A#
network.Latest__Div Latest\sDiv.\</A\>\s+(.*?)\s*\<A#
network.Annual__Div Annual\sDiv.\</A\>\s+(.*?)\s*\</PRE\>#
```

*Id.*, col. 12, Table 2 (detail).

66. Madnick explains that “Table 2 above uses various PERL operators in order to specify the commands which must be issued.” *Id.*, 13:1-2. Madnick describes perl's syntax and notes that “[t]he meaning of PERL regular expressions are well known in the art.” *Id.*, 13:2-7; *see also id.*, 15:3-10 (illustrating another example of perl's regular expression syntax). Madnick explains that “[i]t is preferred that the language used have good text-handling capabilities,” with “PERL” being an expressly preferred language. *Id.*, 16:23-25.

#### **D. Groner (EX1007)**

67. I understand that Groner is prior art to the '796 patent at least because it issued from an application filed on March 16, 2000. EX1007.

68. Groner states that it “relates generally to speech recognition systems as applied to voice and electronic message mailing, and particularly to a system

and method for converting speech to a text message suitable for sending as an e-mail message and for viewing on a text display device.” *Id.*, 1:6-10. Groner notes that “[v]oice mail messages ... are convenient because telephones are ubiquitous and inexpensive” but have disadvantages compared to written messages, which “can be searched easily for content whereas voice messages cannot be as easily searched.” *Id.*, 2:17-20, 2:31-32.

69. To solve this problem, Groner discloses a system and method whereby “[a]n audio message from a caller for a recipient is received[,] [a]n e-mail address for the recipient is determined[,] [a] text message file is generated from the audio message from the caller[,] [and] [t]he text message file is sent to the recipient at the recipient’s e-mail address.” *Id.*, 2:63-67; *see also id.*, 4:38-42, Fig. 2.

Groner explains: “Because the voice messages are in text form, the voice messages can be searched for particular content.” *Id.*, 3:37-39.

70. Groner discloses “a computer system 70 [that] implements the voice-to-electronic mail system 30,” which “automatically converts a spoken message to a text message which is e-mailed to a recipient.” *Id.*, 5:5-9. Among other components, the system includes “speech recognition procedure 116 that receives audio speech, identifies the audio speech and generates a text file 118 corresponding to the identified audio speech.” *Id.*, 6:6-8. Groner notes that “[t]he

accuracy of speech recognition software has improved” and provides examples of commercial embodiments. *Id.*, 2:38-49.

## **VII. Opinions Regarding Obviousness**

71. In my opinion, the challenged claims are unpatentable based upon at least the following three grounds. First, claims 1-10, 12-21, 23, 25 would have been obvious over Sheth. Second, claims 1-10, 12-21, 23, and 25 would have been obvious over Richardson and Madnick. Third, claims 11, 22, and 24 would have been obvious over Richardson, Madnick, and Groner.

### **A. Ground 1: Claims 1-10, 12-21, 23, and 25 Would Have Been Obvious Over Sheth (EX1004)**

#### **1. Independent Claims 1, 12, 23, and 25**

72. In my opinion, independent claims 1, 12, 23, and 25 would have been obvious to a POSITA over Sheth for at least the reasons set forth below.

#### **a. Claim 1**

##### **i. “A method of automatically processing an input sequence of data symbols, the method comprising the steps of”**

73. To the extent the preamble is a limitation, it is my opinion that Sheth discloses it. Sheth describes “a method for creating a database of metadata (metabase) of a variety of digital media content, data sets, including TV and radio content potentially delivered on Internet.” EX1004, 4:54-57. Specifically, Sheth discloses an “extraction process during which metadata for digital media is

discovered, copied from a Web site, enhanced, and inserted into [a] database.” *Id.*, 10:7-10. In Sheth, “[c]rawling and extraction rules are input to WebCrawler and Extractor programs that traverse Web sites and retrieve digital media metadata from selected pages,” such that the “extractor program takes HTML pages and extraction rules as input and generates XML assets.” *Id.*, 10:40-45.

74. Sheth discloses that “[w]hen the WebCrawler determines that it has found an extractable page, it sends the contents of that page ... to another remote piece of software (an Extractor) that retrieves the valuable metadata from the page.” *Id.*, 11:7-9. The processed webpages contain extractable portions of text. *See, e.g., id.*, 11:17-25. A POSITA would have understood that Sheth’s extractable webpages are an embodiment of an input “sequence of symbols” within the meaning of the ’796 patent, which includes “a data stream representative of text.” EX1001, 1:58-89. Thus, Sheth discloses a method of processing an input sequence of data symbols—i.e., processing text in webpages.

75. Moreover, Sheth repeatedly makes clear that its data-processing method is automatic. EX1004, 5:5-7 (describing “a distributed method and apparatus to quickly produce agents which automatically create and manage digital media metadata”), 9:43-45 (extraction scheduled “automatically”), 13:30-31 (“most of the metadata acquisition happens automatically”).

76. Thus, Sheth discloses a method of automatically processing an input sequence of data symbols.

- ii. **“identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression”**

77. It is also my opinion that Sheth discloses, teaches, or suggests the first clause in the body of claim 1. As Sheth explains: “Once an extractor receives a page, it looks up a set of extraction rules 8 associated with the site from which the page came,” which “list the metadata attributes for the type of media that this site contains as well as rules that describe where to find values for these attributes within the page.” EX1004, 11:41-46. “The extractor scans the Web page content for pieces of text that match the pattern specified by the extraction rules.” *Id.*, 11:50-52. Sheth explains that “[t]he ability to retrieve text that matches a particular pattern is provided by the PERL regular expression language.” *Id.*, 11:64-66.

78. In my opinion, a POSITA would have understood that, by “scan[ning] ... for pieces of text that match the pattern specified by the extraction rules,” which are defined by perl regular expressions (*id.*, 11:50-52, 11:64-66), Sheth’s extractors meet claim 1’s step of “identifying at least one regularly identifiable expression

[i.e., the ‘pattern’ specified by the ‘extraction rules’] in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression” (i.e., perl).

**iii. “identifying at least a portion of information associated with the at least one regularly identifiable expression; and”**

79. For similar reasons to those discussed above, it is my opinion that Sheth discloses, teaches, or suggests the second clause in the body of claim 1. Sheth’s “[e]xtractors 7 are programs that are designed to find information about digital media from a Web page.” EX1004, 11:11-12. For example, extraction “rules can be written that reliably retrieve small sections of text from those pages.” *Id.*, 11:18-20. A “typical extraction rule .... designates the pattern for which the extractor should search to locate the value for [a particular] attribute.” *Id.*, 12:30-32. Again, “[t]he ability to retrieve text that matches a particular pattern is provided by the PERL regular expression language,” which allow Sheth’s extractors to “pick out relevant information from a Web page.” *Id.*, 11:64-66, 11:60-63.

80. In my opinion, a POSITA would have understood that, by “locat[ing] the value” for an attribute using a matchable pattern (*id.*, 12:30-32) and “retriev[ing] small sections of text” (*id.*, 11:18-20), Sheth’s extractors meet claim

1's step of "identifying at least a portion of information associated with the at least one regularly identifiable expression"—i.e., the "value" or "small sections of text" that are identified when a regular expression is matched.

**iv. "extracting the portion of information."**

81. For similar reasons to those discussed above, it is also my opinion that Sheth discloses, teaches, or suggests the final clause in the body of claim 1. Again, Sheth's "extractor scans the Web page content for pieces of text that match the pattern specified by the extraction rules." EX1004, 11:50-52. "After the extractor has attempted to find every attribute in its list of extraction rules, it creates an XML document containing attribute-value pairs" (i.e., database attributes matched to the extracted portions of information), which "is sent to the Metabase Agent 10." *Id.*, 11:52-55, Fig. 6; *see also id.*, 10:43-49 ("An extractor program takes HTML pages and extraction rules as input and generates XML assets" that "are sent to a Metabase Agent that is in charge of enhancing and inserting them into a database of records.").

82. In my opinion, a POSITA would have understood that, by "retriev[ing] small sections of text" that match predefined patterns (i.e., regular expressions) and creating "an XML document" with the desired portions of information (*id.*, 11:18-20, 11:52-55), Sheth meets claim 1's step of "extracting the

portion of information”—i.e., the information associated with a regular expression, which is stored in an XML file and database.

83. Thus, it is my opinion that Sheth teaches or suggests all limitations of claim 1 and, accordingly, would have rendered claim 1 obvious to a POSITA.

**b. Claim 12**

84. Claim 12 recites:

12. Apparatus for automatically processing an input sequence of data symbols, the apparatus comprising:  
at least one processor operative to: (i) identify at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression; (ii) identify at least a portion of information associated with the at least one regularly identifiable expression; and (iii) extract the portion of information; and  
memory, operatively coupled to the at least one processor, for storing at least a portion of results associated with the identifying and extracting operations.

85. In my opinion, Sheth renders claim 12 obvious for substantially the same reasons discussed above for claim 1. The only differences between claims 1 and 12 are that claim 12 recites (i) an apparatus instead of a method; (ii) “at least

one processor operative to” perform the steps of claim 1; and (iii) a final memory limitation. In my opinion, Sheth discloses each of these limitations.

86. First, Sheth is directed to both “*a system* and a method for creating a database of metadata (metabase) of a variety of digital media content,” as well as a “method *and apparatus* to quickly produce agents which automatically create and manage digital media metadata.” EX1004, 4:54-57, 5:5-7 (emphasis added); *see also id.*, 8:11-12 (referring to “the apparatus by which metadata is acquired”), 15:51-52 (“The methods and apparatus of the present invention....”), 17:55-65, 19:60-20:12 (reciting invention as system). Thus, Sheth renders obvious the invention discussed in claim 1 in the form of an apparatus.

87. Second, Sheth discloses a processor operative to perform the claimed functions. Sheth explains that “the present invention is directed to a system for generating metadata associated with a data set, the system comprising: *a processor*,” which among other functions “is programmed to ... extract metadata associated with the data set based upon the retrieved extraction information; and storing in the data store the extracted metadata.” EX1004, 17:55-65 (emphasis added). Thus, in my opinion, Sheth discloses a processor that is operative to perform the recited functions.

88. Third, Sheth discloses claim 12’s memory limitation. In addition to a processor, Sheth’s system comprises “a data store coupled to the processor,” which

is used to “store the extracted metadata.” EX1004, 17:57-64. The ’796 patent admits that “memory” broadly encompasses various data stores, including “any memory devices associated with a processor or CPU, such as, for example, RAM, ROM, a fixed memory device (e.g., hard drive), a removable memory device (e.g., diskette), flash memory, etc.” EX1001, 12:35-39. Accordingly, in my opinion, Sheth discloses memory, operatively coupled to the at least one processor, for storing at least a portion of results associated with the identifying and extracting operations.

89. Thus, in my opinion, claim 12 would have been obvious to a POSITA over Sheth.

**c. Claim 23**

90. Claim 23 recites:

23. An article of manufacture for use in automatically processing an input sequence of data symbols, comprising a machine readable medium containing one or more programs which when executed implement the steps of:

identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression;  
identifying at least a portion of information associated with the at least one regularly identifiable expression; and

extracting the portion of information.

91. In my opinion, Sheth renders claim 23 obvious for substantially the same reasons as claims 1 and 12. The only difference between claims 1 and 23 is the preamble: Instead of a method, claim 23 is directed to an article of manufacture comprising a machine-readable medium containing one or more programs which when executed implement the same steps as claim 1. To the extent this preamble is limiting, Sheth discloses it.

92. Sheth states that it “is directed to *software*, a system and a method for creating a database of metadata (metabase)” for Internet media content. EX1004, 4:54-57 (emphasis added); *see also id.*, 6:6-7 (discussing “a software architecture according to the present invention”), 10:57-58, 11:8-9 (describing both WebCrawler 3 and Extractors 7 as “a piece of software”), 17:21-22 (referring to them as “software components”). Moreover, as discussed for claim 12, Sheth discloses a “system” comprising “a processor, and a data store coupled to the processor and storing WorldModel-based extraction information,” wherein “[t]he processor is *programmed* ... to extract metadata” and for “storing ... the extracted metadata,” among other claimed functions. *Id.*, 17:55-65 (emphasis added).

93. Thus, in my opinion, Sheth discloses an article of manufacture comprising a machine-readable medium containing one or more programs which when executed implement the claimed steps, rendering claim 23 obvious.

**d. Claim 25**

94. Claim 25 recites:

25. A method of automatically processing an input document, the method comprising the steps of:

identifying one or more regular expressions in the input document by comparing the input document to one or more previously-stored regular expressions to determine if there is a match between a portion of the input document and at least one of the previously-stored regular expressions, wherein a regular expression represents a pattern that is matchable in accordance with a programming language that supports such a regular expression; and  
identifying at least a portion of information associated with the at least one regular expression for extraction.

95. In my opinion, Sheth renders claim 25 obvious for substantially the same reasons as claim 1. The only difference between claims 1 and 25 is that claim 25 omits the final step in claim 1 that requires extracting information and, instead, specifies that regular expressions are identified by comparing the input document to one or more previously-stored regular expressions to determine if there is a match between a portion of the input document and at least one of the previously-stored regular expressions. Sheth discloses this additional element.

96. Sheth explains that “[o]nce an extractor receives a page, it looks up a *set of extraction rules* 8 associated with the site from which the page came” and “scans the Web page content for *pieces of text that match the pattern specified by the extraction rules.*” EX1004, 11:41-52 (emphasis added). “The ability to retrieve text that *matches a particular pattern* is provided by the PERL regular expression language.” *Id.*, 11:64-66 (emphasis added).

97. In my opinion, a POSITA would have understood that Sheth’s extraction rules comprising regular expressions are previously stored. Sheth discloses that, before the automatic extraction process begins, “a human writes a set of extraction rules for a site.” EX1004, 11:35-36. The extractor then “looks up [the] set of extraction rules” for a particular website and uses the retrieved “list of extraction rules” to find desired attributes. *Id.*, 11:41-43, 11:52-53. Similarly, Sheth explains that “extraction rules are written” first and then these “extraction rules are input to ... Extractor programs.” *Id.*, 10:40-41; *see also id.*, 10:43-44 (“An extractor program takes ... extraction rules as input”). Accordingly, a POSITA would have understood that Sheth’s extraction rules comprising regular expressions are stored (in a “list of extraction rules” that the extractors “look[] up”) prior to identifying the regular expressions in the input document webpage.

98. Thus, in my opinion, claim 25 would have been obvious to a POSITA over Sheth.

## **2. Dependent Claims 2-10 and 13-21**

99. It is also my opinion that the following dependent claims of the '796 patent would have been obvious to a POSITA over Sheth.

### **a. Claims 2 and 13**

100. Claims 2 and 13 depend from claims 1 and 12, respectively, and further recite that identifying the regularly identifiable expression “*comprises comparing the input sequence of data symbols to one or more previously-stored regularly identifiable expressions to determine if there is a match between a portion of the input sequence and at least one of the previously-stored regularly identifiable expressions.*”

101. This limitation is substantively identical to the limitation added by claim 25 and, for the same reasons discussed in the preceding section, is disclosed by Sheth. *Supra* §VII.A.1.d. Thus, in my opinion, claims 2 and 13 would have been obvious to a POSITA over Sheth.

### **b. Claims 3 and 14**

102. Claims 3 and 14 depend from claims 1 and 12, respectively, and further recite “*normaliz[ing] the input sequence of data symbols prior to identifying the regularly identifiable expression.*”

103. In my opinion, Sheth discloses, teaches, or suggests this limitation for at least two reasons. First, Sheth’s “process starts by identifying a source containing digital media data sets” and “determin[ing] whether a source is actually

worth extracting.” EX1004, 10:23-25. To do so, Sheth uses “WebCrawler 3,” whose purpose “is to read in the content of ‘extractable’ Web pages from a site and pass this content on to the Extractors.” *Id.*, 10:55-57. WebCrawler 3 “begins reading pages from a particular site and determines which of these pages are extractable.” *Id.*, 10:57-60. WebCrawler 3 invokes “crawling rules” that “define the characteristics of an extractable page.” *Id.*, 10:60-65. “When the WebCrawler determines that it has found an extractable page, it sends the contents of that page 6 on to another remote piece of software (an Extractor) that retrieves the valuable metadata from the page.” *Id.*, 11:7-10.

104. Sheth’s process of using a WebCrawler to sort between extractable and unextractable webpages and passing only the extractable webpages to the Extractors is a process of normalizing data. The ’796 patent explains that one example of normalization “may include standardizing any characteristics of the input that are irrelevant to, or a hindrance to, information extraction.” EX1001, 4:12-15. Similarly, Sheth standardizes input data from webpages by excluding pages that are unextractable and thus a hindrance to information extraction. Because this normalization takes place before webpages are sent to the Extractors, a POSITA would have understood it occurs prior to identifying the regularly identifiable expression, satisfying claims 3 and 14.

105. Second, Sheth discloses an additional type of normalization by “convert[ing] the format of an extracted value into a standard format”—e.g., “convert[ing] a clip length in minutes:seconds to seconds (the format accepted by the metabase).” EX1004, 12:50-54. Although this normalization occurs after identifying regular expressions in Sheth, it would have been equally obvious to perform it before identifying any regular expressions. Specifically, a POSITA would have recognized that normalizing data to a standard format before identifying regular expressions would simplify the process of writing extraction rules, which Sheth teaches should be designed to “consistently retrieve the correct text” across “several different pages.” *Id.*, 12:18-22. Because webpages might recite the same attribute in different formats (e.g., defining a runtime in either minutes:seconds or seconds), normalizing the webpage data to a uniform format would have made it easier for a POSITA to write a single regular expression that consistently retrieves the correct information regardless of format.

106. At a minimum, in my opinion, practicing Sheth’s step of standardizing data formats before identifying regular expressions would have been obvious to try. I have been informed that an invention would have been obvious to try when there was a design need or market pressure to solve a problem and there were a finite number of identified, predictable solutions. Here, the problem to be solved is standardizing data for entry in a database, and there were only two possible

solutions for when to practice Sheth’s normalization step—i.e., before or after identifying regular expressions. A POSITA implementing Sheth thus would have faced a simple design choice as to whether to standardize data before or after identifying regular expressions, rendering either option obvious to try.

107. Thus, in my opinion, claims 3 and 14 would have been obvious to a POSITA over Sheth.

**c. Claims 4 and 15**

108. Claims 4 and 15 depend from claims 1 and 12, respectively, and further recite “*identify[ing] one or more classes of data symbols in the input sequence prior to identifying the regularly identifiable expression.*”

109. In my opinion, Sheth discloses this limitation for substantially the same reasons discussed above for claims 3 and 14. *Supra* §VII.A.2.b. As discussed, before Sheth’s Extractors identify any regular expressions, WebCrawler 3 “begins reading pages from a particular site and determines which of these pages are extractable.” EX1004, 10:57-60. Only pages deemed extractable by the WebCrawler are passed to the Extractors. *Id.*, 11:7-10. Sheth explains that “[a]n ‘extractable page’ ... follows a template (however loosely) and likely contains a link to a media asset.” *Id.*, 11:1-3. Moreover, “[a] crawler can often recognize an extractable page by simply examining the URL. If the URL of the page being

examined follows a certain pattern, there is a good chance that the page contains a link to an audio or video asset.” *Id.*, 11:3-7.

110. Accordingly, Sheth’s WebCrawler identifies one or more classes of data symbols in the input sequence (e.g., text in the URL and/or other characteristic text of extractable pages), which occurs “prior to identifying the regularly identifiable expressions.” Thus, in my opinion, claims 4 and 15 would have been obvious to a POSITA over Sheth.

**d. Claims 5 and 16**

111. Claims 5 and 16 depend from claims 1 and 12, respectively, and further recite: “*wherein the at least one regularly identifiable expression comprises a characteristic phrase that typically precedes a particular portion of information.*”

112. In my opinion, Sheth discloses, teaches, or suggests this limitation. For example, Sheth provides examples of extraction rules in Figure 8 below, which use the perl regular expression language. EX1004, 11:60-66.

```

# Name : CNNUS
#
producer          [c]CNN
description        [ ](?i)<img src[^>]*alt="Video"[^>]*>([<]*)<
descriptionReplace1 s/\\([^\s])*\s\\//
parentURL          [u](.*)
invalidated        [c]0
surrogate          [ ](?i)([<]*)</h.>
productionSource   [c]tv
location           [S](?i)/date.*?<p>(.?)\\(.*?\\)
locationReplace1   s/. *?<I>.*?<\\ /I>(.) /$1/
locationReplace2   s/(.*)--.*?/$1/
locationReplace3   s/. *?<em>.*?<\\ /em>(.) /$1/
mediaType          [c]video
source             [c]http://www.cnn.com

```

*Id.*, Fig. 8.

113. A POSITA would have understood that Sheth’s Figure 8 includes multiple regular expressions wherein a characteristic phrase typically precedes a portion of information. For example, the attribute “surrogate” shown above includes the string “<img src=”, wherein the HTML keywords “img” and “src” form a phrase that typically precedes a portion of an extractable URL for an image. This reflects commonly understood syntax for regular expressions, with which a POSITA would have been familiar.

114. Thus, in my opinion, claims 5 and 16 would have been obvious to a POSITA over Sheth.

**e. Claims 6 and 17**

115. Claims 6 and 17 depend from claims 1 and 12, respectively, and further recite: “*wherein the at least one regularly identifiable expression comprises a characteristic phrase that typically follows a particular portion of information.*”

116. In my opinion, Sheth discloses, teaches, or suggests this limitation. In Figure 8 of Sheth (reproduced above in ¶112), the regular expressions for each of the “urlReplace[1-5]” attributes includes a phrase that typically follows a particular portion of information. For example, in “urlReplace5,” the regular expression includes the string “/.\*?qtref”, where “qtref” follows a portion of the URL to be retrieved. *Id.*; EX1004, Fig. 8. Again, this reflects commonly understood syntax for regular expressions, with which a POSITA would have been familiar.

117. Thus, in my opinion, claims 6 and 17 would have been obvious to a POSITA over Sheth.

**f. Claims 7 and 18**

118. Claims 7 and 18 depend from claims 1 and 12, respectively, and further recite: “*wherein the extracted portion of information is used to take a specified action.*”

119. In my opinion, Sheth discloses this limitation. Sheth’s “extractor program takes HTML pages and extraction rules as input and generates XML assets,” which “are sent to a Metabase Agent that is in charge of enhancing and

inserting them into a database of records.” EX1004, 10:43-49; *see also id.*, 11:52-55. Sheth’s use of extracted information to generate XML pages, enhance data assets, and their insertion into a database are all examples wherein the extracted portion of information is used to take specified actions.

120. Thus, in my opinion, claims 7 and 18 would have been obvious to a POSITA over Sheth.

**g. Claims 8 and 19**

121. Claims 8 and 19 depend from claims 1 and 12, respectively, and further recite: “*wherein the extracted portion of information is at least one of visually and audibly presented to the user.*”

122. In my opinion, Sheth discloses this limitation. Sheth is expressly directed to “extracting metadata associated with [a] data set ... and accessing content information based on the metadata *for presentation to a user.*” EX1004, 17:47-54 (emphasis added); *see also id.*, 19:28 (claiming “presentation to a user”).

123. For example, Sheth states that its invention “allow[s] users to search the metabase and make use of the semantic information gained in performing” the extraction steps discussed above, such that users obtain “meaningful results.” *Id.*, 13:47-50, 13:56-60. Sheth also discloses interface “skins” that present information to users by topic. *Id.*, 15:12-30. For example, after collecting URLs using the extraction process discussed in prior steps (*supra* §VII.A.1.a), Sheth’s invention

may “*present the user* with links to local Internet radio stations grouped by genre.” EX1004, 15:18-20 (emphasis added). Sheth further discloses that this information is visually presented to the user, as shown by the visual interfaces depicted in Figures 13A-C. *Id.*, 15:12-30, Figs. 13A-C.

124. Thus, in my opinion, claims 8 and 19 would have been obvious to a POSITA over Sheth.

**h. Claims 9 and 20**

125. Claims 9 and 20 depend from claims 1 and 12, respectively, and further recite: “*wherein the regularly identifiable expression identifying [step/operation] is performed in accordance with one or more programs written in one of the flex, lex, and perl programming language.*”

126. In my opinion, Sheth discloses this limitation. Sheth teaches that “[t]he ability to retrieve text that matches a particular pattern is provided by the PERL regular expression language.” EX1004, 11:64-66; *see also id.*, 6:45-47.

127. Thus, in my opinion, claims 9 and 20 would have been obvious to a POSITA over Sheth.

**i. Claims 10 and 21**

128. Claims 10 and 21 depend from claims 1 and 12, respectively, and further recite: “*wherein the input sequence of data symbols is representative of at least one of text data, transcribed spoken data, deoxyribonucleic acid sequence*

*data, ribonucleic acid sequence data, amino-acid sequence data, audio sequence data, and video sequence data.”*

129. In my opinion, Sheth discloses this limitation. As discussed above, Sheth processes and extracts data from webpages that comprise text. EX1004, 11:50-52 (“The extractor scans the Web page content for pieces of text that match the pattern specified by the extraction rules.”); *see also id.*, 11:64-66. Accordingly, in my opinion, the input sequence of data symbols in Sheth is representative of text data, rendering claims 10 and 21 obvious to a POSITA over Sheth.

**B. Ground 2: Claims 1-10, 12-21, 23, and 25 Would Have Been Obvious Over Richardson (EX1005) and Madnick (EX1006)**

130. As an independent basis for my conclusion that claims 1-10, 12-21, 23, and 25 of the ’796 patent are unpatentable as obvious, it is also my opinion that these claims would have been obvious to a POSITA in view of Richardson and Madnick. Below, I first explain my opinion on why a POSITA would have combined Richardson and Madnick. I then explain why the Richardson-Madnick combination meets all limitations of claims 1-10, 12-21, 23, and 25.

**1. Richardson-Madnick Combination**

131. A POSITA would have recognized that Richardson and Madnick are similar in that both teach the use of regular expressions for identifying and extracting information from documents. *Supra* §§VI.B-C.; EX1005, 9:18-20 (“Regular expression list 104 is used to isolate and identify character strings in the

message which match the expressions in the list.”), Figs. 3-4; EX1006 13:24-26 (“[D]ata retriever 620 then extracts the data requested by the data receiver 102 from the semi structured data source 104 using the specification file.”), col. 12, Tbl. 2 (listing regular expressions).

132. A POSITA also would have recognized that Richardson’s process for identifying and extracting information from documents using regular expressions is carried out using a programming language running on server 16, which is “programmed according to [its] use” and stores “programs used to implement the programming.” EX1005, 5:11-13, 5:17-19; *see also id.*, 5:4-6 (server 16 comprises knowledge base 36, message processing module 38), 5:23-28 (knowledge base 36 comprises extraction rule set 50; message processing module 38 comprises natural language extraction (NLE) module 52), 5:29-33 (NLE module 52 comprises regular expression module 60; extraction rule set 50 comprises regular expression lexicon 70), Figs. 1-5.

133. Although Richardson teaches a programmed system and method using regular expressions, Richardson does not explicitly identify which programming language is used to define such regular expressions. In my opinion, this would have led a POSITA to search for and locate Madnick, which expressly discloses that regular expressions are written in perl. Specifically, Madnick discloses that the exemplary regular expressions in Table 2 are “PERL operators” (EX1006,

13:1-2) and recognizes that “PERL regular expressions are well known in the art” (*id.*, 13:2-7). Moreover, Madnick encourages using perl, which is highlighted as a “preferred” language due to its “good text-handling capabilities.” *Id.*, 16:23-25.

134. As shown by Madnick, a POSITA would have understood that Richardson’s regular expressions represent patterns that are matchable in accordance with a programming language that supports such a regularly identifiable expression, e.g., perl. Further, in my opinion, Madnick would have motivated a POSITA, with a reasonable expectation of success, to use perl for Richardson’s regular expressions by (i) teaching its use for the same purpose of identifying information in documents (EX1006, 13:1-7); (ii) providing examples of perl regular expressions and their syntax (*id.*, col. 12, Tbl. 2; *id.*, 13:64-67, 15:1:10); and (iii) teaching a preference for perl because of its text-handling abilities (*id.*, 16:20-25). Using perl as taught by Madnick requires no modification of Richardson and is consistent with Richardson’s objective of extracting textual information using regular expressions. Moreover, the ’796 patent admits that perl has “built-in support for regular expressions” and has been known in the art since at least 1996. EX1001, 2:26-27, 3:23-31. Thus, a POSITA would have been motivated to combine Richardson and Madnick for this limited purpose with a reasonable expectation of success.

## 2. Independent Claims 1, 12, 23, and 25

135. As shown below, it is my opinion that Richardson and Madnick meet all elements of independent claims 1, 12, 23, and 25 of the '796 patent.

### a. Claim 1

#### i. **“A method of automatically processing an input sequence of data symbols, the method comprising the steps of”**

136. To the extent the preamble of claim 1 is a limitation, Richardson discloses it. The '796 patent acknowledges that “symbols” include “letters” and “characters,” including in “a data stream representative of text.” EX1001, 1:58-59.

137. In my opinion, Richardson discloses a method of processing such text symbols. Specifically, Richardson “is directed to a method and apparatus for automatically recognizing and responding to intent in message text.” EX1005, 1:21-26. “The system recognizes phrase types and intents from input text and maps these to domain knowledge pertaining to common kinds of intents.” *Id.*, 4:15-18. “The input message arrives at the system via electronic mail, network transport, or any number of well-known communication methods.” *Id.*, 4:23-25. “FIGS. 5(a)-(b) show an example of an input message 102 being processed....” *Id.*, 8:42-43. “The input message is processed by NLE module 52 to generate tags associated with portions of the input message.” *Id.*, 8:43-44.

138. Thus, in my opinion, Richardson discloses a method of automatically processing an input sequence of data symbols.

- ii. **“identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression”**

139. In my opinion, Richardson and Madnick teach the first clause in the body of claim 1 of the '796 patent.

140. First, it is my opinion that Richardson's process includes identifying a regularly identifiable expression. In Richardson, “[t]he process of understanding and responding to text messages begins with accepting the input text message, which is first processed by NLE module 52.” EX1005, 5:34-37. Within Richardson's NLE module 52, “regular expression module 60 receives the input text message and reads from regular expression lexicon 70.” *Id.*, 5:37-40, Fig. 4. Richardson explains that “[r]egular expression module 60 looks for regular expressions which are known to regular expression lexicon 70.” *Id.*, 5:51-53. “Once a *regular expression[] is identified*, it is excluded from further analysis in the free-form text.” *Id.*, 5:60-61 (emphasis added). Thus, Richardson discloses identifying at least one “regularly identifiable expression.”

141. Second, in my opinion, Richardson's regular expressions represent patterns that are matchable. For example, “[r]egular expression list 104 [that] is used to isolate and identify character strings in the message *which match the*

*expressions* in the list.” EX1005, 9:17-19 (emphasis added). Moreover, Richardson explains that “after the spelling module removes a typographical error, the result may *match a regular expression* which was not earlier matched....” *Id.*, 6:14-19 (emphasis added). Likewise, Richardson’s claim 13 recites “*matching* identified tokens to a database of *regular expression patterns*.” *Id.*, 12:53-54 (emphasis added). A POSITA would also have understood that Richardson’s regular expressions represent matchable patterns, such as the examples in Table 1 below that match patterns for a URL, phone number, or email address.

TABLE 1

| Regular Expression List |       |
|-------------------------|-------|
| Expression              | Type  |
| http://*{ /* }>         | URL   |
| ###-###-####            | Phone |
| [a-zA-Z]@*, *           | Email |
| ...                     | ...   |

EX1005, 8:59-66. Thus, Richardson teaches identifying “at least one regularly identifiable expression that represents a pattern that is matchable.”

142. Third, in my opinion, a POSITA would have understood that Richardson’s regular expressions represent a pattern matchable in accordance with a programming language that supports such a regularly identifiable expression. Richardson’s method of extracting information from input text by identifying regular expressions is implemented using a programming language. In particular, regular expression matching in Richardson is performed by server 16, which is a

“general purpose computer system[] programmed according to [its] use,” for which “[t]he programs used to implement the programming might reside in memory” or other storage devices. EX1005, 5:11-19. Richardson’s NLE module 52, which comprises regular expression module 60 (EX1005, 5:29-30), is part of processing module 38 (*id.*, 5:23-28), which in turn is part of server 16 (*id.*, 5:4-6). *See also* Fig. 1 (server 16), Fig. 2 (server 16 comprises message processing module 38), Fig. 3 (message processing module 38 comprises NLE module 52), Fig. 4 (NLE module 52 comprises regular expression module 60).

143. Thus, a POSITA would have understood that regular expression module 60, which “looks for regular expressions” and “identifies instances of the regular expressions” (EX1005, 5:51-59), functions in accordance with the programming of server 16, where regular expression module 60 resides.

Accordingly, in my opinion, a POSITA would have understood that server 16 is necessarily programmed with “a programming language that supports such a regularly identifiable expression” because server 16 comprises regular expression module 60.

144. Further, a POSITA would also have understood that Richardson’s regular expressions represent patterns matchable in accordance with “a programming language that supports such a regularly identifiable expression” based on Madnick. Madnick acknowledges that “[t]he meaning of PERL regular

expressions are well known in the art” and recites “various PERL operators” to match regular expressions. EX1006, 13:6-7, 13:1-2, col. 12, Tbl. 2; *see also id.*, 15:1-10 (discussing perl syntax for regular expressions). Thus, to a POSITA reading the reference, Richardson would at least implicitly disclose “a programming language that supports a regularly identifiable expression.”

145. At a minimum, a POSITA practicing Richardson would at least have found it obvious to use a programming language that supports a regularly identifiable expression, such as perl, in view of Madnick. As I discussed above in §VI.C, Madnick encourages a POSITA to use perl for coding regular expressions as a “preferred” language with “good text-handling capabilities.” EX1006, 16:22-24. Again, Madnick repeatedly illustrates the use of perl for coding regular expressions in the same context of extracting textual information as Richardson. EX1006, col. 12, Tbl. 2, 13:1-7, 15:1-10. Thus, in my opinion, a POSITA would have been motivated to combine Madnick’s use of perl with Richardson’s process to match a regular expression in accordance with “a programming language that supports such a regularly identifiable expression.”

146. Moreover, I note that the ’796 patent admits that the claimed regularly identifiable expressions may be “stored in one of several common programming languages, e.g., as a ‘flex’ or ‘perl’ program,” which “have built-in-support for regular expressions.” EX1001, 2:23-27. The ’796 further admits that regular

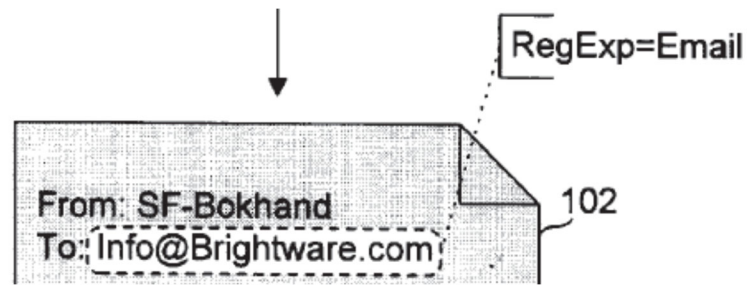
expressions include “any form of a pattern that is matchable in the flex, lex, or perl programming languages,” which were admittedly known in the art. *Id.*, 3:23-31. Accordingly, the ’796 patent itself concedes that regularly identifiable expressions are, by definition, matchable in accordance with “a programming language that supports such a regularly identifiable expression.”

147. Thus, in my opinion, Richardson and Madnick teach the first clause in the body of claim 1 of the ’796 patent.

**iii. “identifying at least a portion of information associated with the at least one regularly identifiable expression; and”**

148. It is also my opinion that Richardson further discloses the second clause in claim 1’s body. Richardson disclose that, “[a]s shown in FIG. 5(a), NLE module 52 uses a regular expression list 104 ... in assigning tags to a message.” EX1005, 8:50-52. Richardson explains that “[r]egular expression list 104 is used to *isolate and identify character strings* in the message *which match the expressions* in the list.” *Id.*, 9:17-19 (emphasis added). “If a regular expression from list 104 is matched in the message, the matching text is tagged with a ‘RegExp’ tag.” *Id.*, 9:19-21.

149. In Table 1, an exemplary regular expression to identify email addresses is “[a-zA-Z]@\*.\*]”. *Id.*, 1:64. In Figure 5(a) excerpted below, “an e-mail address is matched and tagged with a ‘RegExp=Email’ tag.” *Id.*, 9:21-22.



*Id.*, Fig. 5(a) (detail). In this example, the “regularly identifiable expression” within the meaning of claim 1 of the ’796 patent is “[a-zA-Z]@\*.\*” and the portion of information associated with the regular expression is the specific email address “Info@Brightware.com”.

150. Thus, in my opinion, Richardson discloses identifying “at least a portion of information associated with the at least one regularly identifiable expression,” as recited in claim 1 of the ’796 patent.

#### iv. “extracting the portion of information.”

151. It is also my opinion that Richardson further discloses, teaches, or suggests claim 1’s final clause. Notably, the module in Richardson that identifies regular expressions is called the “Natural Language *Extraction* Module,” and the regular expressions themselves are labeled “*Extraction* Rules.” EX1005, Figs. 3-5, 3:43-49; *see also id.*, 5:27-30 (“[A] natural language *extraction* (NLE) module 52” is shown in Fig. 4 “comprising a regular expression module 60.”), 5:32-33 (“*Extraction* rule set 50 is shown comprising a regular expression lexicon 70.”) (emphases added). Thus, a POSITA would have understood that Richardson

expressly characterizes regular expressions as a means for extracting matchable portions of information.

152. Moreover, Richardson discloses specific examples of extracting portions of information identified with regular expressions. As I discussed above, one example is an email address that is identified using a regular expression. *Id.*, Fig. 5(a), 9:21-22. Richardson discloses that the email address may be extracted for use in a database query. *Id.*, 10:24-30 (“Using the tagged phrases and regular expressions, the system performs any necessary database lookups. For example, a database lookup might be used to determine the customer’s full identity just from a name in the message or the e-mail address.”).

153. Richardson also discloses that “[t]he message with its tags”—e.g., portions of information matching regular expressions—“are then provided to intent matching module 54, which identifies the writer’s and system operator’s intents using an intent list 114.” EX1005, 10:35-38. Thus, in Richardson, identified information is extracted for transmission to intent matching module 54.

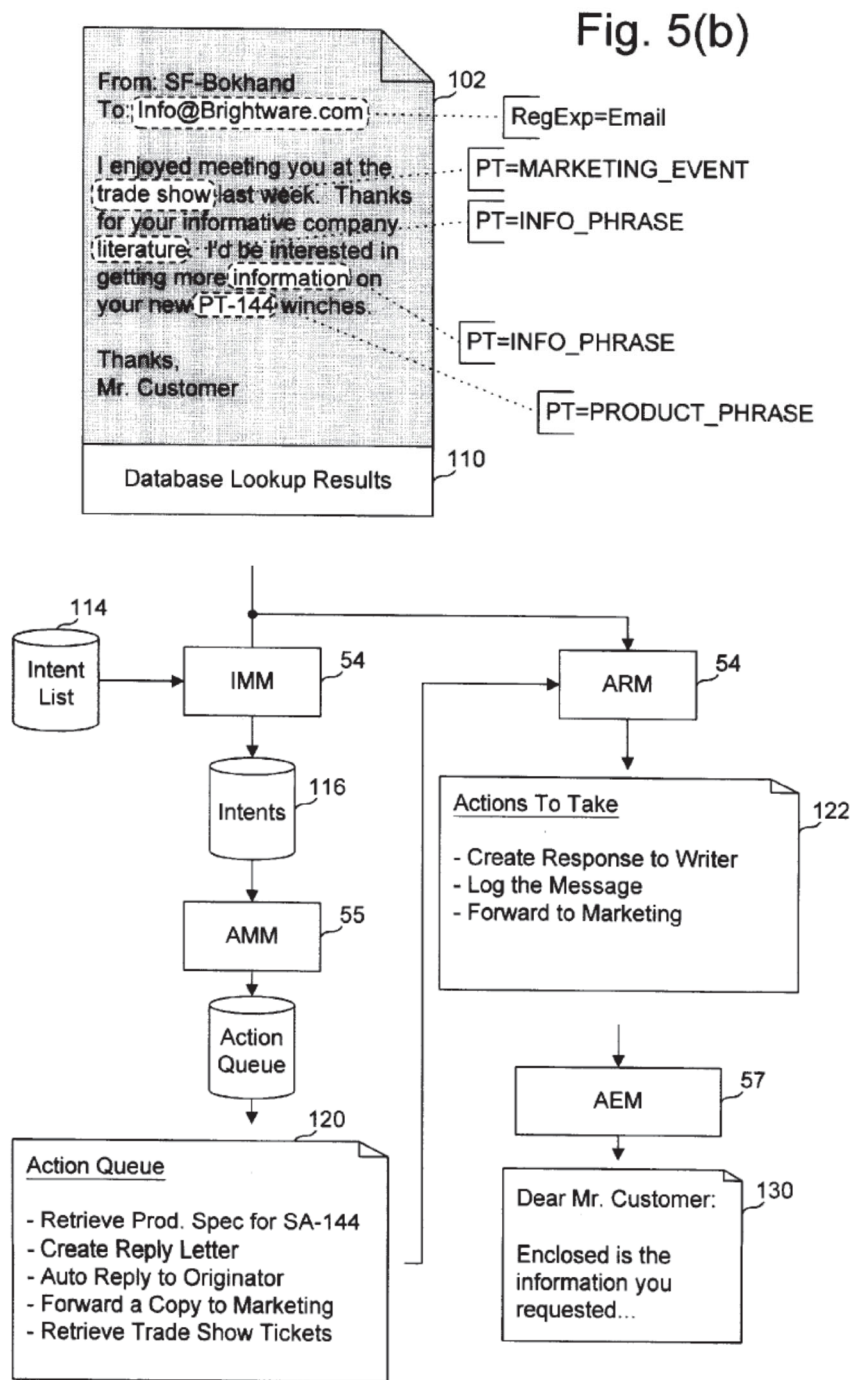
154. In yet another example in Richardson, a reply to the input text is generated that includes extracted information identified by regular expressions:

[A] list 122 of actions to take is passed to action execution module (“AEM”) 57, which performs the actions listed on list 122, such as creating a response 130. *A preferred response 130 includes all or part of the original message, ... and an indication of how the message*

was interpreted. *One way to show how the message was interpreted is to show the tags and the values for tagged regular expressions and tagged phrase types.*

EX1005, 11:17-23 (emphasis added).

155. A POSITA would have understood that each of Richardson's disclosed uses of information extracted by matching regular expressions—i.e., (i) database lookups, (ii) transmission to intent matching module 54, and (iii) preparation of a response comprising tagged regular expressions—is depicted below in Figure 5(b)



EX1005, Fig. 5(b).

156. Relatedly, Richardson discloses that “[i]f one of the actions is a response to the writer, the composition of the response message text may include

... phrases and extracted information,” and “for debugging purposes or legal reasons, the input message text, response results and other processing information might be logged for later analysis, reporting or record keeping.” EX1005, 7:41-58. For this additional reason, a POSITA would have understood that Richardson discloses extracting portions of information identified using regular expressions.

157. Accordingly, in my opinion, Richardson and Madnick teach or suggest all elements of claim 1, which would have been obvious to a POSITA.

**b. Claim 12**

158. Claim 12 recites:

12. Apparatus for automatically processing an input sequence of data symbols, the apparatus comprising:  
at least one processor operative to: (i) identify at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression; (ii) identify at least a portion of information associated with the at least one regularly identifiable expression; and (iii) extract the portion of information; and  
memory, operatively coupled to the at least one processor, for storing at least a portion of results associated with the identifying and extracting operations.

159. In my opinion, Richardson and Madnick render claim 12 obvious for substantially the same reasons as claim 1. The only differences between claims 1 and 12 are that claim 12 recites (i) an apparatus instead of a method; (ii) “at least one processor operative to” perform the claimed steps; and (iii) a final memory limitation. Each of these limitations added by claim 12 would have been obvious to a POSITA in view of Richardson.

160. First, to the extent the preamble of this claim is limiting, it is my opinion that Richardson discloses it. Richardson is “directed to a method *and apparatus* for automatically recognizing and responding to intent in message text” that performs the functions discussed above for claim 1. EX1005, 1:21-23 (emphasis added). Moreover, Richardson discloses an apparatus—“a computer system”—for performing the process discussed above for claim 1. *Id.*, 3:37-39 (“FIG. 1 is a block diagram of a computer system over which messages are sent to a server for automatic handling according to the present invention.”), 3:47-49 (“FIGS. 5(a)-(b) illustrate *the process of using the apparatus* shown in FIGS. 1-3 to automatically respond to a message.”) (emphasis added).

161. Second, a POSITA would have understood that Richardson discloses at least one processor operative to perform the functions of claims 1 and 12. In Richardson, “[c]ustomer system 12 sends input message 14 to server 16 *for processing.*” EX1005, 4:40-41 (emphasis added). Server 16 is a “general purpose

computer system ... comprising one or more *central processing unit*....” *Id.*, 5:11-15 (emphasis added). As discussed, server 16 performs the steps of claim 1 (the same steps as claim 12) within server 16’s NLE module 52 comprising regular expression module 60. *Id.*; EX1005, Figs. 1-3, 5:4-6, 5:23-30. Thus, server 16 comprises at least one processor operative to perform the claimed steps.

162. Third, in my opinion, Richardson teaches or at least suggests claim 12’s memory limitation. The ’796 patent admits that “memory” broadly includes “any memory devices associated with a processor or CPU, such as, for example, RAM, ROM, a fixed memory device (e.g., hard drive), a removable memory device (e.g., diskette), flash memory, etc.” EX1001, 12:35-39. Richardson’s server 16, which performs the steps of claim 12, comprises “memory (ROM and RAM), and associated peripheral equipment, such as disk drives...,” which are all forms of memory according to the ’796 patent. EX1005, 5:11-15. Moreover, Richardson discloses that “[t]he programs used to implement the programming might reside in memory....” EX1005, 5:11-18. Thus, Richardson’s server 16 comprises memory, operatively coupled to the at least one processor (i.e., Richardson’s “one or more central processing unit”). EX1005, 5:11-15.

163. Richardson teaches that server 16’s memory is for storing at least a portion of results associated with the identifying and extracting operations. “In the preferred embodiment, server 16 is ... [an] automated message understanding and

response system.” EX1005, 4:54-56. Specifically, “[s]erver 16 is adapted to take an action based on ... the text of query message,” including “response message 18” (*id.*, 4:44-47), which “may include ... phrases and extracted information” (*id.*, 7:41-44). “A preferred response 130 includes all or part of the original message,” including “tags and the values for tagged regular expressions.” *Id.*, 11:17-23. Moreover, as I discussed above for claim 1, Richardson teaches other examples in which server 16 uses extracted information to query a database (*id.*, 10:24-30), determine a writer’s intent (*id.*, 10:35-38), and log extracted information for recordkeeping (*id.*, 7:41-58). In each of these examples, a POSITA would have understood that server 16 necessarily stores in memory, at least temporarily, at least a portion of results associated with the identifying and extracting operations—i.e., information extracted using regular expressions—for purposes of generating a response, querying a database, determining a writer’s intent, or maintaining a log.

164. At a minimum, a POSITA would have found it obvious to use Richardson’s disclosed memory for storing at least a portion of results associated with the identifying and extracting operations. A POSITA would have been motivated to store at least a portion of the extracted results in server 16’s memory to allow server 16’s processor to utilize the results in performing the operations discussed above. Likewise, a POSITA would have been motivated to store the

extracted results for later retrieval—e.g., to prepare a follow-up response, to repeat or update a database query, and to log results “for later analysis, reporting or record keeping,” as encouraged by Richardson. EX1005, 7:41-58.

165. Thus, in my opinion, claim 12 would have been obvious to a POSITA over Richardson and Madnick.

**c. Claim 23**

166. Claim 23 recites:

23. An article of manufacture for use in automatically processing an input sequence of data symbols, comprising a machine readable medium containing one or more programs which when executed implement the steps of:

identifying at least one regularly identifiable expression in the input sequence of data symbols, wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression;  
identifying at least a portion of information associated with the at least one regularly identifiable expression; and  
extracting the portion of information.

167. In my opinion, Richardson and Madnick render claim 23 obvious for substantially the same reasons as claims 1 and 12. The only difference between claims 1 and 23 is the preamble: Instead of a method, claim 23 recites an article of

manufacture comprising a machine-readable medium containing one or more programs which when executed implement the same steps as claim 1.

168. To the extent the preamble of this claim is limiting, Richardson discloses it. As I discussed above for claim 12, Richardson is “directed to a method *and apparatus* for automatically recognizing and responding to intent in message text” that performs the steps of claim 1. EX1005, 1:21-23 (emphasis added). Moreover, Richardson discloses “a computer system” for performing the claimed process. *Id.*, 3:37-39. Specifically, “[c]ustomer system 12 sends input message 14 to server 16 for processing” (*id.*, 4:40-41), and server 16 is a “general purpose computer system[] *programmed according to [its] use* and comprising one or more central processing unit, memory (ROM and RAM), and associated peripheral equipment, such as disk drives....” (*id.*, 5:11-15) (emphasis added). Richardson further explains that “[t]he programs used to implement the programming might reside in memory, in one of the storage devices, or a combination of the two.” *Id.*, 5:17-19.

169. Thus, a POSITA would have understood that Richardson discloses an article of manufacture (i.e., server 16) for use in automatically processing an input sequence of data symbols, comprising a machine-readable medium containing one or more programs (i.e., server 16’s memory), which when executed implement the recited claim steps—the same steps recited in claims 1 and 12.

170. Similarly, Madnick teaches that its disclosed methods may be performed by “one or more computer-readable programs embodied on or in one or more articles of manufacture,” and “[t]he software programs may be stored on or in one or more articles of manufacture as object code.” EX1006, 16:16-26. A POSITA would have been motivated to store and implement Richardson’s programming on an article of manufacture as taught by Madnick, which also relates to using software to extract information from documents.

171. Thus, in my opinion, claim 23 would have been obvious to a POSITA over Richardson and Madnick.

**d. Claim 25**

172. Claim 25 recites:

25. A method of automatically processing an input document, the method comprising the steps of:

- identifying one or more regular expressions in the input document by comparing the input document to one or more previously-stored regular expressions to determine if there is a match between a portion of the input document and at least one of the previously-stored regular expressions, wherein a regular expression represents a pattern that is matchable in accordance with a programming language that supports such a regular expression; and
- identifying at least a portion of information associated with the at least one regular expression for extraction.

173. In my opinion, Richardson and Madnick render claim 25 obvious for substantially the same reasons as claim 1. The only difference between claims 1 and 25 is that claim 25 omits the final step in claim 1 that requires extracting information and, instead, specifies that regular expressions are identified by comparing the input document to one or more previously-stored regular expressions to determine if there is a match between a portion of the input document and at least one of the previously-stored regular expressions.

174. A POSITA would have understood that this limitation is disclosed by Richardson. First, Richardson's regular expressions are previously stored: the "system isolates the writer's critical words which relate to prototypical statements *stored in the system.*" EX1005, 2:55-57 (emphasis added). Specifically, "[e]xtraction rule set 50 is shown comprising a regular expression lexicon 70," such that "the regular expression module 60 receives the input text message and reads from regular expression lexicon 70." EX1005, 5:32-33, 5:38-40. A POSITA would have understood from Richardson that "regular expression lexicon 70" comprising regular expressions is previously stored in order for "regular expression module 60" to "read[] from" it.

175. Second, Richardson teaches that these previously stored regular expressions are compared to the input document to determine if there is a match. While reading text, "[r]egular expression module 60 looks for regular expressions

which are known to regular expression lexicon 70” according to “rules for identifying regular expressions” and “identifies instances of the regular expressions represented in regular expression lexicon 70.” EX1005, 5:51-60.

176. Specifically, Richardson uses “regular expression list 104,” with an example provided in Table 1. *Id.*, 8:50-66. “Regular expression list 104 is used to isolate and identify character *strings in the message which match the expressions in the list.*” *Id.*, 9:17-19 (emphasis added). This process of comparing and matching previously stored regular expressions is also claimed by Richardson: Claim 1 of Richardson recites “*comparing* the set of standard lexical tokens to phrase expressions in a collection of lexicon entries, wherein each lexicon entry comprises a phrase expression and an associated phrase type,” and claim 4 of Richardson specifies that “the phrase type is selected from ... *a regular expression text pattern.*” *Id.*, 11:56-59, 12:8-10 (emphasis added). Similarly, Richardson’s claim 13 recites “*matching* identified tokens to *a database of regular expression patterns.*” *Id.*, 12:51-54 (emphasis added). Madnick also discloses that when using regular expressions, “[f]or each variable to be retrieved in a given state, the state description contains *a pattern to be matched* against the document....” EX1006, 15:1-3 (emphasis added); *see also id.*, 15:5-10.

177. Thus, a POSITA would have understood that Richardson and Madnick disclose comparing the input document to one or more previously stored

regular expressions to determine if there is a match between a portion of the input document and at least one of the previously stored regular expressions.

178. Accordingly, in my opinion, claim 25 would have been obvious to a POSITA over Richardson and Madnick.

### **3. Dependent Claims 2-10 and 13-21**

179. It is also my opinion that Richardson and Madnick render obvious each of the following dependent claims.

#### **a. Claims 2 and 13**

180. Claims 2 and 13 depend from claims 1 and 12, respectively, and further recite that identifying the regularly identifiable expression “*comprises comparing the input sequence of data symbols to one or more previously-stored regularly identifiable expressions to determine if there is a match between a portion of the input sequence and at least one of the previously-stored regularly identifiable expressions.*”

181. This limitation is substantively identical to the limitation added by claim 25 and, for the same reasons that I discussed in the preceding section, is taught by Richardson and Madnick. *Supra* §VII.B.2.d.

182. Thus, in my opinion, claims 2 and 13 would have been obvious to a POSITA over Richardson and Madnick.

**b. Claims 3 and 14**

183. Claims 3 and 14 depend from claims 1 and 12, respectively, and further recite “*normaliz[ing] the input sequence of data symbols prior to identifying the regularly identifiable expression.*”

184. For context, I note that the ’796 patent provides an example of normalization as it would be understood by a POSITA:

In step 14, the input data is normalized. This process may include standardizing any characteristics of the input that are irrelevant to, or a hindrance to, information extraction. An example of normalization, for text input, is to change all the letters in the text stream to lower-case and to replace multiple blank spaces between words by a single blank space. This makes subsequent pattern matching easier.

EX1001, 4:12-18.

185. Similarly, Richardson discloses that words may be “regularized” and spell-checked prior to identifying regular expressions:

Morphology module 66 uses overall lexicon 72 to transform the words into their regularized form. For example, morphology module 66 might convert the word “said” to “say” in an effort to simplify the message. As should be apparent, modules 60, 62, 64, and 66 can operate on the input message non-sequentially. For example, after the spelling module removes a typographical error, the result may match a regular expression which was not earlier matched, in which case

regular expression module 60 would add additional regular expression tags when the message is revisited.

EX1005, 6:9-19; *see also id.*, 9:38-41. These actions, which as described above may occur before regular expressions are identified, comprise normalization and are similar to the '796 patent's embodiment of changing letters to "make[] subsequent pattern matching easier." EX1001, 4:15-19.

186. Thus, in my opinion, claims 3 and 14 would have been obvious to a POSITA over Richardson and Madnick.

**c. Claims 4 and 15**

187. Claims 4 and 15 depend from claims 1 and 12, respectively, and further recite "*identify[ing] one or more classes of data symbols in the input sequence prior to identifying the regularly identifiable expression.*"

188. In my opinion, Richardson discloses this limitation. Specifically, Richardson discloses pre-processing input text to classify data as structured or unstructured:

Prior to the regular expression parse, the message might be pre-processed for structured data. For example, if attachments are included or data is included in the message in a structured format, such as 'HTML' forms data, comma delimited data records, or an attached spreadsheet, that structured data is identified. Preferably, that structured data is flagged such that NLE module 52 does not need to process it.

EX1005, 9:29-36; *see also id.*, 5:40-46 (“the input text is scanned for these data constructs which are not free-form text”), 12:44-50 (claiming step of “[s]eparating free form text in the content from structured text in the content”).

189. A POSITA would have understood that Richardson’s step for identifying and separating structured data (EX1005, 9:29-36, 12:44-50) comprises identifying classes of data symbols—i.e., structured or unstructured data—and occurs prior to identifying regular expressions. Likewise, a POSITA would have understood that Richardson’s steps of regularizing words and spell-checking discussed above for claims 3 and 14 (EX1005, 6:9-19, 9:38-41) also comprise identifying classes of data symbols—i.e., words that may be regularized/simplified or are misspelled—and occur before identifying regular expressions.

190. Thus, in my opinion, claims 4 and 15 would have been obvious to a POSITA over Richardson and Madnick.

#### **d. Claims 5 and 16**

191. Claims 5 and 16 depend from claims 1 and 12, respectively, and further recite: “*wherein the at least one regularly identifiable expression comprises a characteristic phrase that typically precedes a particular portion of information.*”

192. In my opinion, Richardson discloses, teaches, or suggests this limitation. As an example, Richardson discloses regular expression list 104, which

includes the following expression to identify a URL: “http://\*{ /\* }>”. EX1005, 8:62. In this example, the characteristic phrase “http://” typically precedes a particular portion of information—i.e., the rest of the URL—that the regular expression is designed to extract. Again, this reflects standard and well-known syntax for regular expressions that would have been well understood to a POSITA.

193. Thus, in my opinion, claims 5 and 16 would have been obvious to a POSITA over Richardson and Madnick.

**e. Claims 6 and 17**

194. Claims 6 and 17 depend from claims 1 and 12, respectively, and further recite: “*wherein the at least one regularly identifiable expression comprises a characteristic phrase that typically follows a particular portion of information.*”

195. In my opinion, Richardson discloses, teaches, or suggests this limitation. As an example, Richardson discloses regular expression list 104, which includes the following expression to identify an email address: “[a-zA-Z]@\*.\*”. EX1005, 8:64. In this example, the characteristic phrase “@\*.\*” typically follows at least a portion of information—i.e., the user name portion of an email address—that the regular expression is designed to extract. As with the similar syntax discussed for claims 5 and 16, this would have been well understood to a POSITA.

196. Thus, in my opinion, claims 6 and 17 would have been obvious to a POSITA over Richardson and Madnick.

**f. Claims 7 and 18**

197. Claims 7 and 18 depend from claims 1 and 12, respectively, and further recite: “*wherein the extracted portion of information is used to take a specified action.*”

198. In my opinion, Richardson discloses this limitation. Richardson is directed to “[a]n improved message text understanding and response system,” which uses extracted information to determine “prototypical response actions.” EX1005, 2:53-63. “The assembled potential actions are evaluated and executed,” and “[e]xamples of potential actions include responding to the writer, giving recommendations for the next step in processing, ... routing the message to a third party for automatic or manual response, or invoking an external system.” *Id.*, 3:9-15; *see also id.*, Figs. 3, 5(b) (referring to actions based on data extraction and intent mapping), 4:5-7 (“The system determines the customer’s intent, to the extent that it relates to the company’s customer needs, and *takes an action* appropriate for that intent.”), 4:44-47 (“Server 16 is adapted to *take an action* based on an internal knowledge base and the text of query message 14. If the action is a response message 18, that response message 18 is sent to customer system 12.”) (emphases added), 11:62-12:2 (claiming the steps of selecting, generating, and executing actions), 12:37-40 (claiming exemplary actions).

199. Richardson explains that information extracted using regular expressions is used to determine and take the specified action. For example, “[r]egular expression module 60 looks for regular expressions which are known to regular expression lexicon 70” (*id.*, 5:51-53) and “the input message with all the phrase types identified is passed to an intent matching module (‘IMM’) 54” (*id.*, 6:34-36), which determines “typical cases for which appropriate response actions can be established” (*id.*, 6:42-43). Then, “the message and the intents list is passed to an action mapping module (‘AMM’) 55, which generates a list of proposed actions” (*id.*, 7:18-20) that are processed by “an action resolution module (‘ARM’) 56” (*id.*, 7:35-38), and “[t]he actions taken are ... executed by an action execution module (‘AEM’) 57” (*id.*, 7:45-47). *See also id.*, Fig. 5(b).

200. Thus, in my opinion, claims 7 and 18 would have been obvious to a POSITA over Richardson and Madnick.

**g. Claims 8 and 19**

201. Claims 8 and 19 depend from claims 1 and 12, respectively, and further recite: “*wherein the extracted portion of information is at least one of visually and audibly presented to the user.*”

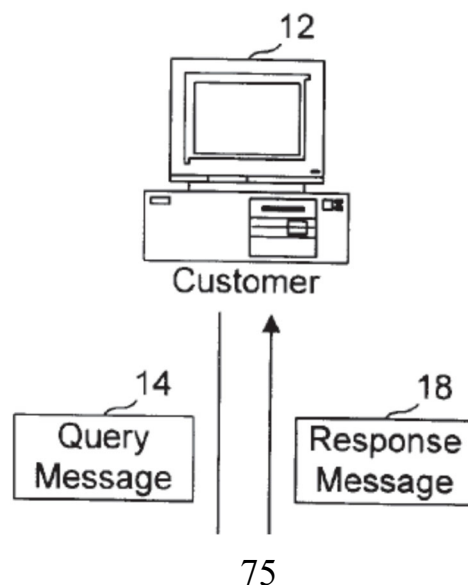
202. In my opinion, Richardson discloses this limitation. As I discussed for the final clause of claim 1 (*supra* §VII.B.2.a.iv), Richardson generates a

response to the user that includes extracted information identified by regular expressions:

[A] list 122 of actions to take is passed to action execution module (“AEM”) 57, which performs the actions listed on list 122, such as creating a response 130. *A preferred response 130 includes all or part of the original message, ... and an indication of how the message was interpreted. One way to show how the message was interpreted is to show the tags and the values for tagged regular expressions and tagged phrase types.*

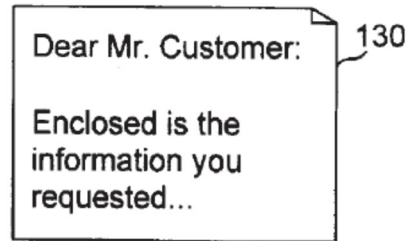
EX1005, 11:17-23 (emphasis added).

203. A POSITA would have understood that this information is visually presented to the user, who receives the response as text displayed on customer system 12, a “general purpose computer system[] ... comprising one or more ... video display terminals.” EX1005, 5:11-17. As shown in Figure 1 of Richardson, partly reproduced below, response message 18 is sent to customer system 12 that comprises such a display (depicted as a CRT monitor):



*Id.*, Fig. 1 (detail).

204. As shown in Figure 5(b), preferred response 130 is a textual message:



*Id.*, Fig. 5(b) (detail). Accordingly, a POSITA would have understood Richardson discloses that the extracted portion of information is visually presented to the user as part of the textual response sent to customer system 12.

205. At a minimum, since Richardson discloses a video display terminal for customer system 12 that receives a response comprising extracted information, a POSITA would have found it obvious to visually present such information to the user on the provided display terminal. *See* EX1005, 5:11-17, 11:17-23, Fig. 1.

206. Thus, in my opinion, claims 8 and 19 would have been obvious to a POSITA over Richardson and Madnick.

#### **h. Claims 9 and 20**

207. Claims 9 and 20 depend from claims 1 and 12, respectively, and further recite: *“wherein the regularly identifiable expression identifying [step/operation] is performed in accordance with one or more programs written in one of the flex, lex, and perl programming language.”*

208. As I discussed above, while Richardson uses programming in server 16 to match regular expressions and extract information (*see* EX1005, 5:11-19), Richardson does not expressly identify the specific programming language used for regular expressions. *Supra* §VII.B.1. Madnick, however, expressly identifies “PERL” as a “preferred” language because of its “good text-handling capabilities.” EX1006, 16:22-25. Madnick recognizes that “[t]he meaning of PERL regular expressions are well known in the art.” *Id.*, 13:6-7. Madnick further explains perl syntax for regular expressions (*id.*, 13:1-6) and more than a dozen examples of regular expressions written in perl (*id.*, col. 12, Tbl. 2). Thus, as shown above in §VII.B.1, a POSITA would have been motivated to use perl as taught by Madnick in Richardson’s process using regular expressions.

209. Moreover, consistent with Madnick’s disclosure, the ’796 patent admits that perl has been known in the art since at least 1996 (EX1001, 3:29-31) and features built-in support for regular expressions (*id.*, 2:26-27).

210. Thus, in my opinion, claims 9 and 20 would have been obvious to a POSITA over Richardson and Madnick.

**i. Claims 10 and 21**

211. Claims 10 and 21 depend from claims 1 and 12, respectively, and further recite: “*wherein the input sequence of data symbols is representative of at least one of text data, transcribed spoken data, deoxyribonucleic acid sequence*

*data, ribonucleic acid sequence data, amino-acid sequence data, audio sequence data, and video sequence data.”*

212. In my opinion, Richardson discloses this limitation. Specifically, Richardson’s input sequence of data symbols is an “input text message.” EX1005, 5:34-40; *see also id.*, Figs. 5(a), 5(b) (showing input text 102). Accordingly, a POSITA would have understood that Richardson’s input sequence of data symbols is representative of text data.

213. Thus, in my opinion, claims 10 and 21 would have been obvious.

**C. Ground 3: Claims 11, 22, and 24 Would Have Been Obvious Over Richardson, Madnick, and Groner**

214. It is further my opinion that the three other claims of the ’796 patent that are not addressed above—claims 11, 22, and 24—would have been obvious to a POSITS over Richardson, Madnick, and Groner. Below, first, I discuss why a POSITA would have further combined the Richardson-Madnick combination I addressed above with Groner. Then, I address how the Richardson-Madnick-Groner combination meets all limitations of these claims.

**1. Richardson-Madnick-Groner Combination**

215. As discussed above, Richardson expressly relates to “automatically recognizing and responding to intent in message text, particularly as used in a large-scale automatic message response system....” EX1005, 1:21-26.

Specifically, Richardson describes a method and system for responding to

“electronic mail” (e.g., *id.*, 3:19, 4:23), including in the context of “a company’s customer service department” (*id.*, 4:2-5). *See also id.*, Figs. 5(a), 5(b) (depicting customer emails). Although Richardson acknowledges the prevalence of “other messaging media, such as telephone calls” (*id.*, 1:32-34), Richardson’s method and system are not expressly described in terms of any messages in forms other than text (e.g., emails). Thus, in my opinion, a POSITA would have been motivated to seek techniques for implementing Richardson’s method and system on these “other messaging media, such as telephone calls,” that were well known. *Id.*

216. This would have led a POSITA to search for and locate Groner. In the same field of endeavor as Richardson, Groner explicitly relates to “speech recognition systems as applied to voice and electronic message mailing, and particularly to a system and method for converting speech to a text message suitable for sending as an e-mail message.” EX1007, 1:6-10. Groner discloses a method and system in which “[a]n audio message from a caller for a recipient is received,” “[a] text message file is generated from the audio message from the caller,” and “[t]he text message file is sent to the recipient at the recipient’s e-mail address.” *Id.*, 2:63-67.

217. A POSITA would have been motivated to combine Richardson and Madnick with Groner to allow Richardson’s input messages (e.g., from customers) to be received over the phone via voice mail and then converted to text for

purposes of identifying and extracting information with regular expressions. Groner expressly motivates this combination, explaining that “[v]oice mail messages ... are convenient because telephones are ubiquitous and inexpensive.” EX1007, 2:17-20. Moreover, by converting voice mail to text, “the voice messages can be searched for particular content.” *Id.*, 3:37-39.

218. Richardson acknowledges that customers use telephones to communicate since one of its exemplary regular expressions is for identifying telephone numbers in customer emails. EX1005, 5:55, 8:63, 9:26-27. While Richardson highlights the “increased popularity of electronic messaging,” it acknowledges “other messaging media, such as telephone calls.” *Id.*, 1:32-34. Groner’s method of converting voice mail to email fits logically with Richardson’s exemplary embodiment of a company receiving customer inquiries because customers may use both email and phone calls to communicate.

219. In combining Richardson-Madnick with Groner, a POSITA would have had a reasonable expectation of success. Because Groner converts voice messages to emails, Richardson’s method of processing emails requires no modification. A POSITA would have understood that combining Richardson-Madnick with Groner simply allows customers to contact the company by phone in addition to the existing ability to contact the company by email. Once the voice

message is transcribed to text and sent via email as described in Groner, it is processed in the same manner as any other email as described in Richardson.

220. Implementing Groner in the context of Richardson-Madnick would have been routine to a POSITA. Groner notes that “[t]he accuracy of speech recognition software has improved,” with several commercial embodiments available, and expressly discloses a computer system with a “voice-to-electronic mail system,” including “a speech recognition procedure.” *Id.*, 2:38-49, 5:5-9, 6:6-8. Moreover, as with Richardson and Madnick, Groner’s invention “is implemented as a computer program product that includes a computer program mechanism embedded in a computer readable storage medium.” *Id.*, 16:35-39. Thus, in my opinion, it would have been routine for a POSITA to implement Groner’s software in Richardson’s computer hardware, i.e., server 16.

## **2. Claims 11 and 22**

221. Claims 11 and 22 depend from claims 1 and 12, respectively, and further recite: “*wherein the input sequence of data symbols is representative of a voice mail message.*”

222. In my opinion, the Richardson-Madnick-Groner combination discussed above meets this limitation. Groner teaches that “[a]n audio message from a caller for a recipient is received,” “[a] text message file is generated from the audio message from the caller,” and “[t]he text message file is sent to the

recipient at the recipient's e-mail address." EX1007, 2:63-67. When combined with Richardson-Madnick, the recipient is a company's customer service department, which processes the input sequence of data symbols in the transcribed email in the same manner discussed in claim 1. *Supra* §VII.C.1. Thus, in the Richardson-Madnick-Groner combination, the input sequence of data symbols is representative of a voice mail message.

223. Accordingly, in my opinion, claims 11 and 22 would have been obvious to a POSITA over Richardson, Madnick, and Groner.

### **3. Claim 24**

224. Claim 24 recites as follows:

24. Apparatus for automatically processing an input sequence of data symbols, the apparatus comprising:  
a data capture device for obtaining the input sequence of data symbols;  
at least one processor, operatively coupled to the data capture device, operative to: (i) identify at least one regularly identifiable expression in the input sequence of data symbols. wherein the at least one regularly identifiable expression represents a pattern that is matchable in accordance with a programming language that supports such a regularly identifiable expression; (ii) identify at least a portion of information associated with the at least one regular expression; and (iii) extract the portion of information;

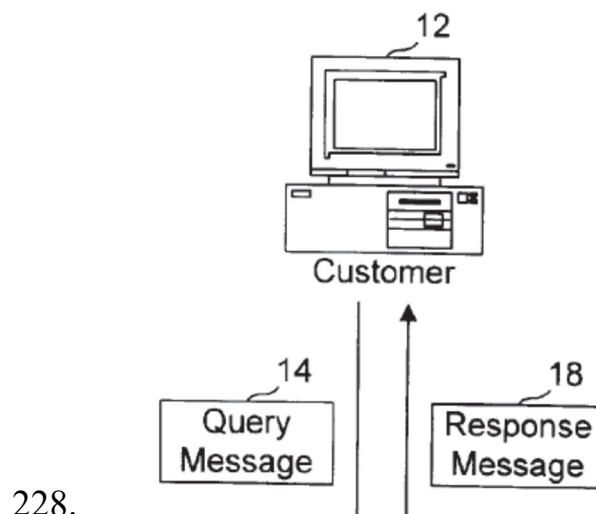
memory, operatively coupled to the at least one processor, for storing at least a portion of results associated with the identifying and extracting operations; and a data output device, operatively coupled to the at least one processor, for presenting the extracted portion of information to a user.

225. In my opinion, claim 24 would have been obvious for substantially the same reasons that I discussed above for claims 1 and 12 in Ground 2. *Supra* §§VII.B.2.a-b. The only difference between claims 24 and 12 is that claim 24 adds (i) “a data capture device for obtaining the input sequence of data symbols”; and (ii) “a data output device, operatively coupled to the at least one processor, for presenting the extracted portion of information to a user.” In my opinion, both limitations are met by the Richardson-Madnick-Groner combination.

226. First, as to the “data capture device,” the ’796 patent explains that a “data capture device may be a speech processing device capable of taking an audio stream recorded in accordance with a voice mail system and converting or transcribing the audio stream to a discrete data or symbol sequence.” EX1001, 11:53-57. A POSITA would have understood that Groner discloses this embodiment. Specifically, Groner discloses “voice-to-electronic mail system 30,” which “automatically converts a spoken message to a text message which is e-mailed to a recipient.” EX1007, 5:5-9. The system comprises “a speech

recognition procedure 116 that receives audio speech, identifies the audio speech and generates a text file 118 corresponding to the identified audio speech.” *Id.*, 6:6-8. Thus, Groner’s voice-to-electronic mail system 30 is a data capture device.

227. Second, as to the “data output device,” the ’796 patent explains that an embodiment of a “data output device [] generally presents the data extracted from the input data stream to a user” and “may be a display for visually presenting the extracted data to the user.” EX1001, 12:52-53, 12:62-63. Likewise, in Richardson, messages are received and sent to customer system 12, a “general purpose computer system[] ... comprising one or more ... video display terminals.” EX1005, 5:11-17. For example, as shown in Figure 1, which is partly reproduced below, response message 18 is sent to customer system 12 comprising a display (depicted as a CRT monitor):



*Id.*, Fig. 1.

229. As I discussed above for claim 1, “the response message text may include ... phrases and extracted information” (*id.*, 7:42-44), including “tags and the values for tagged regular expressions (*id.*, 11:17-23). *See also id.*, Fig. 5(b) (preferred response 130 to customer). A POSITA would have understood that the response generated by Richardson, which includes an extracted portion of information, would be displayed to the customer on a data output device such as the display of customer system 12.

230. At a minimum, a POSITA would have found it obvious to use the display of customer system 12 in Richardson to output response messages comprising extracted information given Richardson’s goal of automating responses to customer inquiries. Thus, claim 24 would have been obvious to a POSITA over Richardson, Madnick, and Groner.

**D. No Secondary Considerations Exist**

231. I am unaware of any secondary considerations of nonobviousness, and I understand no such considerations were alleged during prosecution. I reserve the right to rebut any alleged secondary considerations that Patent Owner or its expert declarant may assert during any instituted *inter partes* review.

## VIII. Conclusion

232. In sum, for the reasons I have discussed above, it is my opinion that claims 1-25 of the '796 patent are unpatentable because they would have been obvious to a POSITA over the prior art references cited above.

233. I understand that this declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also understand that I may be subject to cross-examination concerning this declaration, and I will appear for cross-examination, if required of me, during the time allotted for cross-examination.

234. I hereby declare that all of the statements made herein are true of my own knowledge and that all statements made on information and belief are believed to be true; and further that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code..

Date: January 9, 2023

  
\_\_\_\_\_  
John R. Black, Jr., Ph.D.

# APPENDIX

## John R. Black, Jr.

Department of Computer Science  
430 UCB  
University of Colorado  
Boulder, CO 80309-0430 USA

office: +1 303 492-0573  
FAX: +1 303 492-2844  
secretary: +1 303 492-7514

Email: [jrblack@cs.colorado.edu](mailto:jrblack@cs.colorado.edu)  
WWW: <http://www.cs.colorado.edu/~jrblack/>

---

|                            |                                                                                                                                                                                                                           |                     |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| <b>Position</b>            | Assoc. Prof. Computer Science, University of Colorado at Boulder                                                                                                                                                          | <i>7/08–present</i> |
| <b>Research</b>            | Cryptography, Network Security, Computer Security.                                                                                                                                                                        |                     |
| <b>Past<br/>Employment</b> | <b>SecureSet LLC</b><br>Vice President of Education                                                                                                                                                                       | <i>9/15–7/18</i>    |
|                            | <b>University of Colorado at Boulder</b><br>Assistant Professor of Computer Science.                                                                                                                                      | <i>7/02–7/08</i>    |
|                            | <b>University of Nevada, Reno</b><br>Assistant Professor of Computer Science.                                                                                                                                             | <i>7/00–6/02</i>    |
|                            | <b>University of California, Davis</b><br>Research Assistant                                                                                                                                                              | <i>7/97–7/00</i>    |
|                            | <b>University of California, Davis</b><br>Teaching Assistant                                                                                                                                                              | <i>8/95–6/97</i>    |
|                            | <b>Ingres Corporation</b><br>Senior Member of Technical Staff                                                                                                                                                             | <i>6/88–4/94</i>    |
| <b>Education</b>           | <b>University of California, Davis</b><br>Ph.D. in Computer Science.<br>Thesis: Message Authentication Codes.<br>Advisor: Phillip Rogaway.                                                                                | <i>9/95–9/00</i>    |
|                            | <b>California State University at Hayward</b><br>B.S. in Computer Science and Mathematics, 1988.<br>Honors: Summa Cum Laude                                                                                               | <i>9/84–6/88</i>    |
| <b>Awards</b>              | NSF CAREER Award, 2002<br>Chancellor's Teaching Fellowship, UC Davis, 1998<br>Outstanding Teaching Assistant, UC Davis, 1998<br>Outstanding Teaching Assistant, UC Davis, 1997<br>A Check for \$2.56 from Don Knuth, 1996 |                     |

rev January 9, 2023

## Journal Publications

1. P. Rogaway, M. Bellare and J. Black, “OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption.” *ACM Transactions on Information and System Security (TISSEC)*, Volume 6, Issue 3, pp. 365–403, August, 2003.
2. J. Black, and P. Rogaway, “CBC MACs for Arbitrary Length Messages: The Three-Key Constructions.” *Journal of Cryptology*, Volume 18, Number 2, pp. 111–132, Spring, 2005.
3. J. Black, “The Impossibility of Technology-Based DRM and a Modest Suggestion.” *Journal of Telecommunications and High-Technology Law —JTHTL*, Volume 3, Number 2, pp. 387–396, Spring, 2005.
4. J. Black, M. Cochran and R. Gardner, “An Analysis of the Internet Chess Club.” *IEEE Security and Privacy*, Volume 4, Number 1, pp. 46–52, January, 2006.
5. J. Black, M. Cochran and T. Shrimpton, “On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions.” *Journal of Cryptology*, Volume 22, Number 3, pp. 311–329, Fall, 2009.
6. J. Black, P. Rogaway, T. Shrimpton, and M. Stam “An Analysis of the Blockcipher-Based Hash Functions from PGV.” *Journal of Cryptology*, Volume 23, Number 4, pp. 519–545, Fall, 2010.
7. C. Wilks, M. Cline, E. Weiler, M. Diehkans, B. Craft, C. Martin, D. Murphy, H. Pierce, J. Black, D. Nelson, B. Litzinger, T. Hatton, L. Maltbie, M. Ainsworth, P. Allen, L. Rosewood, E. Mitchell, B. Smith, J. Warner, J. Groboske, H. Telc, D. Wilson, B. Sanford, H. Schmidt, D. Haussler, D. Maltbie “The Cancer Genomics Hub (CGHub): overcoming cancer through the power of torrential data.” *Database*, Volume 2014, doi: 10.1093/database/bau093.

## Book Chapters

1. J. Black, “Cryptography.” Invited article for the Encyclopedia of Life Support Systems under the auspices of UNESCO. See <http://www.eolss.net>. 14 pages, March, 2004.
2. J. Black, “Authenticated Encryption.” Invited article for the *Encyclopedia of Cryptography and Security*, Springer-Verlag. 12 pages. August, 2005.

**Conference  
Publications  
(Refereed)**

1. J. Black, C. Martel, and H. Qi, "Graph and Hashing Algorithms for Modern Architectures: Design and Performance." *Workshop on Algorithm Engineering — WAE '98*, Saarbrücken, Germany. Second Workshop on Algorithm Engineering, proceedings, pp. 37–48. Full version of this paper available at [theory.cs.ucdavis.edu](http://theory.cs.ucdavis.edu).
2. J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, "UMAC: Fast and Secure Message Authentication." *Advances in Cryptology — CRYPTO '99*, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, pp. 216–233, 1999. Full version and updated version of this paper available at [www.cs.ucdavis.edu/~rogaway/umac](http://www.cs.ucdavis.edu/~rogaway/umac).
3. J. Black and P. Rogaway, "CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions." *Advances in Cryptology — CRYPTO 2000*, Lecture Notes in Computer Science, Vol. 1880, Springer-Verlag, pp. 197–215, 2000.
4. P. Rogaway, M. Bellare, J. Black and T. Krovetz, "OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption." *Eighth ACM Conference on Computer and Communications Security (CCS-8)*, ACM Press, pp. 196–205, 2001.
5. J. Black and P. Rogaway, "Enciphering Finite Sets of Arbitrary Size." *RSA Data Security Conference, Cryptographer's Track (RSA-CT)*, Lecture Notes in Computer Science, Vol. 2271, Springer-Verlag, pp. 114–130, 2002.
6. J. Black and P. Rogaway, "A Block-Cipher Mode of Operation for Parallelizable Message Authentication." *Advances in Cryptology — EUROCRYPT 2002*, Lecture Notes in Computer Science, Vol. 2332, Springer-Verlag, pp. 384–397, 2002.
7. J. Black and H. Urtubia, "Side-Channel Attacks on Symmetric Encryption Schemes: The Case for Authenticated Encryption." *USENIX Security Symposium — Security '02*. 10 pages, 2002.
8. J. Black, P. Rogaway, and T. Shrimpton, "Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV." *Advances in Cryptology — CRYPTO 2002*, Lecture Notes in Computer Science, Vol. 2442. 16 pages, 2002.
9. J. Black, P. Rogaway, and T. Shrimpton, "Encryption Scheme Security in the Presence of Key-Dependent Messages." *Selected Areas in Cryptography — SAC 2002*, Lecture Notes in Computer Science, Vol. 2595, 14 pages, 2002.
10. R. Motwani, J. Breidenbach and J. Black, "Collocated Dataglyphs for Large Message Storage and Retrieval." *Security, Steganography, and Watermarking of Multimedia Contents VI*, Society for Imaging Science and Technology (I&ST) jointly with International Society for Optical Engineering (SPIE), Vol. 5306, 19 pages, 2004.
11. J. Black, M. Cochran and T. Shrimpton, "On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions." *Advances in Cryptology — EUROCRYPT 2005*, Lecture Notes in Computer Science, Vol. 3494, Springer-Verlag, pp. 526–541, 2005.
12. J. Black, M. Cochran and R. Gardner, "Lessons Learned: A Security Analysis of the Internet Chess Club.", *Annual Computer Security Applications Conference — ACSAC 2005*, Tucson AZ, USA, pp. 220–228, 2005.
13. J. Black and M. Cochran and T. Highland, "A Study of the MD5 Attacks: Insights and Improvements", *Fast Software Encryption — FSE 2006*, Lecture Notes in Computer Science, Vol. 4047, Springer-Verlag, pp. 262–277, 2006.

**Conference  
Publications  
(cont.)**

14. J. Black, “The Ideal-Cipher Model, Revisited: An Uninstantiable Blockcipher-Based Hash Function.”, *Fast Software Encryption — FSE 2006*, Lecture Notes in Computer Science, Vol. 4047, Springer-Verlag, pp. 328–340, 2006.
15. J. Black, “Compare-by-Hash: A Reasoned Analysis”, *USENIX Annual Technical Conference — USENIX 2006*, 8 pages, 2006.
16. J. Black and M. Cochran, “MAC Reforgeability”, *Fast Software Encryption — FSE 2009*, Lecture Notes in Computer Science, Vol. 5665, Springer-Verlag, pp. 345–362, 2009.
17. J.H. Huang, J. Black, and S. Mishra, “Security and Privacy in a Sensor-Based Search and Rescue System,” *1st ICST/CREATE-NET International Conference on Ad Hoc Networks — ADHOCNETS 2009*, Vol. 28, Springer.
18. A. Sayler, D. Grunwald, J. Black, E. White, M. Monaco, “Supporting CS education via virtualization and packages: tools for successfully accommodating “bring-your-own-device” at scale,” *SIGCSE 2014*, pp. 313-318, 2014.

**Workshop  
Publications  
(Non-  
Refereed)**

1. J. Black and P. Rogaway, “A Suggestion for Handling Arbitrary-Length Messages with the CBC MAC.” *NIST Symmetric Key Block Cipher Modes of Operation Workshop—2000*, 4 pages, Sep 2000.
2. J. Black and P. Rogaway, “OCB: Proposal to NIST.” *2nd NIST Symmetric Key Block Cipher Modes of Operation Workshop—2001*, 36 pages, Aug 2001.
3. J. Black and P. Rogaway, “PMAC: Proposal to NIST.” *2nd NIST Symmetric Key Block Cipher Modes of Operation Workshop—2001*, 27 pages, Aug 2001.

**Selected  
Talks**

1. Data Structures for Fast Graph Algorithms. Presented at the 1997 UC Davis Workshop on Computing, Davis, USA, October 1997. (See Conference Publication #1.)
2. UMAC: Fast and Secure Message Authentication. Presented at CRYPTO '99, Santa Barbara, USA, August 1999. (See Conference Publication #2)
3. CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. Presented at CRYPTO 2000, Santa Barbara, USA, August 2000. (See Conference Publication #3)
4. A Suggestion for Handling Arbitrary-Length Messages with the CBC MAC. Presented at NIST Symmetric Key Block Cipher Modes of Operation Workshop—2000, October, 2000; also presented at the 2nd NIST Modes Workshop in Santa Barbara, USA, August 2001.
5. Enciphering Finite Sets of Arbitrary Size. Presented at RSA-CT '02, San Jose, USA, February 2002. (See Conference Publication #5)
6. A Block-Cipher Mode of Operation for Parallelizable Message Authentication. Presented at EUROCRYPT 2002, Amsterdam, The Netherlands, May 2002. (See Conference Publication #6)
7. Side-Channel Attacks on Symmetric Encryption Schemes. Presented at USENIX Security 2002, San Francisco, USA, August 2002. (See Conference Publication #7)
8. Practical Cryptography and Autonomic Web Computing. Invited talk at the 47th meeting of the IFIP Working Group 10.4. Rincon, Puerto Rico, January 2005.
9. On the Impossibility of Highly-Efficient Blockcipher-Based Hash Functions. Presented at EUROCRYPT 2005, Aarhus, Denmark, May 2005. (See Conference Publication #11)
10. The Ideal-Cipher Model, Revisited: An Uninstantiable Blockcipher-Based Hash Function. Presented at FSE 2006, Graz, Austria, March 2006. (See Conference Publication #14)
11. Compare-by-Hash: A Reasoned Analysis. Presented at USENIX Technical Conference 2007, Boston, MA, June 2006. (See Conference Publication #15)

**Patents**

1. T. McSheery, J. Black, S. Nollet, J. Johnson, and V. Jivan. Distributed-Processing Motion Tracking System for Tracking Individually Modulated Light Points. US Patent 6,324,296 B1. November 2001.

**Funding**

1. NCHIA E-Team Grant. "Entrepreneurship for Undergraduates." PI: John Black. Period: 2000-2001. Amount: \$6,000.
2. University of Nevada Junior Faculty Research Grant. "Fast, Provably-Secure Cryptography." PI: John Black. Period: 2001-2002. Amount: \$10,000.
3. NSF CAREER Award. "Highly-Optimized Provably-Secure Cryptography." PI: John Black. Period: 2002-2007. Amount: \$469,925.
4. NSF NeTS Grant. "NeTS ProWIN: Topology And Routing With Steerable Antennas." PI: Dirk Grunwald. Co-PIs: John Black, Douglas Sicker. Period: 2005-2008. Amount: \$745,215.
5. NSF Cybertrust Grant. "Cryptography for Constrained Environments." PI: John Black. Period: 2005-2008. Amount: \$294,887.

## Teaching History

ECS 122A — Design and Analysis of Algorithms (UC Davis). Co-taught once with Professor Rogaway; subsequently taught the course independently.

CMPSC 290G — Intro to Cryptoanalysis (UCSB).

CS 365 — Discrete Mathematics (UNR).

CS 425 — Software Engineering (UNR).

CS 426 — Senior Projects (UNR). Supervised 11 group projects in topics ranging from fingerprint recognition to audio editing to GUI design.

CS 432 — Computer Networks (UNR). Introduction to low-level networking concepts with an emphasis on network security.

CS 665 — Graduate Analysis of Algorithms (UNR). A typical algorithms course with emphasis on complexity theory.

CS 709 — Modern Cryptography (UNR). A graduate course introducing cryptography and visiting some of the research front.

CS 791G — Computer Network Security (UNR). A seminar course covering various topics related to network security.

CSCI 2270 — Data Structures (CU); Program design, Object orientation, Java, Linked lists, Arrays, Stacks and Queues, Hash tables, Trees, Balanced Binary Trees, Multi-core programming.

CSCI 3104 — Algorithms (CU); Divide-and-conquer, Greedy, Graph Algorithms, NP-Completeness, Quantum Algorithms.

CSCI 3753 — Operating Systems (CU); Scheduling, Virtual Memory, Filesystems, Multi-core systems, Pthreads, Kernel data structures, virtualization, Security.

CSCI 4830 — Network Security (CU); a new course developed to introduce basics of cryptography and network security. Covers SSL, PKI, DDOS attacks, wireless security, buffer overruns, and more.

CSCI 4900 — Solving Puzzles with Computers (CU); a one-unit undergraduate course describing some hard combinatorial puzzles and how computers can be used to attack them.

CSCI 5413 — Ethical Hacking (CU); Network security, nmap, netcat, Kali Linux, Buffer overruns, Format string vulnerabilities, Race Conditions, Web Security, SQL Injections, XSS, CSRF, Wireless.

CSCI 6268 — Foundations of Computer and Network Security (CU); an introductory course covering basic cryptography, cryptographic protocols, attacks, and principles, as well as core network security attack and defense.

CSCI 7000 — Cryptography Seminar (CU); A graduate course introducing basic cryptographic definitions and then making some forays to the research front.

CSCI 7000 — Cryptanalysis Seminar (CU); A graduate course introducing students to cryptanalysis. Differential and linear cryptanalysis, square attack, RSA basics, factoring, protocol errors, lattices, Coppersmith's algorithm.

CSCI 7000 — Quantum Computing (CU); Introduction to quantum circuits, number theory, Shor's Algorithm, Grover's Algorithm.

## Graduate Students

Rakhi Motwani, M.S., Completed: Spring 2002.

Scott Fritzinger, M.S., Completed: Summer 2002.

Hector Urtubia, M.S., Completed: Spring 2003.

Hiba Fayoumi, M.S., Completed: Summer 2004.

Mary Hedges, M.S., Completed: Spring 2007.

Joeseph Dunn, Ph.D., co-advisor with John Bennett, Completed: Summer 2007.

Martin Cochran, Ph.D., Completed: Spring 2008.

Jared Nishikawa, Ph.D., Completed: Spring 2016.

**Undergraduate Students**    Troy Trimble, University of California at San Diego. REU Student, Summer 2003.  
Gagan Sekhon, California State University at Hayward. REU Student, Summer 2003.  
Ryan Gardner, University of Colorado at Boulder. REU Student, Summer 2004.  
Trevor Highland, University of Texas at Austin. REU Student, Summer 2005.

**External  
Service**

Secretary, International Association for Cryptologic Research, 2005–2007.  
Program Committee, ACNS 2015.  
Program Committee, CT-RSA 2015.  
Program Committee, FSE 2014.  
Program Committee, CT-RSA 2014.  
Program Committee, CT-RSA 2013.  
Program Committee, Eurocrypt 2012.  
Program Committee, FSE 2011.  
Program Committee, PKC 2011.  
Program Committee, CANS 2010.  
General Chair, CRYPTO 2009.  
Program Committee, CRYPTO 2008.  
Program Committee, ACNS 2008.  
Program Committee, RSA-CT 2007.  
Program Committee, ISC 2007.  
Program Committee, ACNS 2007.  
Program Committee, ACM CCS 2006.  
Program Committee, CANS 2006.  
Program Committee, ICISC 2006.  
Program Committee, SECURE 2006.  
Program Committee, ACSAC 2006.  
Program Committee, CRYPTO 2005.  
Program Committee, SAC 2005.  
Program Committee, ICISC 2005.  
Program Committee, CANS 2005.  
Program Committee, IEEE SISW 2005.  
Program Committee, CRYPTO 2004.  
Program Committee, EUROCRYPT 2004.  
Program Committee, RSA-CT 2003.  
NSF CISE Panelist, 2001, 2003, 2005, 2006, 2007, 2009.  
Referee for Journal of Cryptography, 1999–2006.  
Referee for Software: Practice and Experience, 2005.  
Referee for IEEE Communications Magazine, 2005.  
Referee for IEEE Transactions on Circuits and Systems I, 2005.  
Referee for IEEE Computer, 2005.  
Referee for Journal of Computer Security, 2004.  
Referee for IEEE Transactions on Information Theory, 2003.  
Referee for IEEE Transactions on Computers, 2002.  
Reviewer for CRYPTO 1999–2002, SODA 1998, SPAA 2002, Asiacrypt 2004, EURO-CRYPT 2006.

Developed **CryptoStats** web site: an application which tracks publication rates by year, by author, by conference in the two main cryptography conferences. It was heavily used in my community (on average 240 hits per month), 2003–2009.

ACM Programming Contest problem composer, 2003–2007.  
ACM Programming Contest site administrator, 2005.

Graduate Student Mixer organizer, CRYPTO 2005.

**Internal  
Service**

Chair, Computing Committee, 2010–2011, 2012–2015.

Chair, Space Committee, 2012–2015.

Liaison, Casey Feldman Foundation, 2010–2015.

Member, Departmental Executive Committee, 2003–2005, 2007–2009, 2013–2015.

Member, Executive Committee, Computer and Communications Security Center, 2003–2006.

Member, Departmental Search Committee, 2003–2006, 2012–2014.

Chair, Departmental Search Committee, 2008–2009.

Member, Graduate Committee, 2005–2006.

Developed departmental voting software, now used for all departmental votes and college votes.