

# EXHIBIT 1

IN THE UNITED STATES DISTRICT COURT  
FOR THE EASTERN DISTRICT OF TEXAS  
MARSHALL DIVISION

BRIGHT RESPONSE, LLC  
F/K/A POLARIS IP, LLC

v.

GOOGLE INC., et al.

NO. 2:07CV-371-TJW-CE

**REPORT OF DEFENDANTS' EXPERT  
L. KARL BRANTING, PH.D, J.D.  
CONCERNING INVALIDITY OF CLAIMS 26, 28, 30, 31, 33, AND 38  
OF U.S. PATENT NO. 6,411,947**

## TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION</b> .....	<b>1</b>
<b>II.</b>	<b>QUALIFICATIONS</b> .....	<b>1</b>
<b>III.</b>	<b>LEGAL PRINCIPLES</b> .....	<b>3</b>
<b>IV.</b>	<b>OVERVIEW OF THE '947 PATENT</b> .....	<b>6</b>
A.	THE '947 PATENT GENERALLY .....	7
B.	THE '947 PATENT CLAIMS .....	10
C.	CHARACTERISTICS OF THE METHODS AND SYSTEM CLAIMED BY THE '947 PATENT .....	12
1.	<i>A method for automatically processing a non-interactive electronic message using a computer (Claim 26[preamble]).</i> .....	12
2.	<i>Receiving the electronic message from a source (Claim 26[a]).</i> .....	12
3.	<i>Interpreting the electronic message using a rule base and case base knowledge engine (Claim 26[b]).</i> ...13	
4.	<i>Retrieving one or more predetermined responses corresponding to the interpretation of the electronic message from a repository for automatic delivery to the source (Claim 26[c]).</i> .....	13
5.	<i>Ordering</i> .....	13
6.	<i>Dependent Claims</i> .....	13
<b>V.</b>	<b>THE SCOPE AND CONTENT OF THE PRIOR ART</b> .....	<b>15</b>
A.	THE PRIOR ART GENERALLY.....	15
B.	EXEMPLARY PRIOR ART REFERENCES.....	20
1.	<i>Allen</i> .....	20
2.	<i>CBR-Express</i> .....	25
3.	<i>Nguyen</i> .....	27
4.	<i>EZ Reader</i> .....	32
5.	<i>GREBE</i> .....	41
6.	<i>Goodman</i> .....	44
7.	<i>Watson</i> .....	48
<b>VI.</b>	<b>THE ASSERTED CLAIMS OF THE '947 PATENT ARE INVALID AS ANTICIPATED</b> .....	<b>52</b>
A.	ALLEN ANTICIPATES CLAIMS 26, 28, 30, 31, AND 38.....	52
1.	<i>Allen anticipates Claim 26.</i> .....	53
2.	<i>Allen anticipates Claim 28.</i> .....	56
3.	<i>Allen anticipates Claim 30.</i> .....	57
4.	<i>Allen anticipates Claim 31.</i> .....	58
5.	<i>Allen anticipates Claim 38.</i> .....	59
B.	THE CBR EXPRESS MANUALS ANTICIPATE AND RENDER OBVIOUS CLAIMS 26, 28, 30, 31, AND 33. ....	60
1.	<i>The CBR Express Manuals anticipate and render obvious claim 26.</i> .....	61
2.	<i>The CBR Express Manuals anticipate and render obvious claim 28.</i> .....	63
3.	<i>The CBR Express Manuals anticipate and render obvious claim 30.</i> .....	64
4.	<i>The CBR Express Manuals anticipate and render obvious claim 31.</i> .....	66
5.	<i>The CBR Express Manuals anticipate and render obvious claim 33.</i> .....	67
C.	NGUYEN ANTICIPATES CLAIMS 26 AND 28.....	67
1.	<i>Nguyen anticipates Claim 26.</i> .....	67
2.	<i>Nguyen anticipates Claim 28.</i> .....	72
D.	EZ READER ANTICIPATES CLAIMS 26, 28, 30, 31, 33, AND 38. ....	73
1.	<i>EZ Reader anticipates claim 26.</i> .....	76
2.	<i>EZ Reader anticipates claim 28.</i> .....	78
3.	<i>EZ Reader anticipates claim 30.</i> .....	78
4.	<i>EZ Reader anticipates claim 31.</i> .....	80
5.	<i>EZ Reader anticipates claim 33.</i> .....	80
6.	<i>EZ Reader anticipates claim 38.</i> .....	80

retrieving one or more predetermined responses corresponding to the interpretation of the electronic message from a repository for automatic delivery to the source.

**A. The '947 Patent Generally**

23. The '947 patent describes a system designed for automatically processing emails.

According to the specification, as businesses go “online” they need to process and respond to an increasing number of emails. Rather than hiring additional employees and/or requiring those employees to work longer hours, the specification details a system for automatically responding to some emails so as to lower the amount of email traffic that employees need to review. ('947 patent, 1:26-59.)

24. The specification acknowledges that there were existing solutions for automatically processing email. One such approach, identified as “rule based reasoning,” applied a series of “IF-THEN” rules (conditions) to determine how to process incoming messages. For instance, if the user knows he will not be in the office that day, he may specify an “out-of-office” email to automatically respond to incoming messages. The user may further specify different responses based on the identity of the sender. ('947 patent, 1:60 – 2:7.)

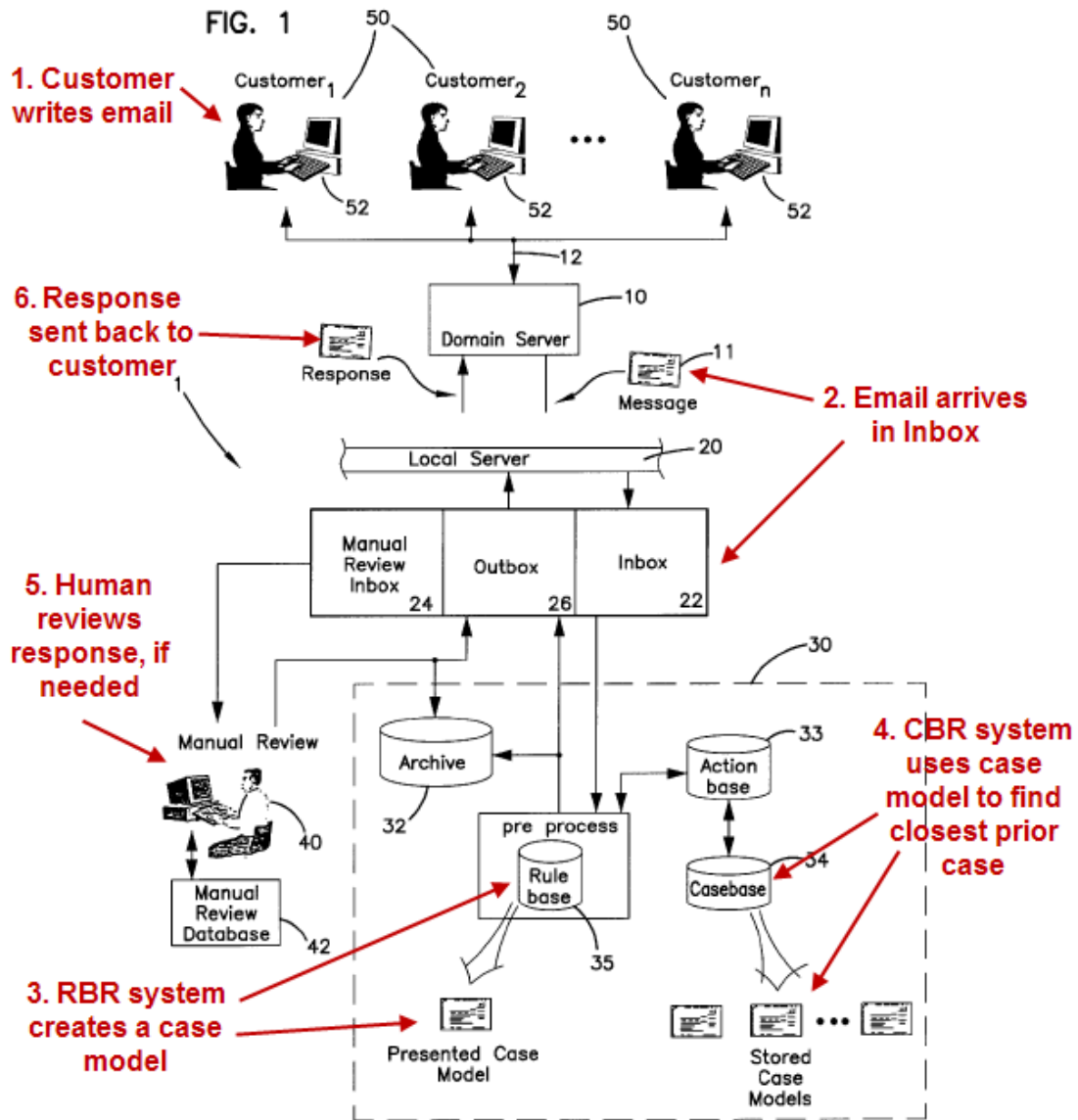
25. The specification also discusses prior art case-base reasoning systems. In particular, the specification discusses U.S. Patent No. 5,581,664 to Allen, which describes a help-desk system that employs case-based reasoning.<sup>2</sup> Allen receives a problem (e.g., “my computer shows a Bluescreen of Death”) and compares it to a stored set of previous problems. Once Allen finds the stored problem that is most similar to the current problem, Allen applies or adapts the previous solution to the current problem. In other words, Allen reasons by analogy: in the Bluescreen example, Allen would look for any previous instances involving a Bluescreen of

---

<sup>2</sup> As noted in section V.B.1 below, Allen also discloses rule based reasoning, though the applicants omitted this fact in their description.

Death, and use the past solution (presumably, “reboot”) as a basis for solving the current problem. (‘947 patent, 2:41-62.)

26. The alleged invention is a method of processing incoming email messages, as depicted in Figure 1. (‘947 patent, 5:54 – 9:35.)



(Annotations added)

27. The rule-based reasoning system disclosed in the ‘947 patent performs two functions. First, it creates a “presented case model” out of the incoming email message. (‘947 patent, 6:53-61.) Second, it may be able to classify the message into either “automatic” (capable

of being responded to automatically) and “referred” or “detected” (not capable of being responded to automatically). (*Id.*, 6:62 – 7:6.) If the email message is classified, then the system skips the case-based reasoning step. (*Id.*, 7:31-33.) Sample question rules are included in Table 1. For instance, if the message is blank, then the message can be automatically responded to, likely with a standard request that the user to include a message. Similarly, if the message requests a change-of-address, there needs to be human review of the message, likely to ensure that the new address is entered into the customer database:

TABLE 1

<u>(Question Rules)</u>	
Question Rule Detects	Action Type: Classification
customer's address	attribute/action: detected
blank body	action: automatic
service cancellation	action: referral
change address	action: referral
no call back	attribute/action: detected
facsimile number	action: detected
phone number	action: detected
request for service	attribute
reference to error	attribute
investments	attribute
foreign country	attribute/action: detected
lengthy message	attribute
specific product	attribute
multiple questions	attribute
specific employee	attribute/action: detected
lost product	action: referral
sensitive info	action: refer, auto, detected

(Highlighting added)

28. If the rule-based reasoning system is unable to classify the message, then the presented case model created earlier is used within the case-based reasoning system:

TABLE 2

Example: Stored Case Model	
TITLE:	Lengthy Inquiry From lmnop.com
SUBJECT:	New Account
DESCRIPTION:	Dear Sirs, I recently saw your advertisement in the newspaper. I am interested in learning whether or not your system would work for me. Please E-mail me back and let me know the details of your system. Thanks,
ACTIONS:	1. type: referral 2. sub-class: sales 3. priority: 3
ATTRIBUTES:	1. address: yes 2. lengthy: yes

29. The patent compares the text (message) and attributes (e.g., whether there is an address) of the presented case model with the text and attributes of each case model stored in the case base. When the text and attributes of the stored case match the text and attributes of the presented case, the match score increases by a predetermined amount. When the text and attributes don't match, the match score decreases by a predetermined amount, which may be zero. ('947 patent, 8:37-57.) The stored case with the highest match score is used as the template for handling the current message: the system may apply or adapt the actions undertaken for the old case model to the new case model. ('947 patent, 7:48 – 9:17.)

30. If the message is classified as “automatic,” then the system retrieves one or more predetermined responses from a repository for delivery to the sender. ('947 patent, 9:24-35.) If not, then the system routes the message to a human operator for review. (*Id.*, 9:43-53.) The human reviewer then reviews the response to be sent back to the customer. (*Id.*, 10:39-47.)

**B. The '947 Patent Claims**

31. The asserted claims are reproduced below:

decreases a stored case model's match score when a feature from the stored case model does not match text and attributes from the presented case model.” (*Id.*) I note that the predetermined mismatch weight may be zero, as claim 32 (which depends from claim 31) specifically requires that this be so.

(d) **Normalizing match scores (Claim 33)**

41. Claim 33 requires that “each score is normalized by dividing the score by a maximum possible score for the stored case model, where the maximum possible score is determined when all of the attributes and text of the case model and the stored case model match.” The parties agreed that the first part of the phrase means “wherein each match score is divided by the maximum possible score for the stored case model.” (Order at 7.)

(e) **Altering the predetermined response (Claim 38)**

42. Claim 38 requires that “the predetermined response is altered in accordance the interpretation of the electronic message before delivery to the source.”

**V. THE SCOPE AND CONTENT OF THE PRIOR ART**

**A. The Prior Art Generally**

43. Case-based reasoning (CBR) is a problem-solving paradigm in which stored cases—which may be solutions to previous problems, prototypes, or exemplars—are used to solve new problems (consistent with the terminology of the '947 patent, a new problem will be referred to as the “presented case,” “new problem,” or “new case”). Case-based reasoning is the computer equivalent of the universal human strategy of solving new problems by reusing solutions to old problems. *See, e.g.,* A. Aamodt, E. Plaza (1994), *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches Artificial Intelligence Communications*, IOS Press, Vol. 7: 1, pp. 39-59. For example, the best way to design a new house is often to start with a good existing house design and modify it to fit the specific needs of the new buyer, such as the shape of the lot, roof color, window types, and so forth.



44. Research in CBR dates back at least to the early 1980's, with DARPA-sponsored workshops in CBR held in the United States in 1988, 1989, and 1991. By the early 1990s, more than a hundred different CBR systems had been developed for a wide range of applications, including, among many others, diagnosis of heart disease, automated message answering, arbitration, hearing disorder diagnosis, mechanical and architectural design, military planning, legal analysis, computer-aided instruction, jet aircraft repair, autoclave load configuration for jet aircraft part construction, cost estimation, vacation planning, cash-flow forecasting, geometry, chemical synthesis, telephone routing, radiation therapy planning, support of rural health workers, route planning, and agricultural pest management. *See, e.g.*, Watson 1994, pg. 20; The Proceedings of the DARPA Workshop on Case-Based Reasoning, Pensacola Beach, FL, May 31-June 2 1989 (Morgan Kaufmann, San Mateo, CA); The Proceedings of the DARPA Workshop on Case-Based Reasoning, Washington, D.C., May 8-10, 1991 (Morgan Kaufmann, San Mateo, CA); The Proceedings of the DARPA Workshop on Case-Based Reasoning, Clearwater Beach, FL, May 10-13 1988 (Morgan Kaufmann, San Mateo, CA); Case-Based Reasoning Research and Development, Proceedings of the First International Conference, ICCBR-95, Sesimbra, Portugal, Lecture Notes in Artificial Intelligence 1010, Spring (1995).

45. A wide variety of case representations have been used in CBR in the prior art, including key-value pairs, relational structures such as frames and semantic networks, free text, and mixtures of these elements. A key-value pair represents information about a single entity. For example, key-value pairs for a person might include “hair = brown,” “eyes = green,” and “height = 72-inches.” Key-value pairs for an email might include “sender = Mary Smith” and “date = 07/04/2010.” Values in key-value pairs may be symbolic (e.g., “high,” “low”), Boolean (e.g., “yes,” “no”), numeric (e.g., “472”), strings (e.g., “Dear Mr. Jones”), ordinals (e.g., “A, B, C, D, or F”), or selections from lists (e.g., “SUV, compact, pickup, minivan, or sports car”).

Relational information represents how different individuals or things are connected, such as “John is the father of Mary,” and “IBM is the employer of John.” Free text is ordinary written language, such as the text of an email. *See, e.g.,* Bergmann, R., Kolodner, J., and Plaza, E. 2005. Representation in case-based reasoning. *Knowl. Eng. Rev.* 20, 3 (Sep. 2005), 209-213 (citing examples from the early 1990’s of each type of case representation).

46. A rule-based reasoning knowledge engine can be thought of as a series of “IF-THEN” statements that trigger various courses of action; i.e. “a knowledge engine that tests whether one or more conditions are met and, if so, applies specific action.”<sup>4</sup> (Order at 7.) For instance, “IF the phone number dialed is 9-1-1, THEN route the caller to emergency dispatch.” Or “IF the phone number begins with 1, THEN treat the next three digits as the area code.” *See, e.g.,* Buchanan, B. G. and Shortliffe, E. H. 1984 *Rule Based Expert Systems: the Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley Series in Artificial Intelligence)*. Addison-Wesley Longman Publishing Co., Inc.

47. In my experience, many CBR systems also employ rule-based reasoning. Indeed, numerous studies and papers in the prior art disclose that very combination. *See, e.g.,* Edwina L. Rissland & David B. Skalak, “Combining Case-Based and Rule-Based Reasoning: A Heuristic Approach” (1989); M. Fathi-Torbanhan and D. Meyer, “ICARUS: Integrating Rule-Based and Case-Based Reasoning on the Base of Unsharp Systems” (1995); Andrew R. Golding and Paul S. Rosenbloom, “Improving Rule-Based Systems through Case-Based Reasoning” (1991); Andrew R. Golding and Paul S. Rosenbloom, “Improving Accuracy by Combining Rule-Based and Case-Based Reasoning” (1996); Jerzy Surma and Koen Vanhoof, “Integrating Rules and Cases for the Classification Task” (1995); Robert T. H. Chi and Melody Y. Kiang, “An Integrated Approach of Rule-Based and Case-Based Reasoning for Decision Support” (1991); George

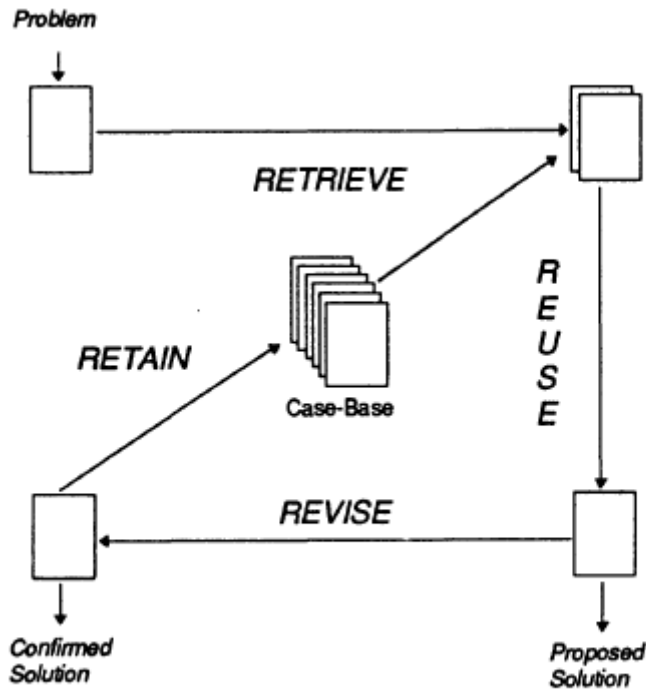
---

<sup>4</sup> The conditions are the “IF”s; the specified actions are the “THEN”s.

Vossos et al., “An Example in Integrating Legal Case Based Reasoning with Object-Oriented Rule-Based Systems: IKBALS II” (1991); and Soumitra Dutta and Piero P. Bonissone, “Integrating Case Based and Rule Based Reasoning: the Possibilistic Connection” (1991).

48. Typical uses of rules within a CBR system include inferring attributes not explicitly stated in the presented case, reasoning about whether the facts of a presented case (e.g., symptoms) can be explained by the facts of a stored case (e.g., a disease), combining the results of multiple case matches, and providing a supplemental or alternative problem-solving procedure after the best case has been located. *See, e.g.*, L. Karl Branting and B. Porter, “Rules and Precedents as Complementary Warrants,” Proceedings of the Ninth National Conference on Artificial Intelligence 1991 (AAAI-91) pp. 3-9.

49. Case-based reasoning systems typically comprise a mechanism that retrieves one or more stored cases that are most relevant to the new problem, adapts the solutions of the retrieved cases to the new problem in light of any differences between the new and old cases, applies the adapted solution to the new problem, and optionally saves the new problem and its solution as a new case. These steps are sometimes referred to as the “Four Rs”: Retrieve the best matching case(s); Reuse those cases to solve the problem; Revise the solution if needed; and Retain the new solution in a new case. *See, e.g.*, Watson 330:



**Figure 1** The CBR cycle (adapted from Aamodt & Plaza, 1994)

50. For example, a CBR system may be used to determine how much to charge a driver for auto insurance. In the “case creation” step, the system would build a case model of the new driver, including information such as the driver’s age, sex, marital status, driving history, make and model of car, etc. In the “retrieval” step, the system would look through its database of already insured drivers to determine how much to charge the new driver. Typically, there will not be an exact match, so the system will select several driver profiles which are “close” to the new driver. Some attributes may be more important than others—for instance, the system may consider drivers having the same driving history to be “closer” than drivers having the same sex. In the “reuse” step, the system would determine how much each of other drivers was charged, and use that data to compute an insurance quote for the new driver. In the “revise” step, the system may adjust the solution computed in the “reuse” step if needed. Finally, in the optional “retain” step, the system may save the new driver’s information and insurance rate in the system, so that it can be used during the next search. *See, e.g.,* Andrew R. Golding and Paul S.

Rosenbloom, “Improving accuracy by combining rule-based and case-based reasoning,” *Artificial Intelligence* 87 (1996) 215-254.

51. Commercial vendors of CBR technology, such as Inference Corporation and Cognitive Systems, marketed software CBR “Tools” such CBR Express and ART\**Enterprise*, which provided reusable software for creating libraries of cases for use in new CBR systems. CBR tools typically provided software for each of the stages of CBR—retrieval, reuse, revise, and retain—permitting users of the tools to provide only information specific to their particular domain. Tool users were typically not able to devise new case representations, retrieval procedures, adaptation mechanisms, or other new CBR elements that were not already provided by the tool. Thus, CBR systems created using a CBR tool were typically limited to prior art because such tools inherently precluded any technical novelty regarding the components that they provide.

**B. Exemplary Prior Art References**

**1. Allen**

52. U.S. Patent 5,581,664 “Case-Based Reasoning System” by Allen et al. (“Allen”), which was filed on May 23, 1994 and granted on December 3, 1996, describes an invention that integrates case-based reasoning with rule-based reasoning in a single “inference engine” (Allen 3:10-22). Indeed, the Abstract begins with “[a] case-based reasoning system which is smoothly integrated into a rule-based reasoning system...”

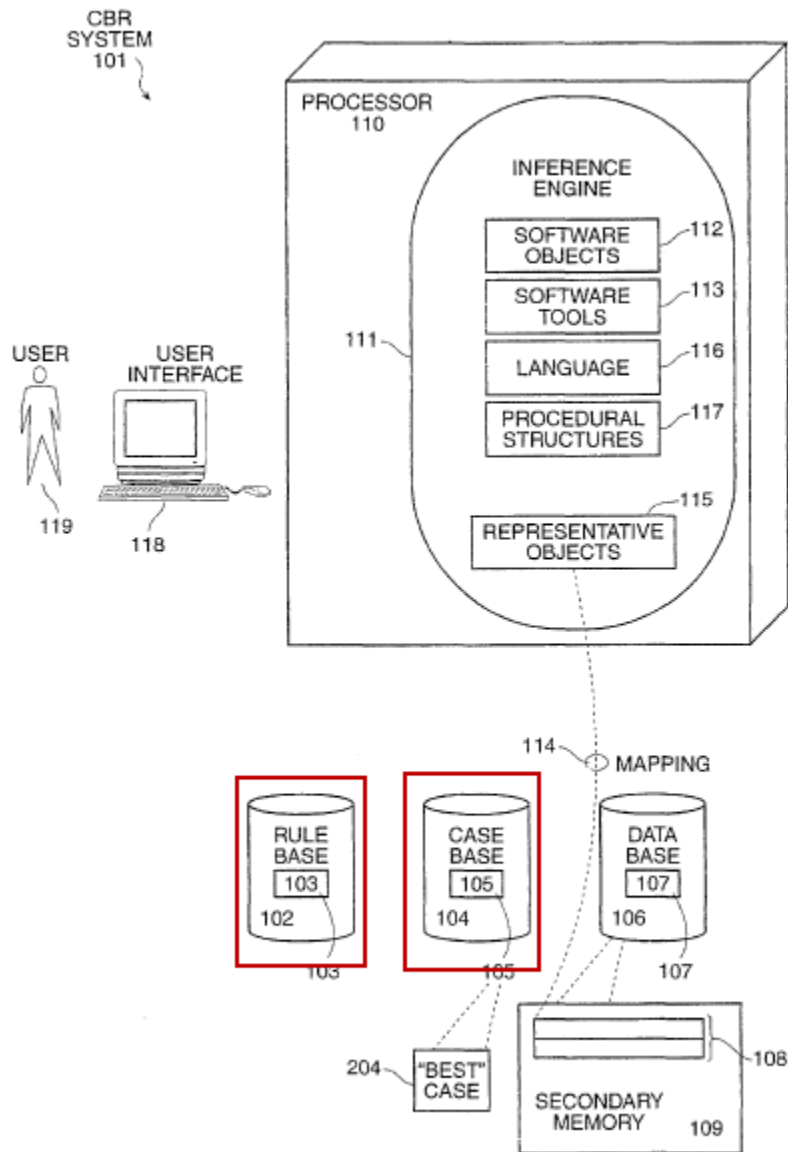


FIG. 1

(Annotations added)

53. The inference engine “retrieves a description of the facts of a particular situation (the 'problem')” and “attempts to match the problem to one or more cases in the case base” (Allen 3:66-4:1). The inference engine attempts to find the best case, note the corresponding action, and perform the action (Allen Fig. 2, 4:3-28).

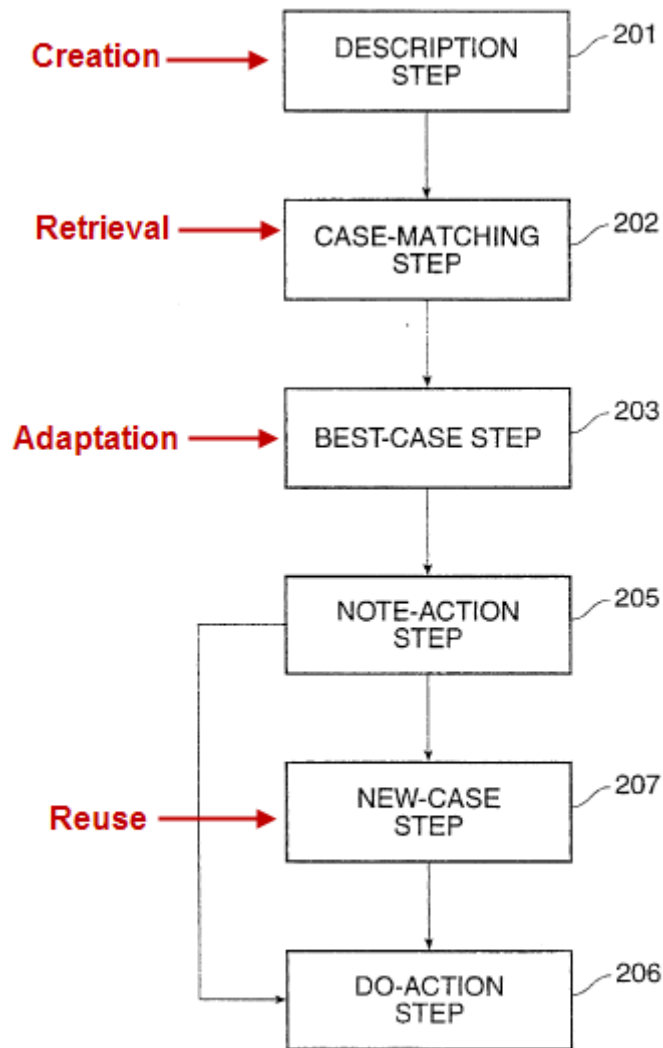


FIG. 2

(Annotations added)

54. Cases may contain a heterogeneous mixture of features, such a Boolean (i.e., yes/no questions), numeric, selections from a list, or textual features (Allen 6:46-51). Text features can be compared by exact string match, word match with stop-words (e.g., articles and conjunctions) removed, character trigram matches, or a weighted combination of all three (*Id.* 6:53-59). Allen matches an incoming case to the cases within the case base by comparing their respective features. (*Id.* 5:20-23.) Each match increases the overall match score by a predetermined amount. (*Id.* 5:23-27.) Allen then computes a match score for those cases, which is used to rank the applicability of the stored case to the incoming problem. (*Id.* 5:28-35.)

55. In the rule-based reasoning portion of Allen, rules may be matched against a set of facts or cases and “may perform procedural actions on them” (Allen 7:16). These procedural actions may include adaptation of the previous case to fit a new problem: “the processor may select the case which is the best match for the problem, but may act differently from the precise action prescribed for that case” (Allen 1:67-2:2). Allen incorporates the CBR-Express Manual, described below, by reference (Allen 10:40-43).

56. Allen discloses a general-purpose rule-base and case-base engine, which can be used to solve any type of problem. This could include diagnosing telephone connection problems, computing auto-insurance rates, or really any other problem with varying degrees of accuracy. Allen also discloses a specific embodiment of a “help desk” system used by operators while dealing with call-in complaints (Allen 8:62 – 10:39):

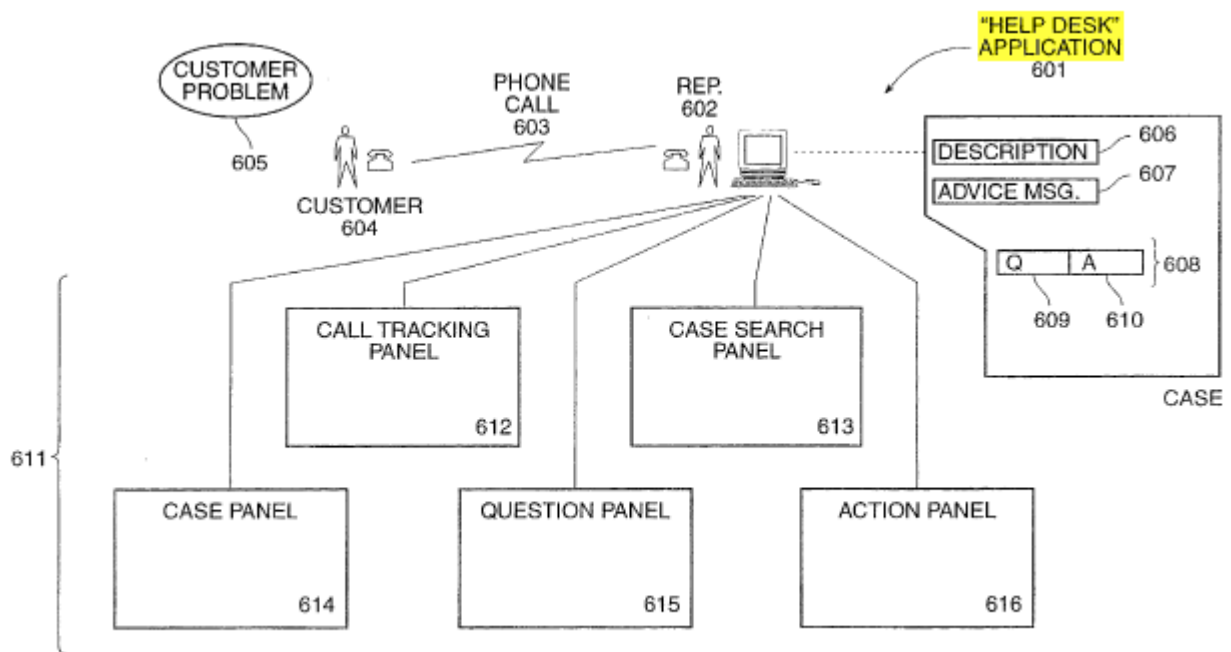


FIG. 6

(Highlighting added)

57. In the help desk application, a set of customer problems and corresponding advice are stored as cases. (Allen 9:10-11.) The customer service representative enters a fact pattern corresponding to the customer’s problem, e.g. “computer does not turn on.” (*Id.* 9:18-20.) Allen then searches through the case-base, trying to match the message text to each of the cases. (*Id.*



9:20-23.) If it finds a good match, the system retrieves the advice associated with that match and presents it to the user, who may then repeat the advice to the customer. (*Id.* 9:23-29.)

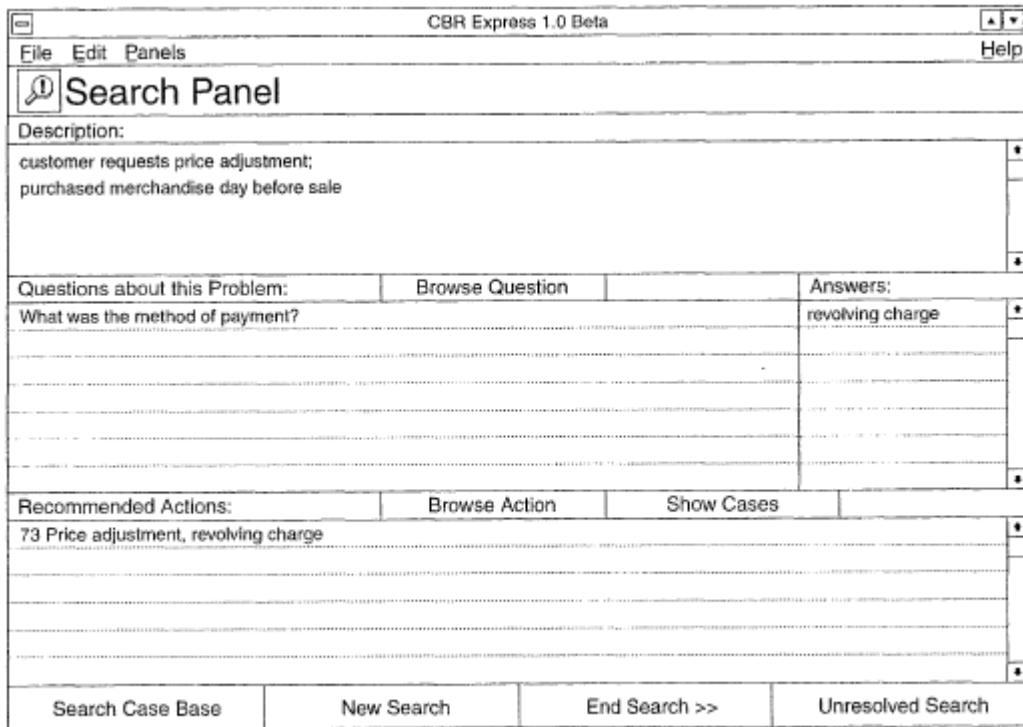


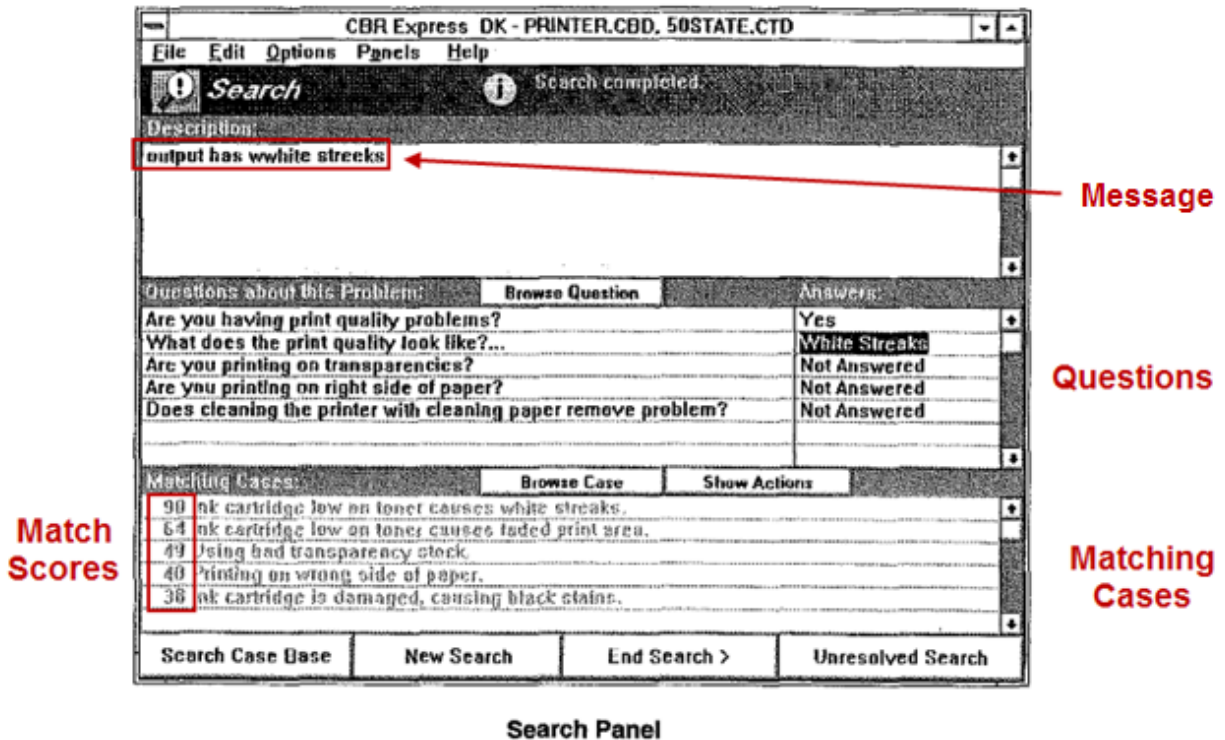
FIG. 6B

58. If the help desk application does not find a good match, then it presents a set of possible matches to the user. The system also presents a series of questions to the user, e.g. “is the power light flashing?” With each answer, the system re-rates the possible matches in order to narrow the search. If it manages to find a good match after the question phase is over, it presents the corresponding advice to the user, who can then repeat it to the customer. (Allen 9:30-41.)

59. If the help desk *still* cannot locate a good matching case, it simply asks the user to enter the new case information into the case base. Once the customer’s problem is resolved, the user can also add the corresponding advice to that case base entry. In this manner, the case base grows when it encounters new problems, and future users can make use of the learned solution. (Allen 9:42-50.)

## 2. CBR-Express

60. The Inference Corporation CBR Express 2.0 for Windows Users Guide, Copyright 1990-1995, (“User’s Guide”) and The Inference Corporation CBR Express CBR Express 2.0 for Windows Reference Manual, Copyright 1990-1995, (“Reference Manual”) describe a commercial help-desk product for development of case-based reasoning applications. (See June 28, 2010 Declaration of Bradley Allen.) This corresponds very closely to the preferred embodiment of Allen; indeed, Allen explicitly discloses that “a preferred example case-based reasoning system 101 for providing user help on call-in complaints is more fully described in ‘CBR Express User's Guide’, available from Inference Corporation of El Segundo, Calif.” (Allen 10:40-44.) Page 51 of the User’s Guide shows a sample input screen:



(Annotations added)

61. The user enters a message, e.g. “output has wwhite [sic] streaks.” CBR-Express then attempts to match the words in the message with the cases within the case base. Prospective matching cases are listed at the bottom of the screen, along with a match score between 0 and

100. (User's Guide, p. 51.) "0" corresponds to no match at all; "100" corresponds to a perfect match. (Reference Manual, p. 15.)

62. CBR-Express may also present a series of questions to the user, though an administrator may disable this feature. (Reference Manual, p. 14.) As users answer each question, e.g. "Are you having print quality problems," CBR-Express re-computes match scores and re-ranks the cases that are presenting to the user. (User's Guide. pp. 52-53.) The questions correspond to features of the case models, and may accept Yes/No answers, an answer selected from a list of options, numeric entries, and text entries. (*Id.*)

63. As users answer questions, they may browse the matching cases presented in the window. While the case with the highest match score is likely the best solution, it is possible than a lesser ranked case may be more appropriate. Users may freely browse any of the available cases during their search. (User's Guide, p. 55.) If the user is unable to find a matching case, he may "flag" the question so that it can be addressed by a more senior technical expert. (*Id.*, p. 56.)

64. Behind the scenes, CBR-Express employs matching algorithms similar to those described in the Allen patent. CBR-Express employs a character matching algorithm to attempt to match the words within the user's message to the text description of each case in the case base. After discarding stop words (e.g., "the"), punctuation marks, suffixes, etc., CBR-Express employs trigram (three character) matching. (Reference Manual, p. 18.) Each time a trigram from the message matches a trigram from the case description, the match score for that case increases by some amount. (*Id.*) CBR-Express thus computes match scores for all the cases in the case-base, then presents the best results to the user.

65. Additionally, CBR-Express presents questions that correspond to the features of the top cases. For instance, CBR-Express may ask the user "Are you printing on transparencies," which has a Boolean or "yes/no" answer. Assuming the user answers "yes," cases that have the "printing on transparencies" feature would have their match scores incremented by a match

weight, while cases that do not print on transparencies<sup>5</sup> would have their match scores decremented by a mismatch weight. (Reference Manual, pp. 14-15.) The match weight and mismatch weight may differ depending on the importance of the question. For example, the “patient is pregnant” case may have a massive mismatch score if the patient is not female!

66. CBR-Express compares the features of each case in the case base to the features of the incoming case. (Reference Manual, pp. 14-15.) The resultant match weights and mismatch weights are added together to form a total match score, which CBR-Express normalizes to a range between 0 and 100. (*Id.*)

### 3. Nguyen

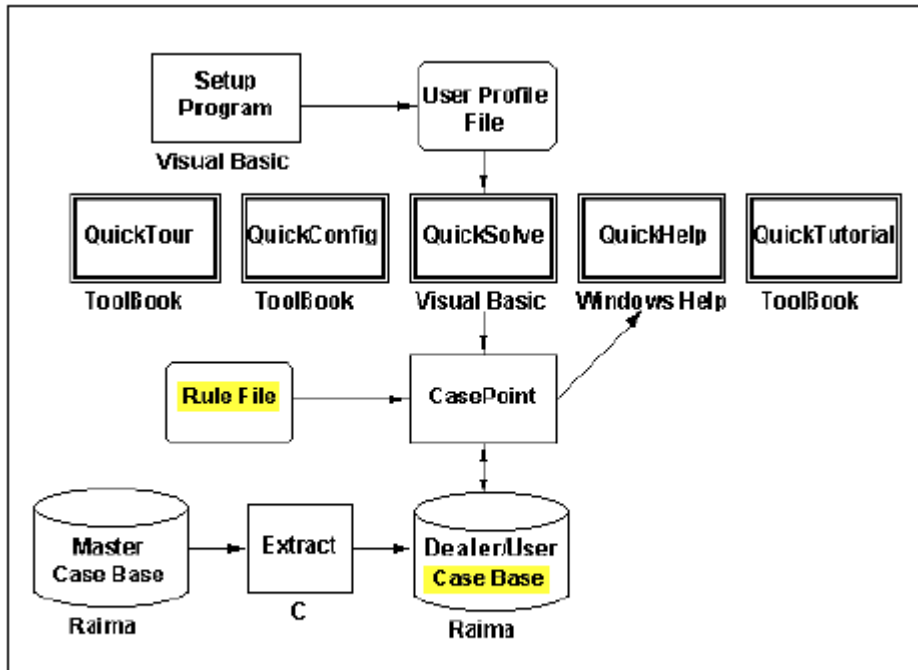
67. Nguyen<sup>6</sup> describes the “QuickSource” system, a help-desk application system for Compaq printers. QuickSource<sup>7</sup> is termed the “second-generation” of the Smart system, a help-desk application used by Compaq’s technical support staff and implemented using the CBR Express engine detailed above. (Nguyen p. 50.) The idea was to take the help-desk system meant for technical support staff and make it accessible to other types of users. Rather than calling Compaq for assistance, the customer can simply use QuickSource to find a solution himself. Smart and QuickSource were developed to function with both CBR-Express as well as CasePoint, a front-end CBR system sold by Inference. (*Id.* at 51.)

---

<sup>5</sup> That is, specify that they do not print on transparencies, as opposed to not mentioning transparencies at all.

<sup>6</sup> T. Nguyen, M. Czerwinski, and D. Lee, “Compaq QuickSource: providing the consumer with the power of AI,” *AI Magazine* 14:3 (1993).

<sup>7</sup> The CBR portion of the QuickSource product is called “QuickSolve.” Since I primarily focus on the CBR portion of QuickSource, I often use the two terms interchangeably. Other portions include “QuickTour, QuickConfig, and QuickTutorial. (Nguyen, p. 52.)



*Figure 1. QUICKSOURCE System Architecture.*

(Highlighting added)

68. QuickSolve stores a set of cases within its case based. Each case contains a “description” field for matching against the electronic message, a “question” field containing questions used to refine the search results, and an “action” field detailing the proposed solution to the problem encapsulated in the case. The case base itself was developed using CBR-Express. (Nguyen, p. 54.)

**Title:** Win, MS Word - Fonts not updated correctly

**Description:**  
H/W - N/A  
HOST - PC  
OPERATING ENVIRONMENT - Windows 3.x  
APPLICATION S/W - MS Word for Windows 1.1a  
SYMPTOMS - The fonts just added in ATM are not available in the Microsoft Word Font menu.

**Questions**

What printer issue do you need help on?  
*Software*

What printer software issue? (pick the first that applies)  
*Application software*

What operating environment are you using?  
*Windows*

What major release of Windows are you running?  
*Windows 3.X*

What is the name of the software you are having an issue with?  
*Microsoft Word for Windows*

What revision of this software are you using?  
*1.1a*

**Actions**

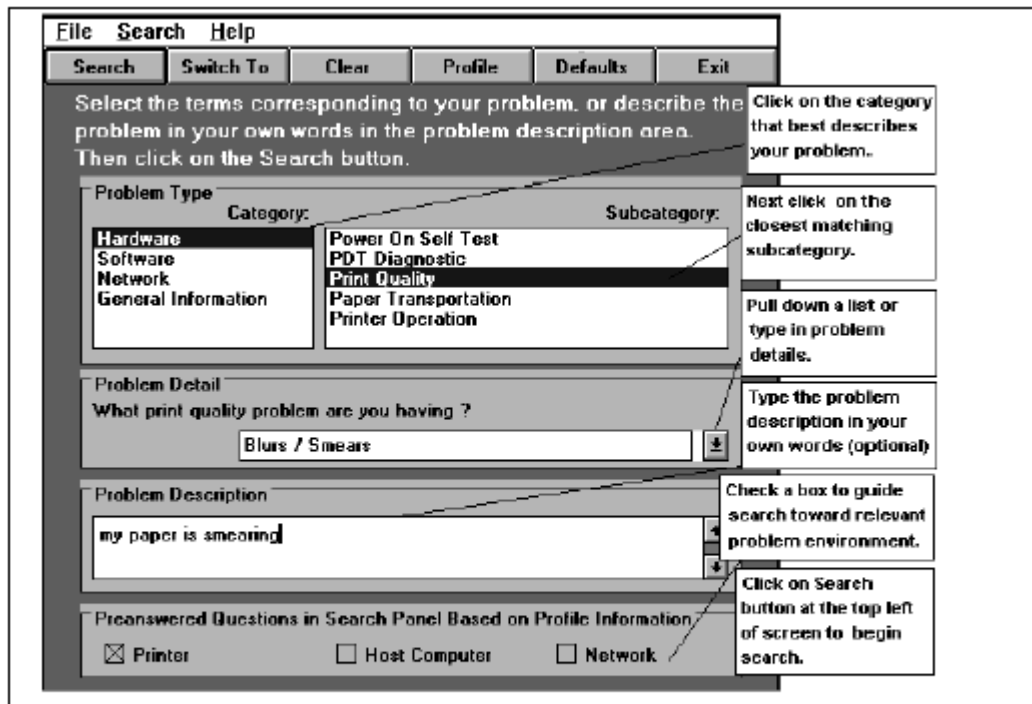
Click on Word for Windows Printer Setup to update the Font selection.

**Action Text**  
Whenever you add or remove fonts from the ATM control screen, and you are using Word for Windows, you will need to perform the following procedures the next time you use the Word for Windows program: Pull down the File pop-down menu and select Printer Setup. Once the Printer Setup dialogue box appears, simply click on the OK button. Doing this forces Word for Windows to scan all available fonts and to associate them with the active printer driver.

*Figure 7. Example of a Software Case from the QUICKSOLVE Case Base.*

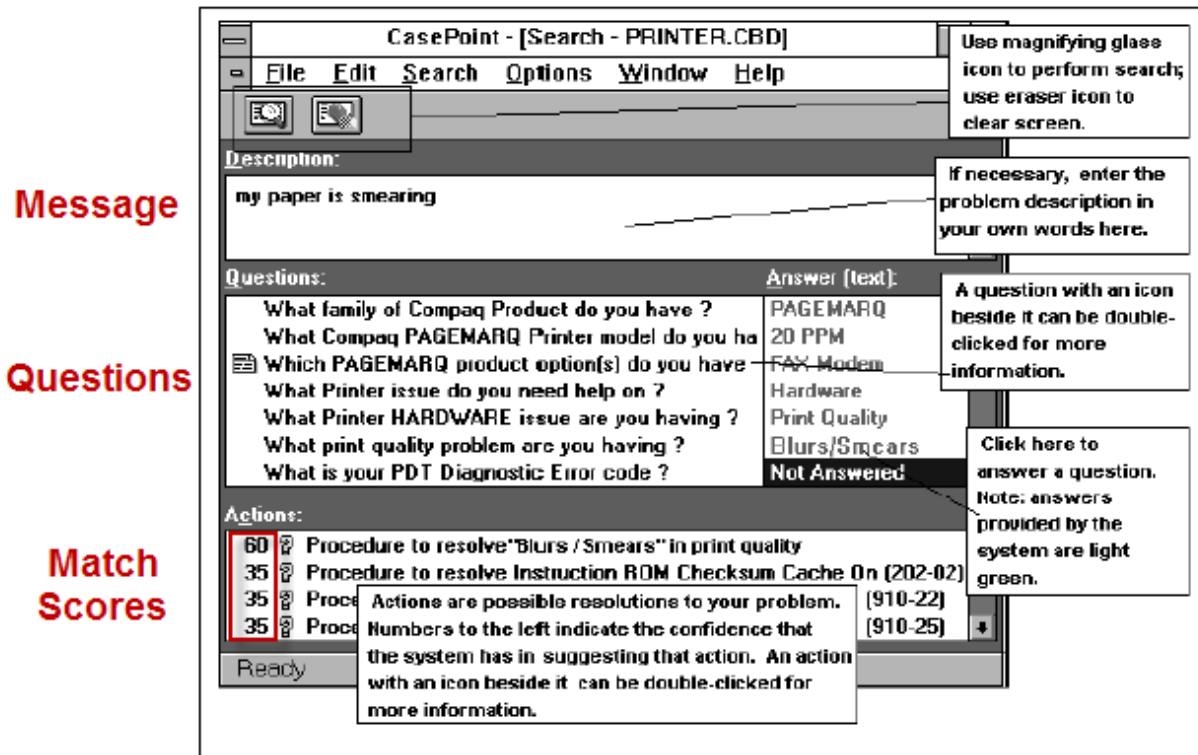
(Highlighting added)

69. Users begin their searches by selecting a category, subcategory, problem detail, and problem description (non-interactive electronic message). QuickSolve then match the test in the message with the text in the description fields of the cases in the case base:



*Figure 9. Screen Example from QUICKSOLVE.*

70. As with CBR Express, QuickSolve presents a list of possible solutions to the user, ordered by match score. QuickSolve also provides a list of questions; each answered question may adjust the match scores to bring more relevant solutions to the top of the list. Thus, as the user fills in the attributes of the presented case via answering questions, Nguyen recomputes match scores by comparing the attributes of the presented case to each of the stored cases in the case base. Different questions have different match weights. (Nguyen 54.) Of note, QuickSolve may “pre-answer” questions based on information stored in the user’s profile (e.g., printer type), as well as information entered in the initial search page. (Nguyen 56.)



*Figure 11. Example of Search Screen with Preanswered Questions.*  
(Annotations in red added)

71. As indicated above, some of the answers to the questions may be pre-entered by QuickSolve. This feature is implemented using rule-based-reasoning, and was added to address user frustration at having to answer multiple questions. (Nguyen 57, 58.) A sample of rules used to pre-answer questions appears below:



```

0 "blurs"
0 "print quality"
=>
1 "Blurs-Smears"
/* If "blurs" and "print quality" strings are detected in the description
field (represented by "0"), then answer the question "What is your print
quality? (question number 1) with "Blurs/Smears". */
34 "3.1"
=>
31 "Windows 3.X"
/* If "3.1" is the answer for the "What version of Windows 3.X are you
using?" question (i.e., question number 34) then answer the "What
major release of Windows are you using?" question (question number
31) with "Windows 3.X". */

```

Figure 12. Examples of Rules Used by QUICKSOLVE to Preanswer Questions.

#### 4. EZ Reader

72. EZ Reader was a system employed by Chase Manhattan bank for automatically classifying, responding to, and/or routing incoming email. The EZ Reader system is described in a paper<sup>8</sup> presented at the 1996 at the Innovative Applications of Artificial Intelligence Conference (IAAI), which consists of case studies of deployed applications with measurable benefits.<sup>9</sup> According to the paper, EZ Reader was deployed in the first quarter of 1996 and handled up to 80% of incoming mail automatically. (Rice 1507.)

---

<sup>8</sup> Amy Rice, Julie Hsu, Angotti Piccolo, Rosanna Piccolo: EZ Reader: Embedded AI for Automatic Electronic Mail Interpretation and Routing, *Proceedings of IAAI'96*, 1507-1517 (1996).

<sup>9</sup> <http://www.aaai.org/Conferences/IAAI/iaai.php>

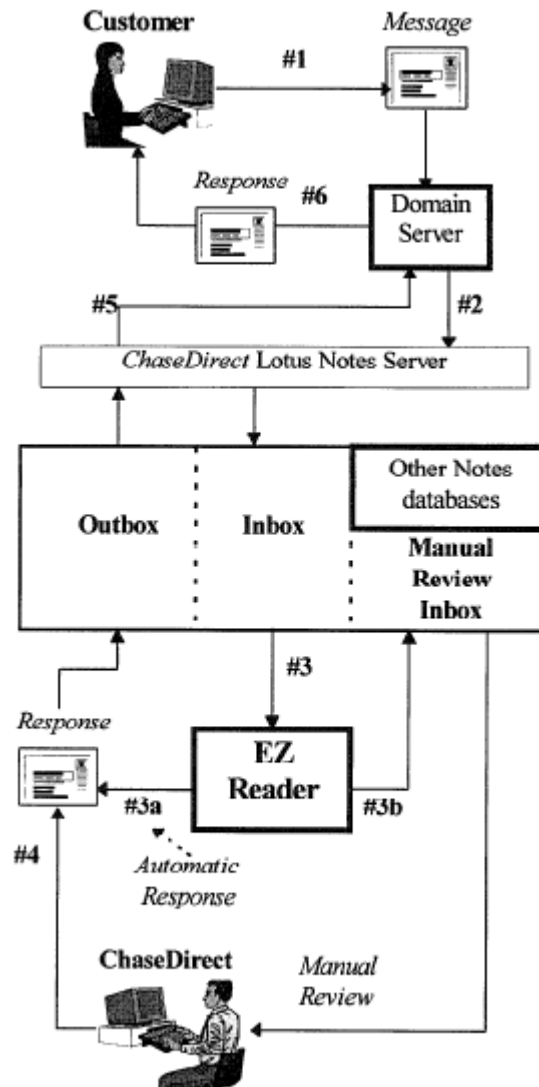


Figure 2. Email Path through ChaseDirect

73. EZ Reader periodically checks the Inbox for new messages. After a customer's message arrives in the Inbox, EZ Reader retrieves the message and interprets it using rule-based and case-based reasoning. If EZ Reader is able to interpret the message to a satisfactory degree, the message is classified as "automatic," and the system creates a response consisting of one or more prepared email ("canned responses") which is then sent back to the customer. If EZ Reader is unable to interpret the message, the message is classified as "referral" or "detected," and the system sends the message to a human reviewer, potentially with one or more suggested replies. (Rice 1509-1511.)

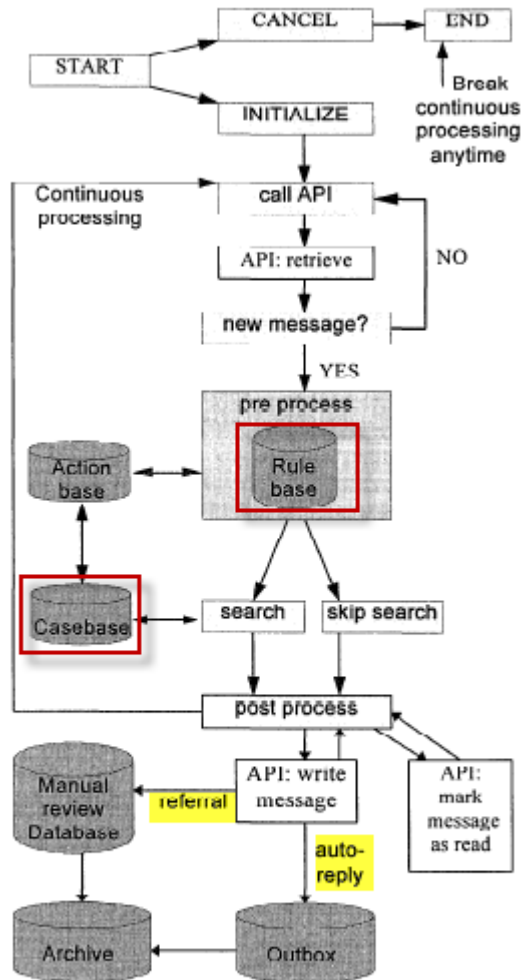


Figure 3. EZ Reader Internal Processing Flow

(Annotations and highlighting added)

74. EZ Reader employs a number of rules to attempt to detect various features within the email message. These features are then set in a presented case model later used by the case-based reasoning system. For instance, EZ Reader attempts to determine whether the email message contains a foreign phone number by looking for certain character strings within the message:

```

RULE foreign-phone
(or
(masked-member$ " +99 " ?message-body)
(masked-member$ " +99-" ?message-body)
(masked-member$ "(+99)" ?message-body)
etc.
)
any other conditions...
=>
(printout "Foreign phone number detected.")
any other actions...

```

75. If the RBR system is unable to classify the system, EZ Reader employs case-based reasoning to try to locate the nearest prior case. The case-based reasoning system is implemented using ART\**Enterprise*, a CBR system originally developed by Inference (the same company that made CBR-Express and CasePoint). EZ Reader matches the text and derived attributes of the incoming email with the text and attributes of the stored cases in the case base and assigns match scores, and uses the cases with the highest match scores. The idea is that whatever response was used to resolve the past case can be used or modified to solve the current case. (Rice 1512.)

```

CASE001:
title = "Sign-Up Kit request; Refer."
subject = "chase direct"
message text = "Please send me a
                ChaseDirect sign-up kit.
                My address is"
address? = "Yes"
action = refer:sign-up-kit,
detected:address,
auto:sign-up-ack

CASE002:
title = "Sign-Up Kit request/no address;
Auto Respond."
subject = "chase direct"
message text = "Please send me a
                ChaseDirect sign-up kit."
action = chase-direct-std

```

76. EZ Reader employs trigram (three-character) matching, similar to Allen and CBR Express. Cases with matching attributes have their match scores increased; cases with mismatching attributes may have their match scores decreased, although EZ Reader defaults to a

mismatch weight of zero. (Rice 1512.) “Since stored cases can contain different numbers of features, a presented case’s raw score is normalized by dividing the raw score by the maximum possible match score for the case.” (*Id.*)

77. EZ Reader is further described in The EZ Reader User's Guide and Reference Manual (“EZ Reader Manual”). As depicted in the manual, EZ Reader retrieves emails from a Lotus Notes server inbox and processes each email by “either automatically respond[ing] to it by placing it a Lotus Notes 'outbox' or by forward[ing] it the ChaseDirect 'inbox' for human review and response” (EZ Reader Manual p. 10).

78. The EZ Reader manual is dated February 5, 1996. (EZ Reader Manual p. 3.) The manual further states that “This document describes EZ Reader, currently in use by the ChaseDirect unit of Chase Manhattan Bank.” (*Id.* 6.)

79. Figure 1 of the EZ Reader Manual depicts an overview of the email handling process (EZ Reader Manual p. 17):

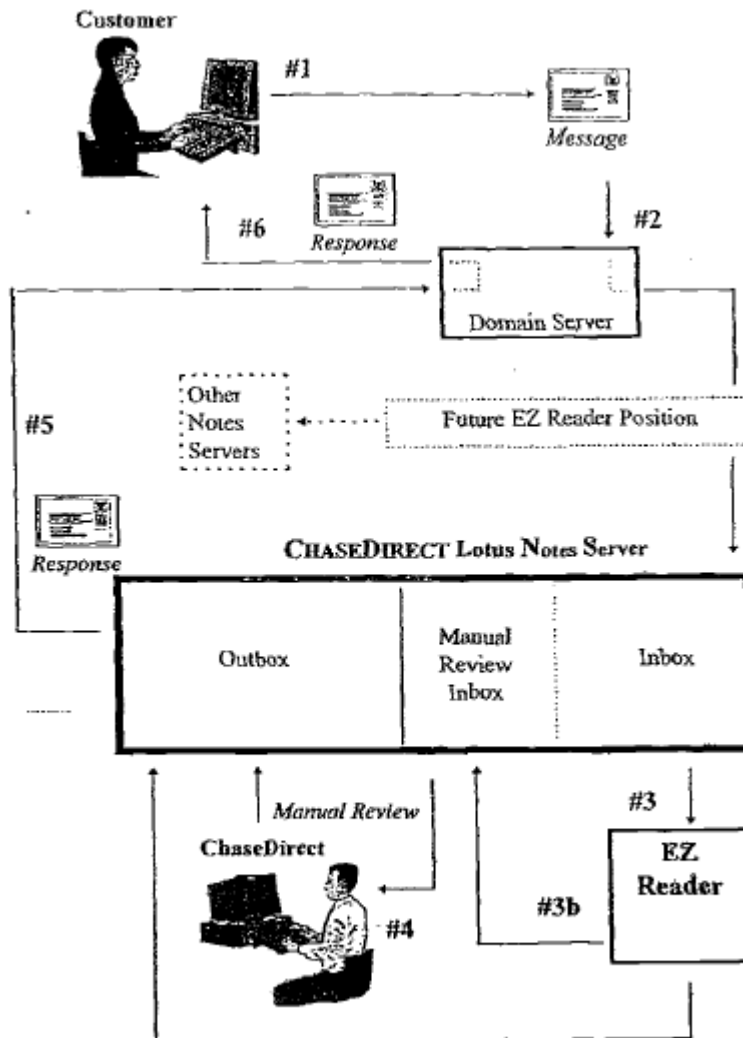


Figure 1: EZ Reader High-Level Architecture

80. The EZ Reader manual also describes each of the numbered steps in Figure 1. (EZ Reader Manual p. 18):
- a. Step 1: The customer sends an email to Chase Manhattan Bank.
  - b. Step 2 (Retrieval): The email is delivered to the Lotus Notes inbox, where it will eventually be detected by EZ Reader.
  - c. Step 3 (Interpretation): EZ Reader compares the message to a library of actual customer messages, categorizes it, and based on the message's category and priority and routes the mail to one or more Lotus Notes mailboxes according to one of two action types:
    - i. Step 3a (Automatic): EZ Reader automatically generates a respond to the mail message, then routes its response directly to the outbox.

ii. Step 3b (Referral): EZ Reader cannot respond to the message, so it routes the message to another inbox for human review. EZ Reader also assigns a priority to the message and suggests a response, based on message type.

d. Step 4 (Response): A human reviewer composes responses to the “referral” messages, and those responses are also routed to the outbox.

e. Step 5: The Lotus Notes server transmits the emails in the outbox

f. Step 6: The customer receives a response to the email.

81. EZ Reader performs Step 3 in the process flow described above: interpreting the message and either responding to it or forwarding it to a human reviewer. This step is implemented using rules and cases:

The knowledgebase portion of EZ Reader, written in the ART\*Enterprise® language, combines case-based analysis and rule-based reasoning to interpret incoming email messages. Rules are used to drive the flow of processing, but also are utilized in a pre-processing phase, to identify and flag certain characteristics of a message. A case-based retrieval is then performed, searching for the best matching case of the current email against the casebase. If any characteristics were tagged in pre-processing phase, they will contribute to the overall casebase score. (EZ Reader Manual p. 19.)

Thus, the case-base matching algorithm compares the characteristics (or features) of the presented case with the characteristics of the stored cases in the case base.

82. The rulebase contains 2 types of rules: phase-processing rules, and question rules. Phase rules, which “are related to the process flow of the system” (EZ Reader Manual p. 32), are set forth in Table 2. For example, one rule triggers the search of the casebase for the best match to the current case (EZ Reader Manual p. 33).

**Table 2: Phase-processing rules**

Rule Name	Phase(s)/event(s) that triggers rule
ez-reader:main:start	none – always fires upon start
ez-reader:main:init-event	(gtk:event activate ergw:pb:start \$?) – fires when Start button is clicked in Greeting Window
ez-reader:main:cancel-event	(gtk:event activate ergw:pb:cancel \$?) – fires when Cancel button is clicked in Greeting Window
ez-reader:mail:retrieve:continuous:api	(phase ready-for-next-email), (continuous-process), (api)
ez-reader:mail:retrieve:continuous	(phase ready-for-next-email), (continuous-process), (no-api)
ez-reader:mail:retrieve>manual	(phase ready-for-next-email), (manual-process)
ez-reader:mail:process:pre	(phase ready-to-preprocess-email)
ez-reader:mail:process:skip-search	(phase ready-to-process-email) and (phase skip-search)
ez-reader:mail:process:search	(phase ready-to-process-email) and (not (phase skip-search))
ez-reader:mail:process:post continuous	(phase ready-to-postprocess-email), (continuous-process)
ez-reader:mail:process:post manual	(phase ready-to-postprocess-email), (manual-process)
ez-reader:main:end	(phase end-of-processing)

83. Questions rules are used for tagging characteristics of, or answering questions about, the current email. Question rules themselves fall into 3 categories (EZ Reader Manual pp. 33-36):

- a. *action-setting rules*, e.g., “Does the message request cancellation? If so, the type is ‘referral.’”
- b. *attribute-setting rules*, e.g. “Does the message mention a foreign country? If so, set the foreign-country attribute to ‘true.’”
- c. *action-and-attribute-setting rules*, e.g. “Does the message mention a specific Chase person? If so, flag the person and set type to ‘referral.’”

**Table 3: Question rules**

Rule name	Question: Does the email/customer...	Type/action
q.address?	reference customer's address?	attribute- and action-setting: detected
q.blank?	contain an empty/blank body?	action-setting: automatic
q.cancel?	request cancellation of services?	action-setting: referral
q.change-of-address?	request a change of customer's address?	action-setting: referral
q.do-not-call-customer?	request specifically not to be called?	attribute- and action-setting: detected



q:fax?	contain a fax number?	action-setting: detected
q:fix-service?	request service or describe an error?	attribute-setting
q:investment?	contain investment references?	attribute-setting
q:foreign?	reference a foreign country?	attribute- and action-setting: detected
q:lengthy?	contain a long message?	attribute-setting
q:macintosh?	reference Macintosh/Apple products?	attribute-setting
q:merger?	reference the Chase/Chemical merger?	attribute-setting
q:multiple?	contain multiple questions?	attribute-setting
q:pay-on-line?	contain references to using Pay-On-Line?	attribute-setting
q:person-referenced?	reference a specific (Chase) person?	attribute- and action-setting: detected
q:phone?	contain a phone number?	action-setting: detected
q:replacement?	reference to a lost or stolen ATM/credit card	action-setting: referral
q:sensitive?	contain information defined as sensitive?	action-setting: referral, auto, and detected

84. Each case in EZ Reader's casebase consists of an actual message with customer-specific information, such as names and addresses, removed (EZ Reader Manual p. 37). A casebase containing at least 200 cases is recommended (EZ Reader Manual p. 39). CBR is used to classify each new message into three general action types (EZ Reader Manual p. 41):

- a. Automatic – No manual review necessary.
- b. Referral – Needs manual review. Referred emails are further classified into 2 categories with one of 4 priorities.
- c. Detected – Information found which matches a pre-specified keyword, phrase, or numbering scheme.

85. The process flow cycle within EZ Reader starts when the “ready-to-preprocess” phase is initiated by receipt of a new message. During this phase, questions rules may fire to set attributes of the current case (EZ Reader Manual p. 34). Phase-processing rules control the progression to the “process-email” phase, in which the casebase is searched for the most similar case, and “postprocess-email” phase in which the appropriate action is taken (EZ Reader Manual p. 32). This rule-controlled process flow is a standard ART\**Enterprise* forward-chaining rule-based reasoning (EZ Reader Manual p. 32, footnote). Similarly, while the case-based reasoning process itself is not explicitly described, it appears to be a standard application of the ART\**Enterprise* case-based reasoning system.

86. The description of EZ Reader's rule-based reasoning and case-reasoning mechanisms (as distinct from the rules and cases themselves) consists of references to ART\**Enterprise* documentation, e.g. “It is strongly recommended that one read and understand the ART\**Enterprise*® documentation (especially for an understanding of rules and case-based reasoning) before attempting to make modifications to the EZ Reader code.” (EZ Reader Manual p. 28.) The manual itself primarily provides information on the creation and maintenance of rules and cases for the specific ChaseBank application following the conventions of ART\**Enterprise*. Thus, EZ Reader appears to be a typical application of ART\**Enterprise* to the kind of business application—automated handling of routine customer messages—for which ART\**Enterprise* was designed.

## 5. GREBE

87. My doctoral dissertation, entitled “Integrating Rules and Precedents for Classification and Explanation: Automating Legal Analysis,” was submitted in May 1991. It describes GREBE (Generator of Exemplar-Based Explanations), a system for legal analysis under Texas worker’s compensation law. (Grebe 5.) GREBE contained a rule base consisting of 57 legal and common-sense rules and a case base containing 35 cases, each of which was a fact pattern drawn from a prior legal case decided under Texas law (Grebe 24-25). GREBE used these rules and cases to determine whether an employee was entitled to worker’s compensation under a given set of facts. The best arguments for and against compensation were returned in the form of a legal memo (GREBE 61-64). My dissertation was and is available through a standard dissertation service (<http://disexpress.umi.com/dxweb>), and dissertations are also all available at the UT graduate library.

88. Suppose, for example, that GREBE is presented with the following new case and is asked whether Jarek is entitled to worker’s compensation:

Jarek was employed as a railroad porter and normally worked from 8:00 A.M. to 5:00 P.M. Because of an unusual work-load, Jarek's employer asked him to work late. Jarek requested and was given permission to walk several blocks

home to tell his wife that he would be working late. He slipped and was seriously injured while walking home. (Grebe 44.)

89. One of the rules in GREBE's rule base was a Texas statute under which an employer is liable to his employee for worker's compensation if the injury is "sustained in the course" of the employee's employment, i.e. if the injury occurred while the employee was "engaged in or about the furtherance of his employer's affairs or business" and the injury "was of a kind and character that had to do with and originated in" the employment. (Grebe 34.) GREBE would use this rule to reason that the employee, Jarak, could recover worker's compensation only if 1) the accident occurred when the employee was engaged in an activity that was furthering his employer's business and 2) that his injury was consistent with his employment. (GREBE 65-66). GREBE would then try to find rules or cases to help it decide these two questions given the facts of the new problem. (Grebe 66).

90. In the case above GREBE, would find another Texas rule that if the injury occurred during traveling, worker's compensation is available only if the employee was "directed in his employment" to travel. There are no rules that say when an employee is "directed in his employment" to travel, but there are example cases. (Grebe 66).

91. One of the cases in GREBE's library, *Vaughn v. Highland Underwriters Ins. Co.*, 445 S.W. 2d 234 (1969), has facts that are an example of being "directed in employment." *Vaughn* has the following fact pattern:

Vaughn worked as a truck driver hauling three loads of sulfur per night from a mine to a factory. Each round trip from the factory to the sulphur mine and back again took approximately 4 hours. Vaughn normally stopped to eat each night at a roadside restaurant during his second return trip to the factory. On the night of the accident, a technical problem at the factory delayed unloading the first load of sulfur. Vaughn's boss told him that to get back on schedule, he would not be able to stop to eat on his second trip, but should instead eat during the delay in unloading the truck. Vaughn therefore set out on his motorcycle toward a nearby restaurant, but was injured in an accident that occurred on the way to the restaurant. (Grebe 31)

92. To show that Jarek's traveling was "directed in his employment" in the same was as Vaughn's travel, GREBE would match the facts and associated relationships in Jarek with the

facts and associated relationships of the cases in the Vaughn case. GREBE assigns equal, predetermined match weights to all facts of a stored case, and uses fractional match scores for partial matches. (Grebe 62.) GREBE then sums the match weights for each matching fact and divides by the maximum possible match weight. (*Id.*) Assuming that there is a good match between Jarek and *Vaughn*, GREBE would then try to reason whether the outcome in *Vaughn* can predict the outcome in Jarek. See, e.g., Fig. 3.9 (Grebe 50):

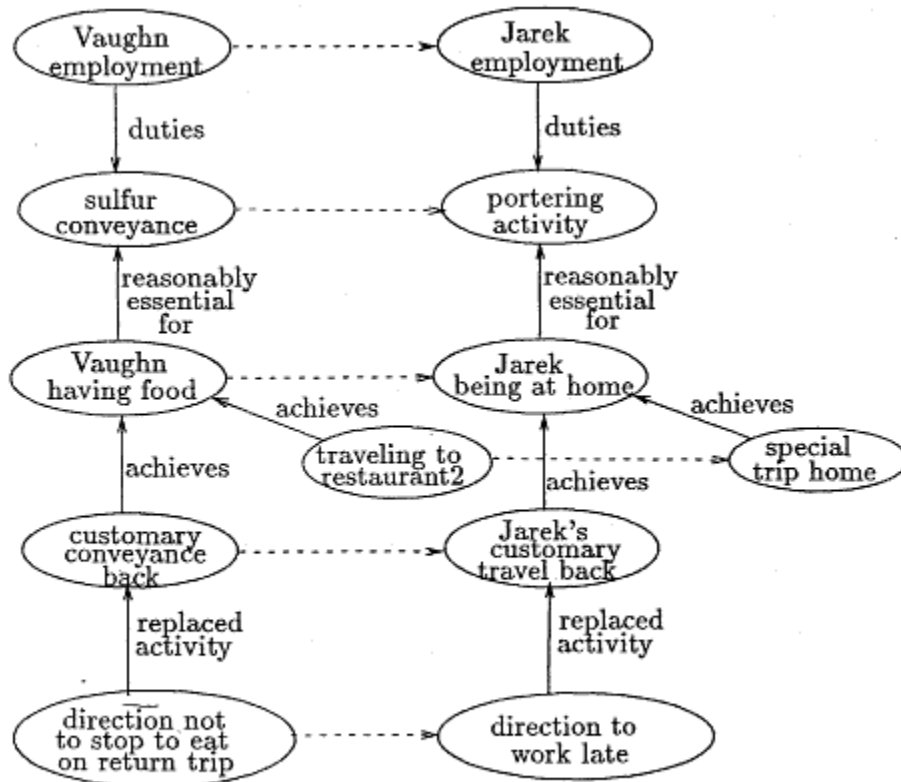


Figure 3.9: The best mapping from the criterial facts of *Vaughn* for “necessitation” to the facts of *Jarek’s Case*. Horizontal dashed arrows represent object pairings.

93. GREBE also used rules to help improve the match between pairs of factual patterns, e.g., by reasoning that walking home and driving home are similar because walking and driving are both kinds of traveling (Grebe 70).

94. In answering queries, GREBE typically combined several case-based reasoning steps, each involving a match between the new case and a factual pattern in the case base, as well as multiple Texas legal rules. (Grebe 61-88). GREBE presented the results in the form of a legal memo. (Grebe 61.) A sample memo in which different aspects of *Jarek’s Case* are matched with three different prior cases, including *Vaughn*, can be found at Grebe 69-74.

## 6. Goodman

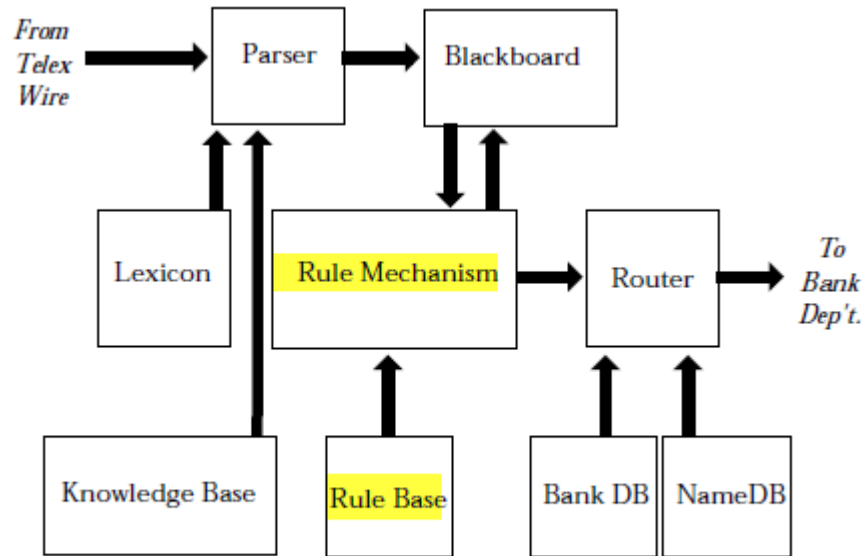
95. Goodman<sup>10</sup> describes the case-based “Prism” system for automated routing of telex<sup>11</sup> communications amongst banks. (Goodman 25.) The Prism system employed rule-base and case-based reasoning to classify and route telexes automatically, thereby increasing response time and cutting down on human involvement. Of note, Prism was implemented at Chase Manhattan Bank—the same organization that implemented the EZ Reader application that forms the basis of the alleged invention. (Goodman 25.) It further appears that some or all of the named inventors of the ‘947 patent were aware of Goodman. *See, e.g.*, Rice 1509: “Other text interpretation applications have successfully used a hybrid approach (Sahin & Sawyer 1989) (Goodman 1991)” (emphasis added). The principle difference lies in the form of the electronic message: in 1990, banks still received many of their electronic messages by telex, whereas by 1996, more electronic messages were received via email.

96. Prism began as a rule-based system for interpreting and routing telexes. The system consisted of approximately 700 rules which semantically parsed the message text and routed the message accordingly. (Goodman 27-28.) While the pure RBR system was fast and accurate, it was both difficult and costly to expand the rule base to deal with new problems. (*Id.* 28.) Accordingly, it was determined that the second version of Prism should employ case-based reasoning. (*Id.*) A representation of the original, costly-to-maintain rule-based version of Prism appears below:

---

10 M. Goodman, Prism: a case-based telex classifier, *Proceedings of IAAI-90*, p. 25-37 (1990).

<sup>11</sup> Telex systems are routed versions of telegrams, and essentially functioned like email systems do today. With the rise of and commercial acceptance of the Internet in the mid-to-late 90s, telex has been largely replaced by email.

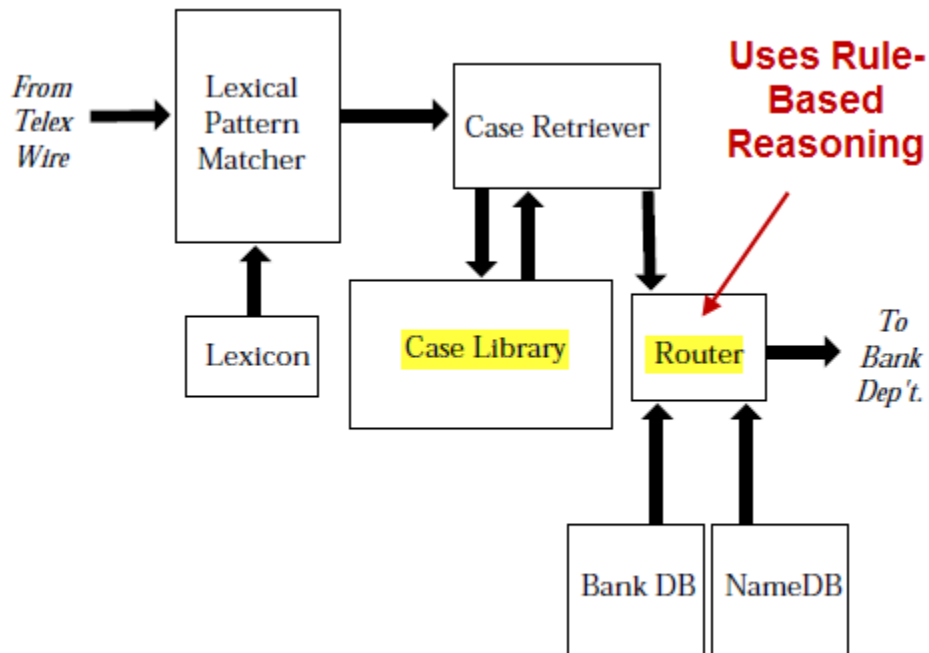


*Figure 1. Rule-Based Prism.*

(Highlighting added)

97. In Case-Based Prism, the system uses a lexical pattern matcher to extract features from the text of the telex, e.g. “Sender,” “Pay,” etc. These attributes form a presented case which is then fed into the CBR module. The presented case and the stored case models contain a number of features of different types. (Goodman 29.) The module then selects the best matches from the case library. Cases are selected using a credit (weight) assignment algorithm that evaluates cases based on a comparison of their features. (Goodman 30.)

98. These retrieved cases are in turn passed into a case adapter, which uses a set of adaptation metrics to compare the problem description with the retrieved cases and "customer-specific rules for extracting additional information from the telex and deciding on the final routing code" (Goodman 31) to adapt solutions to account for any remaining differences from the problem description. The result of this adaptation is a new solution for the incoming problem, which classifies the telex into one of 109 content-based classifications. The classification is then passed on to a rule-based router, which extracts additional information from the telex and determines the final routing code. (Goodman 31.) A depiction of the structure of the final CBR Prism is included below:



*Figure 11. Case-Based Prism*

(Annotation and highlighting added)

99. Goodman receives messages via telex, which functions similarly to email. (Goodman 25-26.) Senders generally do not provide any additional information after the message has been received; thus, the message is non-interactive. Goodman uses a lexical pattern matcher to extract text and attributes from the incoming telex, then creates a presented case based on that telex.

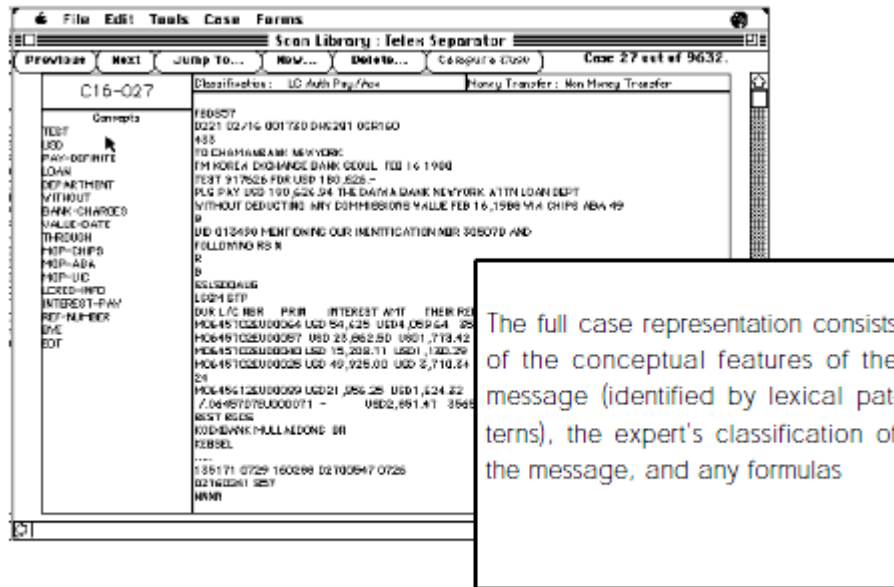


Figure 8. Full Representation of Case.

Goodman then compares the presented case with the stored cases of the case base in order to locate similar cases from the case library:

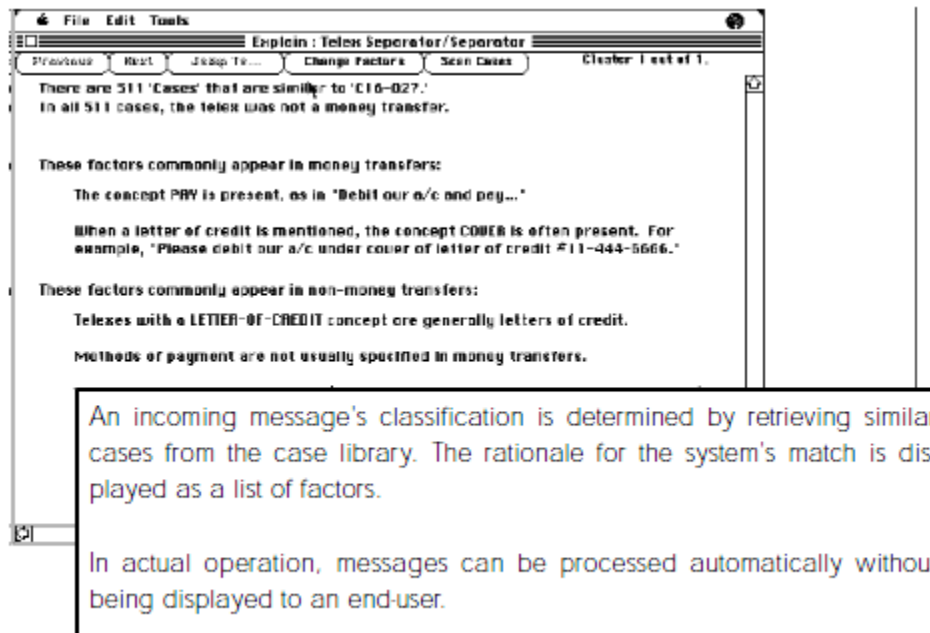


Figure 10. Retrieving Cases.

The retrieved case is used to determine the classification of the incoming message. Goodman then employs a rule-based router, “which contains which contains customer-specific rules for extracting additional information from the telex and deciding on the final routing code.”



(Goodman 31.) After Goodman determines the nearest cases, it extracts a classification for that electronic message based on those near cases:

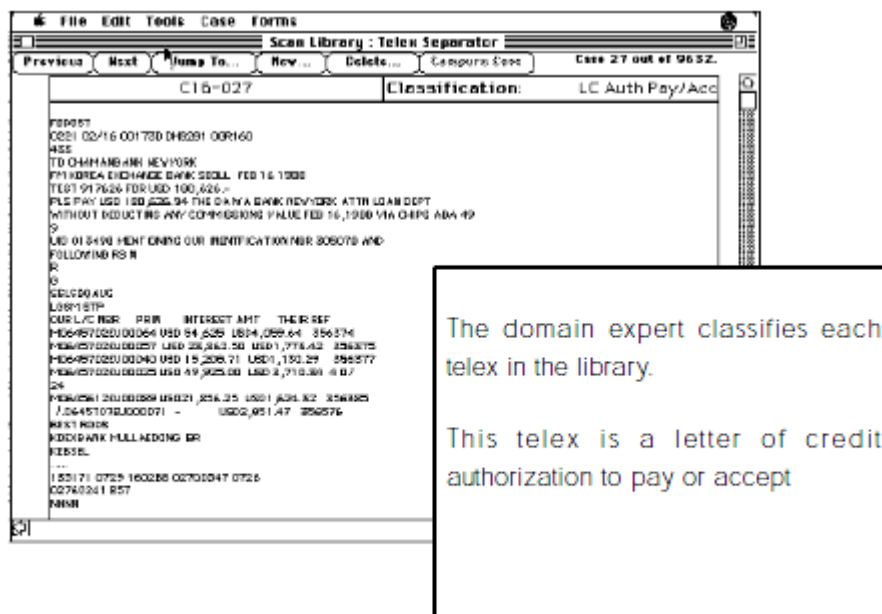


Figure 5. Classification of Messages.

The classification is used to determine how to handle the incoming telex, e.g. as “a letter of credit authorization to pay or accept.”

100. Retrieved cases are passed into a case adapter (Goodman 29, Fig. 2), which uses a set of adaptation metrics to compare the problem description with the retrieved cases and “customer-specific rules for extracting additional information from the telex and deciding on the final routing code” to adapt solutions to account for any remaining differences from the problem description (Goodman 31). Accordingly, the predetermined response—the routing code—may be altered before being used.

## 7. Watson

101. Watson<sup>12</sup> presents a review of CBR practice as of 1994. It includes a history of case-based reasoning, beginning with Roger Schank at Yale University and including

---

12 I. Watson and F. Marir, Case-based reasoning: a review, *The Knowledge Engineering Review*, 9:4, p. 1-34 (1994).

contributions from Janet Kolodner, Bruce Porter, Edwina Rissland, Derek Sleeman, Mike Keane, Michael Richter, Kalus Althoff, Agnar Aamodt, and myself. (Watson 328-330.)

102. Watson sets forth the well-known 4-step CBR cycle consisting of retrieving the most similar case(s), reusing the case(s) to attempt to solve the problem, revising the proposed solution if necessary, and retaining the new solution as part of a new case. (Watson 330.)

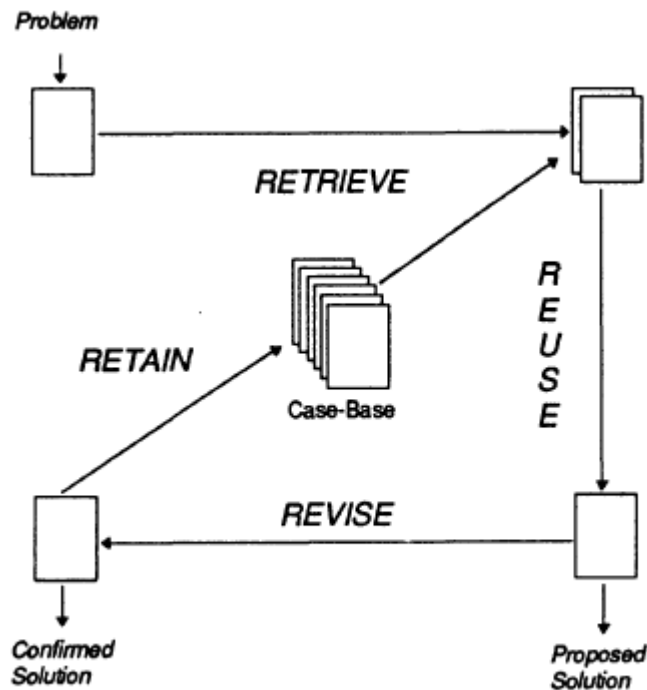


Figure 1 The CBR cycle (adapted from Aamodt & Plaza, 1994)

103. As Watson discloses, “[t]his cycle currently rarely occurs without human intervention. For example, many CBR tools act primarily as case retrieval and reuse systems. Case revision (i.e. adaptation) often being undertaken by managers of the case base. [sic] However, it should not be viewed as a weakness of CBR that it encourages human collaboration in decision support.” (Watson 330.)

104. Prior to retrieval, the problem must first be converted into a *case*, so that it can be compared with the other cases in the case base. As Watson acknowledges, there was no clear consensus on the types of information that should be stored in a case. (Watson 331.) However, cases generally contain the *problem*, the *solution*, and/or the *outcome* of that solution. (See, e.g., references discussed above.)

105. The cases comprising the case base must also be selected and entered into the system. One selection strategy is termed the “category-exemplar model”—essentially, that the cases are selected so as to be a good representation of the types of cases that the CBR system may encounter. The cases contain a number of “features” of the cases, which are usually stored as name-value pairs. For instance, if we were building a case base to determine auto insurance rates, features of the case might include names like “sex,” “age,” “marital status,” etc. Each case (customer) would fill in values for each name, e.g. “sex = male,” “age = 24,” etc. Furthermore, some features would be more important or have greater “weight” than others; for instance, the auto-insurer would likely care more about whether you’d been in any accidents than how many children you have. (Watson 332-333.) These weights become important in the retrieval stage.

106. During retrieval, the CBR system looks for cases that are similar to the instance case. As exact matches are unlikely, CBR systems need to be able to determine how “close” two cases are, with cases that are closest being selected for the adaptation stage (described below). Well-known methods for case retrieval include the “nearest neighbor” algorithm, induction, knowledge guided induction, and template retrieval. (Watson 333.)

$$\frac{\sum_{i=1}^n w_i \times \text{sim}(f_i^I, f_i^R)}{\sum_{i=1}^n w_i}$$

**Figure 2** Nearest neighbour algorithm

107. The nearest neighbor algorithm computes a match score for a stored case by comparing its case features (key-value pairs) with the case features of the presented problem. The result of each comparison is multiplied by the weight or importance of the feature to create a score. After comparing across all features and thus deriving a number of scores, these scores are added together to get the final match score. The nearest neighbor algorithm also normalizes the final match score by dividing by the maximum possible score (i.e., the score when the similarity

function returns “1” for each feature comparison). Thus, all scores are scaled between 0 and 100%, making it easier to compare match scores.

108. As an example, suppose my hypothetical car insurance prediction program was trying to compute a rate for an unmarried 37-year-old male who drives a Toyota Camry, received one speeding ticket in the past year, and lives in Columbus, Ohio. It’s unlikely that the program would have already have someone with those exact characteristics, so it needs to compute match scores for the cases it does have. Suppose one of those cases is a married 34-year-old male driving a Chevy Malibu living in New York City with one speeding ticket. The nearest neighbor algorithm would compare each feature of the two cases. Since the new applicant is unmarried while the existing customer is not, those features are not similar (similarity = 0), and the match score is not affected. However, both drivers are male (similarity = 1), so the match score would increase by the “same gender” amount. The drivers are almost the same age, 34 vs. 37, so the similarity may be 91%, and thus the match score would be increase by 91% of the “same age” amount. And so on. After all the numbers are added together for all of the matching features, the nearest neighbor algorithm divides by the maximum possible match score (i.e., the score when all the similarities are “1”) to obtain the final match percentage.

109. After the closest matches are found, a CBR system may attempt to *adapt* or *revise* the solutions associated with the matched cases to meet the current problem. This adaptation typically occurs through the application of rules. (Watson 334.)

110. In addition to giving an overview of case-base reasoning, Watson also describes popular CBR software tools at the time. CBR-Express is described as “perhaps the most successful CBR product to date.” (Watson 335.) Watson also describes CasePoint, “a runtime version of CBR-Express,” meaning that users cannot add new cases to the regular CBR engine. (*Id.* 336-337; *see also* CBR-Express User’s Manual 6.) Watson further describes Art\*Enterprise, which contains a number of AI paradigms, and relies on the same engine used in CBR-Express. (*Id.* 337.) All three products were developed by Inference Corporation.<sup>13</sup> Other

---

<sup>13</sup> Note also that Inference is the assignee of the Allen patent.

patent. However, case-based reasoning is not related to, but is instead distinct from, logistic regression and gradient descent.

**X. TO THE EXTENT THAT THE ASSERTED CLAIMS ARE READ TO COVER SEARCH QUERIES, THEY ARE INVALID FOR LACK OF ADEQUATE WRITTEN DESCRIPTION.**

276. I have been informed by counsel that to meet the written description requirement, an application must describe an invention, and do so in sufficient detail, that one skilled in the art can clearly conclude that the inventor invented the full scope of the claimed invention as of the filing date sought. I understand the question is not whether a claimed invention is an obvious variant of that which is disclosed in the specification.

277. I am of the opinion that at the time the '947 patent was filed, one of ordinary skill in the art would not understand that the specification described in sufficient detail an invention to receive, interpret, and retrieve one or more responses to an Internet search query, an Internet user's click or a web page, which I understand is what Plaintiff contends meets the non-interactive electronic message limitation in the accused products.

**XI. MATERIALITY OF OMITTED REFERENCES.**

278. As I demonstrate above,<sup>30</sup> the EZ Reader product as described in Rice et al. 1996 and in the EZ Reader User's Guide invalidates all of the asserted claims of the '947 patent because it was in public use in the United States more than one year prior to the date of the patent application.

279. The EZ Reader product is not cumulative of the references that were before the examiner. Rice et al. 1996 discloses the use of a rule base and a case base for electronic message interpretation, which is an element of the '947 patent claim 26. I have examined each of the

---

<sup>30</sup> *see supra*, section VI.D.

references before the Examiner and was unable to find the use of a rule base and case base knowledge engine for electronic message interpretation in any of them.

280. As I demonstrate above,<sup>31</sup> the EZ Reader User's Guide confirms that the EZ Reader product, which invalidates each asserted claim of the '947 patent, was in public use in the first quarter of 1996—more than one year prior to the date of the patent application.

281. As I demonstrate above,<sup>32</sup> the Allen patent invalidates all of the asserted claims of the '947 patent.

282. Allen is not cumulative of the references that were before the examiner. Allen discloses the use of a rule base and a case base for electronic message interpretation, which is an element of the '947 patent claim 26. I have examined each of the references before the Examiner and was unable to find the use of a rule base and case base knowledge engine for electronic message interpretation in any of them.

283. The specification's description of Allen is incomplete and misleading because it fails to acknowledge that Allen discloses not a mere case-based system, but rather a hybrid case-based and rule-based system. ('947 patent 2:41-51; Allen 8:13-18 and Fig. 1, Items 102 and 103.) Presented with the specification's misleading description of Allen, one of skill in the art would not have been prompted to review Allen to determine whether it is an invalidating reference. Rather, it is my opinion that one of skill in the art would have wrongly assumed that Allen does not invalidate the claims of the '947 patent.

## **XII. CONCLUSIONS**

284. None of the Asserted Claims is valid.

285. All the Asserted Claims are anticipated.

---

<sup>31</sup> *see supra*, section VI.D.

<sup>32</sup> *see supra*, sections VI.A and VII.B.1(a).

286. All the Asserted Claims are obvious.

Executed on July 6, 2010, in Columbia, MD.

A handwritten signature in cursive script that reads "Luther Karl Branting". The signature is written in black ink and is positioned above a horizontal line.

---

L. Karl Branting, Ph.D., J.D.