

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE PATENT TRIAL AND APPEAL BOARD

In re U.S. Patent No. 5,944,839

Filed: March 19, 1997

Issued: August 31, 1999

Inventor: Henri J. Isenberg

Assignee: Clouding IP, LLC

Title: System and Method for Automatically Maintaining a Computer
System

Mail Stop PATENT BOARD, PTAB
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

**PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 5,944,839
UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.100 ET SEQ.**

TABLE OF CONTENTS

TABLE OF CONTENTS	ii
EXHIBIT LIST	iv
I. INTRODUCTION	1
II. MANDATORY NOTICES	3
A. Real Party-In-Interest.....	3
B. Related Matters	3
C. Lead and Back-Up Counsel	4
D. Service Information	4
III. PAYMENT OF FEES	4
IV. REQUIREMENTS FOR <i>INTER PARTES</i> REVIEW	5
A. Grounds For Standing	5
B. Identification of Challenge	5
1. The Specific Art and Statutory Ground(s) on Which the Challenge is Based	5
2. How the Construed Claims Are Unpatentable Under the Statutory Grounds Identified in 37 C.F.R. § 42.204(B)(2) and Supporting Evidence Relied upon to Support the Challenge	7
V. FACTUAL BACKGROUND	8
A. Declaration Evidence	8
B. The State of the Art.....	9
C. The ‘573 Patent Application	11
D. The Prosecution History	12
VI. BROADEST REASONABLE CONSTRUCTION	14
A. Sensors	15
B. Artificial Intelligence (AI) Engine.....	16
VII. REPRESENTATIVE PROPOSED REJECTIONS SHOWING THAT PETITIONER HAS A REASONABLE LIKELIHOOD OF PREVAILING	17
A. Claims 1, 2, 6, 8, 14, 15 and 17 are Rendered Obvious by Gurer Under 35 U.S.C. § 103	17
B. Claims 1, 2, 6, 8, 14, 15 and 17 are Rendered Obvious by Allen ‘218 Under 35 U.S.C. § 103.....	34
C. Claims 6, 8 and 14 are Rendered Obvious by Barnett Taken in View of Allen ‘664 Under 35 U.S.C. § 103.....	47

VIII. CONCLUSION	54
CERTIFICATE OF SERVICE	55

EXHIBIT LIST

- 1001 “System and Method for Automatically Maintaining a Computer System,” U.S. Patent No. 5,944,839 to Isenberg, issued August 31, 1999 (*i.e.*, the ‘839 patent) (for *inter partes* review).

- 1002 Excerpts from the Prosecution History of Application No. 08/820,573, which matured into the ‘839 patent.

- 1003 *An Artificial Intelligence Approach to Network Fault Management* by Gurer, D. W., et al. (“Gurer”), published in February 1996.

- 1004 “Autonomous Learning and Reasoning Agent,” U.S. Patent No. 5,586,218 to Allen (“Allen ‘218”), filed May 23, 1994, issued December 17, 1996.

- 1005 “System and Method for Managing Faults in a Distributed System,” U.S. Patent No. 5,664,093 to Barnett et al. (“Barnett”), filed July 25, 1996, issued September 2, 1997.

- 1006 “Case-based Reasoning System,” U.S. Patent No. 5,581,664 to Allen (“Allen ‘664”), filed May 23, 1994, issued December 3, 1996.

- 1007 Declaration of Dr. Todd Mowry, Ph.D.

I. INTRODUCTION

Petitioner Oracle Corporation (“Oracle” or “Petitioner”) respectfully requests *inter partes* review for claims 1, 2, 6, 8, 14, 15 and 17 of U.S. Patent No. 5,944,839 (the “‘839 patent,” attached as Ex. 1001) in accordance with 35 U.S.C. §§ 311–319 and 37 C.F.R. § 42.100 *et seq.*

The ‘839 patent is generally directed to a system and method for automated maintenance of a computer system. More particularly, the ‘839 patent describes a maintenance tool that uses a set of sensors in combination with a case base database to diagnose and solve computer system problems. (*Ex. 1001 at 1:57-59*). If the information gathered by the sensors indicates a problem with the computer system, then the sensors activate an artificial intelligence (AI) engine. (*Id. at Abstract*). The AI engine uses the sensor inputs and information stored in the case base knowledge database to diagnose the likely cause of the problem and determine the best solution. (*Id. at 2:4-5*). If the information necessary to evaluate a case is not in the knowledge database, then the engine activates a sensor to gather additional information. (*Id. at Abstract*). Once the appropriate solution is determined from the data, the AI engine activates the appropriate sensor to perform the repair. (*Id. at 2:9-11*). If the maintenance tool has gathered all the possible data and still does not have a solution to the computer problem, then the AI engine has failed to find a solution in the knowledge database. (*Id. at 4:66 – 5:2*).

Accordingly, the tool saves the state of the computer system and the knowledge database to a location where the state and database can be examined by a human computer expert. (*Id. at 5:3-6*). Presumably, the human expert can then solve the computer problem and add the solution to the knowledge database. (*Id. at 5:7-8*).

As demonstrated by various references which were not before the Examiner, this type of maintenance tool was developed and published several years prior to the earliest claimed priority date. For instance, Gurer describes a system that uses case-based reasoning to automatically diagnose and correct faults in a computer network that it is monitoring. (*Ex. 1003 at 1:10-16*). The raw input to the system is a set of “alarms” which are produced by either the element manager software on a particular network element (e.g., an ATM switch) when it notices a hard error (e.g., a link is down), or through software that performs statistical analysis of the network when it notices a statistical error (e.g., performance degradation due to congestion). (*Id. at 1:30-34*). Gurer further describes how the case-based reasoning system uses its library of cases (i.e., knowledge database) to determine a likely solution to the problem, which potentially involves deciding that the sensors should collect more information regarding the state of the network, and how the gathered information is stored in the knowledge database in the form of new cases. (*Id. at 7:1-10*). Likewise, Allen ‘218 describes a software agent which performs autonomous learning in a real-world environment, implemented in a case-based

reasoning system and coupled to a sensor for gathering information from its environment. (*Ex. 1004 at Abstract*).

As shown below, Gurer, Allen ‘218, and other references render obvious the challenged claims of the ‘839 patent.

II. MANDATORY NOTICES

Pursuant to 37 C.F.R. § 42.8(a)(1), Oracle provides the following mandatory disclosures.

A. Real Party-In-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner certifies that Oracle is the real party-in-interest.

B. Related Matters

Pursuant to 37 C.F.R. § 42.8(b)(2), Petitioner states that the ‘839 Patent is asserted in co-pending litigation captioned *Clouding IP, LLC v. Oracle Corp.*, D.Del., Case No. 1:12-cv-00642. This litigation remains pending. The patents-in-suit are U.S. Patents 6,631,449; 6,918,014; 7,596,784; 7,065,637; 6,738,799; 5,944,839; 5,825,891; 5,678,042; 5,495,607; 7,254,621; 6,925,481. This IPR petition is directed to U.S. Patent 5,944,839; however, petitions corresponding to the remaining patents will be filed in the forthcoming weeks.

C. Lead and Back-Up Counsel

Pursuant to 37 C.F.R. § 42.8(b)(3), Petitioner provides the following designation of counsel: Lead counsel is Greg Gardella (Reg. No. 46,045) and back-up counsel is Scott A. McKeown (Reg. No. 42,866).

D. Service Information

Pursuant to 37 C.F.R. § 42.8(b)(4), papers concerning this matter should be served on the following.

Address: Greg Gardella or Scott McKeown
Oblon Spivak
1940 Duke Street
Alexandria, VA 22314
Email: cpdocketgardella@oblon.com and
cpdocketmckeown@oblon.com
Telephone: (703) 413-3000
Fax: (703) 413-2220

III. PAYMENT OF FEES

The undersigned authorizes the Office to charge \$27,200 to Deposit Account No. 15-0030 as the fee required by 37 C.F.R. § 42.15(a) for this Petition for *inter partes* review. The undersigned further authorizes payment for any additional fees that might be due in connection with this Petition to be charged to the above referenced Deposit Account.

IV. REQUIREMENTS FOR *INTER PARTES* REVIEW

As set forth below and pursuant to 37 C.F.R. § 42.104, each requirement for *inter partes* review of the ‘839 patent is satisfied.

A. Grounds For Standing

Pursuant to 37 C.F.R. § 42.104(a), Petitioner hereby certifies that the ‘839 patent is available for *inter partes* review and that the Petitioner is not barred or estopped from requesting *inter partes* review challenging the claims of the ‘839 patent on the grounds identified herein. The ‘839 patent has not been subject to a previous estoppel based proceeding of the AIA, and, the complaint served on Oracle referenced above in Section II(B) was served within the last 12 months.

B. Identification of Challenge

Pursuant to 37 C.F.R. §§ 42.104(b) and (b)(1), Petitioner requests *inter partes* review of claims 1, 2, 6, 8, 14, 15 and 17 of the ‘839 patent, and that the Patent Trial and Appeal Board (“PTAB”) invalidate the same.

1. The Specific Art and Statutory Ground(s) on Which the Challenge is Based

Pursuant to 37 C.F.R. § 42.204(b)(2), *inter partes* review of the ‘839 patent is requested in view of the following references, each of which is prior art to the ‘839 patent under 35 U.S.C. § 102(a) and/or (e):

(1) *An Artificial Intelligence Approach to Network Fault Management* by Gurer, D. W., et al. (“Gurer,” Ex. 1003) published in February 1996. Gurer is prior art to the ‘839 patent under at least 35 U.S.C. § 102(a).

(2) U.S. Patent No. 5,586,218 to Allen (“Allen ‘218,” Ex. 1004), issued December 17, 1996 from an application filed August 24, 1995. Allen ‘218 is prior art to the ‘839 patent under at least 35 U.S.C. § 102(e).

(3) U.S. Patent No. 5,664,093 to Barnett (Ex. 1005), issued September 2, 1997 from an application filed July 25, 1996. Barnett is prior art to the ‘839 patent under at least 35 U.S.C. § 102(e).

(4) U.S. Patent No. 5,581,664 to Allen (“Allen ‘664,” Ex. 1006), issued December 3, 1996 from an application filed May 23, 1994. Allen ‘664 is prior art to the ‘839 patent under at least 35 U.S.C. § 102(e).

Gurer (Ex. 1003) renders obvious claims 1, 2, 6, 8, 14, 15 and 17 of the ‘839 patent under 35 U.S.C. § 103.

Allen ‘218 (Ex. 1004) renders obvious claims 1, 2, 6, 8, 14, 15 and 17 of the ‘839 patent under 35 U.S.C. § 103.

Barnett (Ex. 1005) in view of Allen ‘664 (Ex. 1006) renders obvious claims 6, 8 and 14 of the ‘839 patent under 35 U.S.C. § 103.

2. How the Construed Claims Are Unpatentable Under the Statutory Grounds Identified in 37 C.F.R. § 42.204(B)(2) and Supporting Evidence Relied upon to Support the Challenge

Pursuant to 37 C.F.R. § 42.204(b)(4), an explanation of how claims 1, 2, 6, 8, 14, 15 and 17 of the '839 patent are unpatentable under the statutory grounds identified above, including the identification of where each element of the claim is found in the prior art, is provided in Section VII, below, in the form of claims charts. Pursuant to 37 C.F.R. § 42.204(b)(5), the appendix numbers of the supporting evidence relied upon to support the challenges and the relevance of the evidence to the challenges raised, including identifying specific portions of the evidence that support the challenges, are provided in Section VII, below, in the form of claim charts.

V. FACTUAL BACKGROUND

A. Declaration Evidence

This Petition is supported by the declaration of Professor Todd C. Mowry from Carnegie Mellon University. (Ex. 1007). Professor Mowry offers his opinion with respect to the content and state of the prior art.

Prof. Mowry is a Professor in Carnegie Mellon's Department of Computer Science, has studied, taught, and practiced in the field of computer science for almost 20 years, and has been a professor of computer science since 1993. (*Id. at* ¶ 1). Prof. Mowry was an Assistant Professor in the ECE and CS departments at the University of Toronto prior to joining Carnegie Mellon University in July, 1997. (*Id. at* ¶ 3). Professor Mowry's research interests span the areas of computer architecture, compilers, operating systems, parallel processing, database performance, and modular robotics. He has supervised 11 Ph.D. students and advised numerous other graduate students. (*Id. at* ¶ 4).

Prof. Mowry has authored over 70 publications and technical reports in the field of computer science. (*Id.*) He is an Associate Editor of ACM Transactions on Computer Systems (TOCS). Further, Prof. Mowry has received a Sloan Research Fellowship and the TR35 Award from MIT's Technology Review. (*Id. at* ¶ 6).

B. The State of the Art

The artificial intelligence (AI) engine disclosed in the '839 patent includes a large database, or case base, of knowledge held by computer experts. This knowledge includes that necessary to diagnose and correct problems with the general operation of the computer system. (*See Ex. 1001 at 1:65 – 2:1*). “Case-based reasoning” (CBR) is an AI problem-solving technique that originated with Roger Schank and his PhD students in the mid-1980s at Yale University. (*Ex. 1007 at ¶ 12*). According to Schank, “[a] case-based reasoner solves new problems by adapting solutions that were used to solve old problems.” (*Id.*) The key word in this quote is “adapting.” (*Id.*) In contrast with *rule-based* reasoning, which performs a scripted action for a rule whenever the specific conditional test for that rule is satisfied, the motivation behind case-based reasoning was to take a more flexible and adaptive approach to problem-solving that draws upon *analogies* to earlier solutions of related (by somewhat different) problems. (*Id.*) The proponents of case-based reasoning argue that drawing upon analogies to solve problems corresponds well to how humans solve problems, i.e., by recalling situations that remind them of their current problem, and by attempting to adapt the previous solution to the current circumstances. (*Id.*)

By 1997, case-based reasoning had become a mature research area that was being actively explored by dozens of research groups around the world. In fact,

there was so much published work on CBR by the early 1990's that multiple survey articles were published on this topic: Kolodner and Aamodt. (*Id.* at ¶ 15). The latter of these two articles, Aamodt, cites 75 papers on CBR. (*Id.*) The Aamodt survey paper also describes “The CBR Cycle” in Section 3.3 and Figure 1, which is the basic structure of nearly all CBR systems, and which has been cited numerous times. (*Id.*)

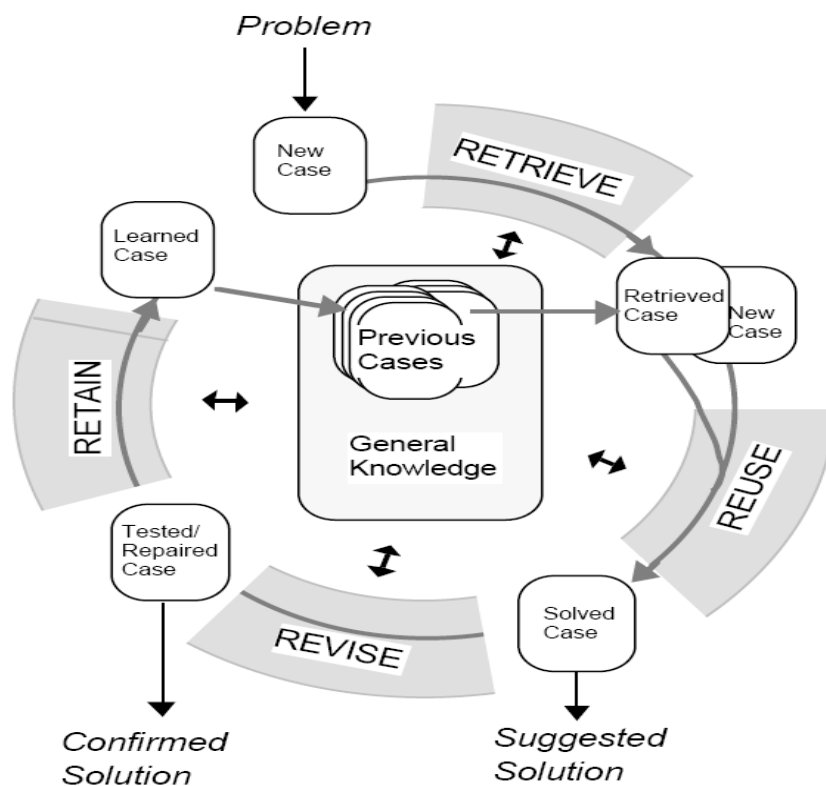


Fig. 1. The CBR Cycle

Conferences and workshops were also being created that were devoted entirely to CBR: the 1st European Workshop on CBR (EWCBR) began in 1993, and the 1st International Conference on CBR (ICCBR) began in 1995. (*Id.*)

C. The ‘573 Patent Application

Application No. 08/820,573 (“the ‘573 application”), which issued as the ‘839 patent, was filed on March 19, 1997.

The ‘573 application describes a system and method for the automated maintenance of a computer system using a set of sensors in combination with a knowledge database to diagnose and solve computer system problems. The maintenance tool performs the following basic steps.

First, a set of sensors act as a monitoring system that monitors the operation of the computer system. When one of the sensors detects a problem, the sensor activates an artificial intelligence (“AI”) engine. (*Ex. 1001 at 1:61-64*).

Second, the AI engine uses the sensor inputs to diagnose the likely cause of the problem and determine the best solution. The AI engine includes a large database, or case base, of knowledge held by computer experts. This knowledge includes that necessary to diagnose and correct problems with the general operation of the computer system. (*Id. at 1:65 – 2:1*).

Third, while processing the problem, the AI engine may reach a point where additional data are needed. If so, the AI engine requests the data from the appropriate sensor or sensors. (*Id. at 2:5-8*).

Fourth, once the appropriate solution is determined from the data, the AI engine activates the appropriate sensor to perform the repair. (*Id. at 2:9-11*).

D. The Prosecution History

In the November 6, 1998 office action claims 1-19 of the '573 application were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,809,493 ("Ahamed") in view of U.S. Patent No. 5,619,656 ("Graf"). In response, in the Amendment dated January 21, 1999, the Patent Owner amended, amongst other claims, independent claims 1 and 7, and added new independent claim 20.

As amended, independent claim 1 recites:

a knowledge database stored in the memory and holding a plurality of cases describing potential computer problems and corresponding likely solutions;...

an AI engine executing on the processor in response to detection of a computer problem and utilizing the plurality of cases to determine a likely solution to the detected computer problem, wherein ...

when the knowledge database does not describe a likely solution to the computer problem, the AI engine saves the gathered data in the knowledge database as a new case.

(Ex. 1002 at January 27, 1999 Amendment, pg. 6).

The Patent Owner summarized the amendments to independent claim 1 as follows:

Thus, claim 1 recites a knowledge database holding cases and an AI engine utilizing the cases. If the knowledge database does not describe a solution to a detected computer problem, the AI engine saves data about the computer in the knowledge database as a new case. An advantage of the claimed behavior is that a human expert can later examine the case and input a likely solution. This limitation was previously recited by claim 3.

(Id. at pg. 7).

The Patent Owner then distinguished Ahamed and Graf on the basis that

“neither Ahamed nor Graf discloses an AI engine utilizing cases as claimed. Moreover, neither Ahamed nor Graf discloses **saving** gathered data in the knowledge database as claimed.” (*Id.*) Further, the Patent Owner stated that “one of ordinary skill in the art would not find saving a state as claimed to be obvious in light of Ahamed and Graf. Neither reference discloses or suggests saving the state of the computer system when a likely solution cannot be determined. Independent claims 7 and 20 recite similar limitations.” (*Id. at pg. 8*).

The Examiner then issued a notice of allowability which did not include a statement of reasons for allowance.

VI. BROADEST REASONABLE CONSTRUCTION

Pursuant to 37 C.F.R. § 42.204(b)(3), the claims subject to *inter partes* review shall receive the “broadest reasonable construction in light of the specification of the patent in which [they] appear[.]” *See also In re Swanson*, No. 07-1534 (Fed. Cir. 2008); *In re Trans Texas Holding Corp.*, 498 F.3d 1290, 1298 (Fed. Cir. 2007) (citing *In re Yamamoto*, 740 F.2d 1569, 1571 (Fed. Cir. 1984)). As the Federal Circuit noted in *Trans Texas*, the Office has traditionally applied a broader standard than a Court does when interpreting claim scope. Moreover, the Office is not bound by any district court claim construction. *Trans Texas*, 498 F.3d at 1297-98, 1301. Rather,

the PTO applies to verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in applicant’s specification. *In re Morris*, 127 F.3d 1048, 1054-55, 44 U.S.P.Q.2d 1023, 1027-28 (Fed. Cir. 1997).

Because the standards of claim interpretation used by the Courts in patent litigation are different from the claim interpretation standards used by the Office in claim examination proceedings (including *inter partes* review), any claim interpretations submitted herein for the purpose of demonstrating a Reasonable Likelihood of Prevailing are neither binding upon litigants in any litigation, nor do such claim

interpretations correspond to the construction of claims under the legal standards that are mandated to be used by the Courts in litigation.

The interpretation of the claims presented either implicitly or explicitly herein should not be viewed as constituting, in whole or in part, Petitioner's own interpretation and/or construction of such claims for the purposes of the underlying litigation. Instead, such constructions in this proceeding should be viewed only as constituting an interpretation of the claims under the "broadest reasonable construction" standard.

All claim terms not specifically addressed below have been accorded their broadest reasonable interpretation in light of the patent specification including their plain and ordinary meaning to the extent such a meaning could be determined by a skilled artisan.

A. Sensors

The '839 patent describes the sensors as follows:

The sensors 112 are software programs that gather information from the computer system 300. (*See, e.g., Ex. 1001 at 3:16-17*).

Accordingly, and as explained in the Declaration of Professor Mowry, the term "sensors" should be interpreted as including, under the broadest reasonable construction, different aspects of the same software program or different components of the same application. (*Ex. 1007 at ¶¶ 21, 22, 24*).

B. Artificial Intelligence (AI) Engine

The term “AI engine” should be interpreted as including, under the broadest reasonable construction, a different aspect of the same software program as the “sensors” discussed above in Section A. Both the sensors and the AI engine can be different components of the same software program or different components of the same application. (*Ex. 1007 at ¶¶ 23-24*).

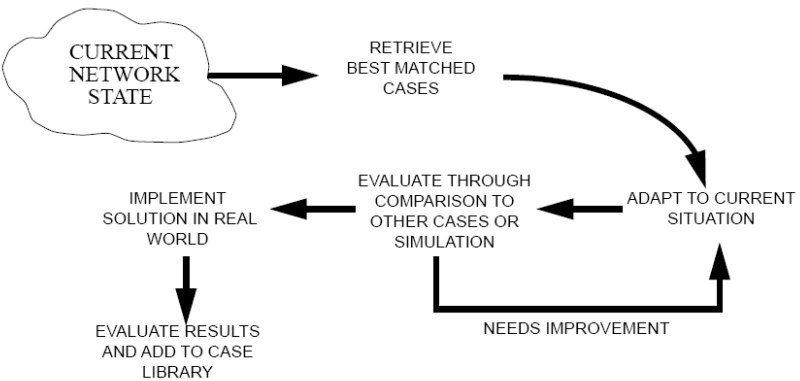
VII. REPRESENTATIVE PROPOSED REJECTIONS SHOWING THAT PETITIONER HAS A REASONABLE LIKELIHOOD OF PREVAILING

The references addressed below each provide the teaching believed by the Examiner to be missing from the prior art and variously anticipate or render obvious the claimed subject matter. It should be understood that rejections may be premised on alternative combinations of, or citations within, these same references.

A. Claims 1, 2, 6, 8, 14, 15 and 17 are Rendered Obvious by Gurer Under 35 U.S.C. § 103

An Artificial Intelligence Approach to Network Fault Management published by Gurer et al. (“Gurer,” Ex. 1003) was not considered during the original prosecution of the ‘839 patent, nor is it cumulative of any prior art considered by the Examiner. Gurer was published in February 1996. The earliest effective priority date of the ‘839 patent is March 19, 1997. Therefore, Gurer is available as prior art against all claims of the ‘839 patent under 35 U.S.C. § 102(a). The following claim chart demonstrates, on a limitation-by-limitation basis, how claims 1, 2, 6, 8, 14, 15 and 17 of the ‘839 patent are rendered obvious by Gurer under 35 U.S.C. § 103(a). (See *Ex. 1007 at ¶¶ 25-48*).

US 5,944,839 Claim Language	Correspondence to Gurer
1. A tool for automatically maintaining a computer system	Gurer discloses that “[a]utomation of network management activities can benefit from the use of artificial intelligence (AI)”

US 5,944,839 Claim Language	Correspondence to Gurer
<p>having a processor and a memory, the tool comprising:</p>	<p>technologies, including fault management, performance analysis, and traffic management. Here we focus on fault management, where the goal is to proactively diagnose the cause of abnormal network behavior and to propose, and if possible, take corrective actions.” (See <i>Ex. 1003 at 1:11-14</i>).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the network management system of Gurer would necessarily have a processor and a memory associated therewith. (<i>Ex. 1007 at ¶¶ 28-29</i>).</p>
<p>a knowledge database stored in the memory and holding a plurality of cases describing potential computer problems and corresponding likely solutions</p>	<p>Gurer discloses a knowledge database (“case library”) holding a plurality of cases describing potential computer problems and corresponding likely solutions.</p> <p>“Case-based reasoning is based on the premise that situations recur with regularity. Studies of experts and their problem solving techniques have found that experts rely quite strongly on applying their previous experiences to the current problem at hand. CBR can be thought of as such an expert that applies previous experiences stored as cases in a <i>case library</i>. Thus, the problem-solving process becomes one of recalling old experiences and interpreting the new situation in terms of those old experiences.” (<i>Ex. 1003 at 6:35-39</i>).</p>  <pre> graph TD A([CURRENT NETWORK STATE]) --> B[RETRIEVE BEST MATCHED CASES] B --> C[ADAPT TO CURRENT SITUATION] C --> D[EVALUATE THROUGH COMPARISON TO OTHER CASES OR SIMULATION] D --> E[IMPLEMENT SOLUTION IN REAL WORLD] E --> F[EVALUATE RESULTS AND ADD TO CASE LIBRARY] F -- "NEEDS IMPROVEMENT" --> C </pre> <p>Figure 3: The case-based reasoning process.</p>

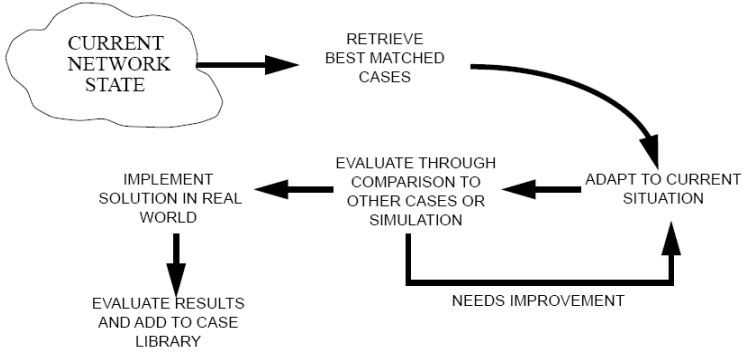
US 5,944,839 Claim Language	Correspondence to Gurer
	<p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the knowledge database (or “case library” of Gurer, which is part of the case-based reasoning system) would necessarily be stored in a memory of the computer system. (<i>Ex. 1007 at ¶¶ 30-31</i>).</p>
<p>a plurality of sensors stored in the memory and executing on the processor and adapted for gathering data about the computer system, storing the data in the knowledge database, and detecting whether a computer problem exists from the data and the plurality of cases; and</p>	<p>Gurer discloses a plurality of sensors (“alarms”) adapted for gathering data about the computer system, storing the data in the knowledge database, and detecting whether a computer problem exists from the data and the plurality of cases.</p> <p>“The first step in fault management is to collect <i>monitoring and performance alarms</i>. Typically <u>alarms are produced by either managed network elements</u> (e.g., ATM switches, customer premise equipment) <u>or by a statistical analysis of the network that monitors trends and threshold crossings</u>. Alarms can be classified into two categories, physical and logical, where physical alarms are hard errors (e.g., a link is down), typically reported through an element manager, and logical alarms are statistical errors (e.g., performance degradation due to congestion).” (<i>Ex. 1003 at 1</i>).</p> <p>“Alarm filtering is a process that analyzes the multitude of alarms received and eliminates the redundant alarms (e.g., multiple occurrences of the same alarm). Alarm correlation is the interpretation of multiple alarms such that new conceptual meanings can be assigned to the alarms, creating derived alarms. Faults are identified by analyzing the filtered and correlated alarms and by requesting tests and status updates from the element managers, which provide additional information for diagnosis.” (<i>Id. at 2:1-</i></p>

US 5,944,839 Claim Language	Correspondence to Gurer
	<p>5).</p> <p>“The more complex processes of fault management include alarm filtering and correlation, fault identification, and correction. Many of these functions involve analysis, correlation, pattern recognition, clustering or categorization, problem solving, planning, and <i>interpreting data from a knowledge base</i> that contains descriptions of network elements and topology.” (<i>Id. at 3:2-4</i>).</p> <div data-bbox="638 779 1369 1350" data-label="Diagram"> <pre> graph TD NETWORK((NETWORK)) --> PHYSICAL[PHYSICAL ALARMS] NETWORK --> LOGICAL[LOGICAL ALARMS] PHYSICAL --> COLLECT[COLLECT ALARMS] LOGICAL --> COLLECT COLLECT --> TAKEACTION[TAKE ACTION to ENSURE CUSTOMER SATISFACTION] TAKEACTION --> FILTER[FILTER and CORRELATE ALARMS (Neural Networks or Bayesian Belief Networks)] FILTER --> DIAGNOSE[DIAGNOSE FAULTS (Case-based Reasoning or Expert Systems)] DIAGNOSE --> CORRECTIVE[DEVELOP and IMPLEMENT CORRECTIVE PLAN (Case-based Reasoning, Intelligent Planning, or Expert Systems)] CORRECTIVE --> VERIFY[VERIFY FAULT IS ELIMINATED] VERIFY -- "Fault is Eliminated" --> RECORD[RECORD EVENTS and ANALYZE FAULT MANAGEMENT PERFORMANCE] VERIFY -- "Fault not Eliminated" --> COLLECT </pre> <p>The flowchart illustrates a fault management process. It begins with a 'NETWORK' cloud at the top, which generates 'PHYSICAL ALARMS' and 'LOGICAL ALARMS'. These alarms are collected in a 'COLLECT ALARMS' box. From there, the process moves to 'TAKE ACTION to ENSURE CUSTOMER SATISFACTION', then to 'FILTER and CORRELATE ALARMS (Neural Networks or Bayesian Belief Networks)', and then to 'DIAGNOSE FAULTS (Case-based Reasoning or Expert Systems)'. The next step is 'DEVELOP and IMPLEMENT CORRECTIVE PLAN (Case-based Reasoning, Intelligent Planning, or Expert Systems)', followed by 'VERIFY FAULT IS ELIMINATED'. If the fault is eliminated, the process proceeds to 'RECORD EVENTS and ANALYZE FAULT MANAGEMENT PERFORMANCE'. If the fault is not eliminated, the process loops back to 'COLLECT ALARMS'.</p> </div> <p>Figure 1: The fault management process and possible AI technologies.</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the network elements and network system of Gurer, which generate the “alarms,” would necessarily be associated with a processor and a memory therewith. (<i>Ex. 1007 at ¶¶ 32-35</i>). Further, Prof. Mowry explains why one skilled in the art would understand that the collected alarms are stored somewhere, e.g., in the knowledge database, because fault identification and correction involve interpreting data from a knowledge base that contains descriptions of network</p>

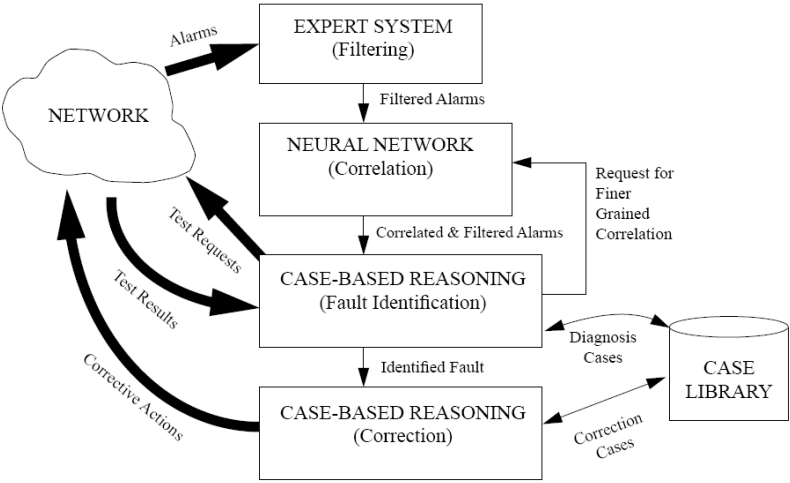
US 5,944,839 Claim Language	Correspondence to Gurer
<p>an AI engine stored in the memory and executing on the processor in response to detection of a computer problem and [the AI engine] utilizing the plurality of cases to determine a likely solution to the detected computer problem,</p>	<p>elements (which produce the alarms) and topology. (<i>Id.</i>)</p> <p>Gurer discloses an AI system executing in response to detection of a computer problem. Gurer discloses:</p> <p>“Automation of network management activities can benefit from the use of artificial intelligence (AI) technologies, including fault management, performance analysis, and traffic management. Here we focus on fault management, where the goal is to proactively diagnose the cause of abnormal network behavior and to propose, and if possible, take corrective actions . . . AI technologies may be used to automate the fault management process, in particular neural networks (NNs) and case-based reasoning (CBR).” (<i>Ex. 1003 at 1:11-16</i>).</p> <p>Further, Gurer discloses that:</p> <p>“[a]nother area of fault management where AI technologies can have a positive impact, is fault correction. CBR systems, ESs [Expert Systems], or intelligent planning systems can develop plans or courses of action that will correct a fault that has been identified and verified.” (<i>Id. at 3:27-29</i>).</p> <div data-bbox="609 1320 1390 1797" data-label="Diagram"> <pre> graph TD NETWORK((NETWORK)) -- Alarms --> EXPERT[EXPERT SYSTEM
(Filtering)] EXPERT -- Filtered Alarms --> NEURAL[NEURAL NETWORK
(Correlation)] NEURAL -- Correlated & Filtered Alarms --> CBR1[CASE-BASED REASONING
(Fault Identification)] CBR1 -- Identified Fault --> CBR2[CASE-BASED REASONING
(Correction)] CBR2 -- Corrective Actions --> NETWORK CBR1 -- Test Requests --> NETWORK NETWORK -- Test Results --> CBR1 CBR2 -- Request for Finer Grained Correlation --> NEURAL CBR2 -- Diagnosis Cases --> CASELIB[(CASE LIBRARY)] CASELIB -- Correction Cases --> CBR2 </pre> </div> <p>Figure 4: A hybrid AI system for automated network fault management.</p>

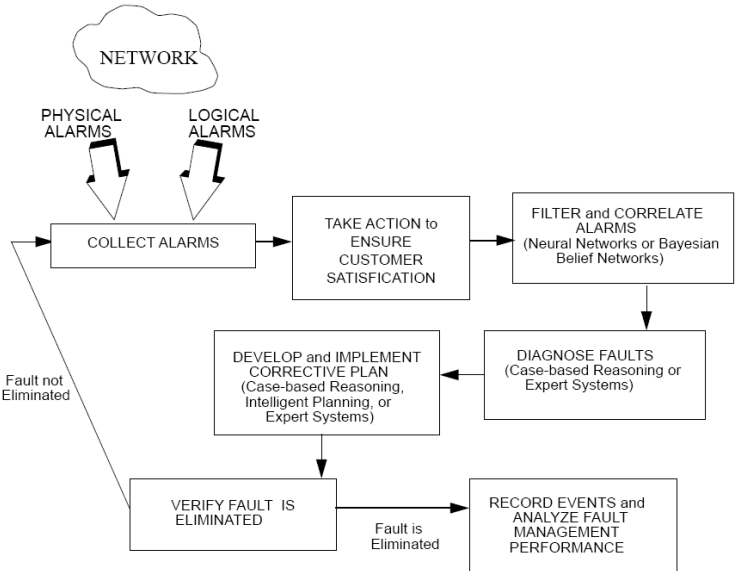
US 5,944,839 Claim Language	Correspondence to Gurer
	<p>See also the five-step CBR problem solving process shown at Ex. 1003, p. 7.</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the AI engine of Gurer would necessarily be stored in a memory of the computer system. (<i>Ex. 1007 at ¶¶ 36-38</i>).</p>
<p>wherein when the knowledge database lacks data necessary to determine a likely solution to the computer problem, the AI engine activates a particular sensor in the plurality of sensors to gather the necessary data and store the data in the knowledge database, and</p>	<p>Gurer discloses</p> <p>“The filtering and correlation of alarms is the first step of fault diagnosis. The second step involves further analysis and identification of the exact cause of the alarms, or the fault. This process is an iterative one where <i>alarm data are analyzed and decisions are made whether more data should be gathered</i>, a finer grained analysis should be executed, or problem solving should be performed. <i>Gathering more data can consist of sending tests to network elements or requesting network performance data.</i>” (<i>Ex. 1003 at 6:18-22</i>).</p> <p>Gurer further discloses that “[t]ypically alarms are produced by . . . network elements (e.g., ATM switches, customer premise equipment).” (<i>Id. at 1:30-31</i>).</p>
<p>wherein when the knowledge database does not describe a likely solution to the computer problem, the AI engine saves the gathered data in the knowledge database as a new case.</p>	<p>Gurer discloses that</p> <p>“[T]he front-end of an application that uses CBR will have a graphical user interface (GUI) that allows a user to easily input new cases without requiring any detailed understanding of the underlying CBR engine . . . A key benefit in using CBR systems is their ability to learn. One obvious method of learning is to add new cases with their new solutions and evaluations to the case library.” (<i>Ex. 1003 at 6:43-47</i>).</p> <p>“Once a fault is identified, the whole problem</p>

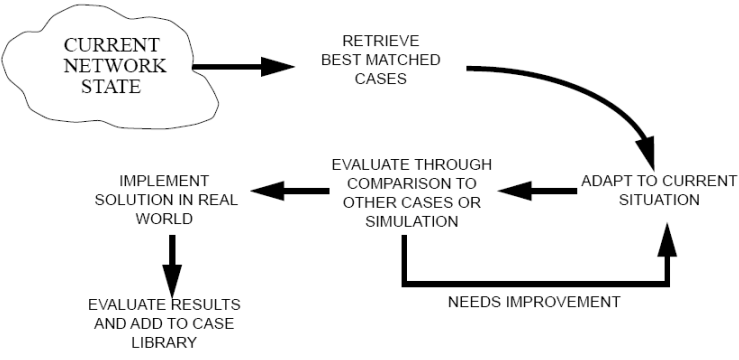
US 5,944,839 Claim Language	Correspondence to Gurer
	<p>solving episode should be analyzed. The value of the tests (i.e., useful, not useful), the steps taken, any circuitous paths or dead-ends taken, and the success of the analysis should be stored into a new case. This information, in addition to contextual information, comprise a new case which is then indexed into the case library.” (<i>Id. at 8:6-8</i>).</p>
<p>2. The tool of claim 1, wherein each case comprises:</p> <p>at least one question asking about a particular aspect of the computer system that can be answered by the data gathered by the plurality of sensors; and</p>	<p>The ‘839 patent discloses that “[q]uestions belong to one or four categories: Yes/No; Numeric; Text; and List. Yes/No questions are those that have Yes or No answers. Numeric questions are those having answers that are integers. Text questions have textual answers. Finally, List questions have answers selected from a list of legal answers.” (<i>See Ex. 1001 at 3:52-57</i>).</p> <p>Gurer discloses that</p> <p>“CBR problem solving can be depicted as a five-step process, as shown in Figure 3: (1) retrieval, (2) interpretation and adaptation, (3) evaluation and repair, (4) implementation, and (5) evaluation and learning. The first step is retrieving cases that best match the current situation or case. Thus it is crucial to use an appropriate indexing method, such as <i>decision trees</i> or nearest neighbor matching. Once a case is retrieved, it must be interpreted and then adapted. The interpretation process is a simple comparison between the retrieved cases and the current case. Adaptation is a complicated, domain-dependent process that uses rules to adapt the current case to the problem situation and propose an initial solution, based on the similarities and differences . . . ” (<i>Ex. 1003 at 7:1-6</i>).</p>

US 5,944,839 Claim Language	Correspondence to Gurer
	 <p data-bbox="857 657 1162 678">Figure 3: The case-based reasoning process.</p> <p data-bbox="560 737 1430 947">The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the “decision trees” used in the CBR problem solving process of Gurer would include at least one question about a particular aspect of the computer system. (<i>Ex. 1007 at ¶¶ 39-41</i>).</p>
<p data-bbox="186 953 521 1199">at least one action describing a likely solution to a potential computer problem stored in the knowledge database.</p>	<p data-bbox="560 953 1414 1073">The ‘839 patent discloses that “[a]n action is a statement that indicates a solution to a computer problem.” (<i>See Ex. 1001 at 3:58-59</i>).</p> <p data-bbox="560 1121 1386 1199">Gurer discloses that a case indexed into the case library includes:</p> <p data-bbox="667 1247 1422 1587">“Once a fault is identified, the whole problem solving episode should be analyzed. The value of the tests (i.e., useful, not useful), the steps taken, any circuitous paths or dead-ends taken, and the success of the analysis should be stored into a new case. This information, in addition to contextual information, comprise a new case which is then indexed into the case library.” (<i>Ex. 1003 at 8:6-8</i>).</p> <p data-bbox="560 1635 1430 1837">Gurer also states that “[a]daptation is a complicated, domain-dependent process that uses rules to adapt the current case to the problem situation and propose an initial solution, based on the similarities and differences...” (<i>Ex. 1003 at 7</i>)</p>
<p data-bbox="186 1850 418 1881">6. A method of</p>	<p data-bbox="560 1850 857 1881">Gurer discloses that</p>

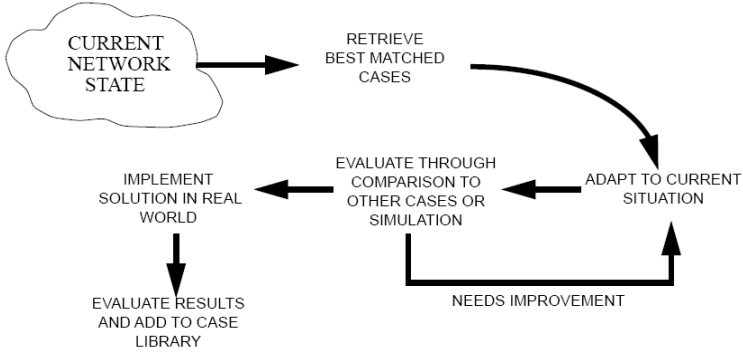
US 5,944,839 Claim Language	Correspondence to Gurer
optimizing a computer system, the method comprising the steps of:	<p>“[a]utomation of network management activities can benefit from the use of artificial intelligence (AI) technologies, including fault management, performance analysis, and traffic management. Here we focus on fault management, where the goal is to <i>proactively diagnose the cause of abnormal network behavior and to propose, and if possible, take corrective actions.</i>” (Ex. 1003 at 1:11-14).</p>
detecting a problem in the computer system;	<p>Gurer discloses that</p> <p>“[a]utomation of network management activities can benefit from the use of artificial intelligence (AI) technologies, including fault management, performance analysis, and traffic management. Here we focus on fault management, where the goal is to proactively diagnose the cause of <i>abnormal network behavior</i> and to propose, and if possible, take corrective actions.” (Ex. 1003 at 1:11-14).</p> <p>See also discussion in connection with claim 1.</p>
activating an AI engine in response to the problem detection;	<p>Gurer discloses an AI system executing in response to detection of a computer problem. Gurer discloses:</p> <p>“[a]utomation of network management activities can benefit from the use of artificial intelligence (AI) technologies, including fault management, performance analysis, and traffic management. Here we focus on fault management, where the goal is to proactively diagnose the cause of abnormal network behavior and to propose, and if possible, take corrective actions . . . AI technologies may be used to automate the fault management process, in particular neural networks (NNs) and case-based reasoning (CBR).” (Ex. 1003 at 1:11-16).</p> <p>Further, Gurer discloses that:</p>

US 5,944,839 Claim Language	Correspondence to Gurer
	<p>“[a]nother area of fault management where AI technologies can have a positive impact, is fault correction. CBR systems, ESs [Expert Systems], or intelligent planning systems can develop plans or courses of action that will correct a fault that has been identified and verified.” (<i>Id. at 3:27-29</i>).</p>  <p>Figure 4: A hybrid AI system for automated network fault management.</p>
utilizing, by the AI engine, selected ones of a plurality of sensors to gather information about the computer system;	<p>Gurer discloses sensors (“alarms”) adapted for gathering information about the computer system.</p> <p>“The first step in fault management is to collect <i>monitoring and performance alarms</i>. Typically <u>alarms are produced by either managed network elements (e.g., ATM switches, customer premise equipment) or by a statistical analysis of the network that monitors trends and threshold crossings</u>. Alarms can be classified into two categories, physical and logical, where physical alarms are hard errors (e.g., a link is down), typically reported through an element manager, and logical alarms are statistical errors (e.g., performance degradation due to congestion).” (<i>Ex.</i></p>

US 5,944,839 Claim Language	Correspondence to Gurer
	<p>1003 at 1:30-34).</p> <p>“Alarm filtering is a process that analyzes the multitude of alarms received and eliminates the redundant alarms (e.g., multiple occurrences of the same alarm). Alarm correlation is the interpretation of multiple alarms such that new conceptual meanings can be assigned to the alarms, creating derived alarms. Faults are identified by analyzing the filtered and correlated alarms and by requesting tests and status updates from the element managers, which provide additional information for diagnosis.” (<i>Id.</i> at 2:1-5).</p>  <pre> graph TD NETWORK((NETWORK)) --> PHYSICAL[PHYSICAL ALARMS] NETWORK --> LOGICAL[LOGICAL ALARMS] PHYSICAL --> COLLECT[COLLECT ALARMS] LOGICAL --> COLLECT COLLECT --> ACTION[TAKE ACTION to ENSURE CUSTOMER SATISFACTION] ACTION --> FILTER[FILTER and CORRELATE ALARMS (Neural Networks or Bayesian Belief Networks)] FILTER --> DIAGNOSE[DIAGNOSE FAULTS (Case-based Reasoning or Expert Systems)] DIAGNOSE --> PLAN[DEVELOP and IMPLEMENT CORRECTIVE PLAN (Case-based Reasoning, Intelligent Planning, or Expert Systems)] PLAN --> VERIFY[VERIFY FAULT IS ELIMINATED] VERIFY -- "Fault is Eliminated" --> RECORD[RECORD EVENTS and ANALYZE FAULT MANAGEMENT PERFORMANCE] VERIFY -- "Fault not Eliminated" --> COLLECT </pre> <p>Figure 1: The fault management process and possible AI technologies.</p> <p>See also discussion in connection with claim 1.</p>
<p>determining, by the AI engine, a likely solution to the problem from the gathered information; and</p>	<p>Gurer discloses that</p> <p>“CBR problem solving can be depicted as a five-step process, as shown in Figure 3: (1) retrieval, (2) interpretation and adaptation, (3) evaluation and repair, (4) implementation, and (5) evaluation and</p>

US 5,944,839 Claim Language	Correspondence to Gurer
	<p>learning. The first step is retrieving cases that best match the current situation or case. Thus it is crucial to use an appropriate indexing method, such as decision trees or nearest neighbor matching. Once a case is retrieved, it must be interpreted and then adapted. The interpretation process is a simple comparison between the retrieved cases and the current case. Adaptation is a complicated, domain-dependent process that uses rules to adapt the current case to the problem situation and propose an initial solution, based on the similarities and differences. The next step is an evaluation and repair cycle where the proposed solution is evaluated through comparisons to cases with similar solutions or through simulation, and the solution is modified accordingly. After the CBR system has found its best solution, the solution is implemented and the results are evaluated. The resulting evaluation, solution steps, and problem context are entered into a new case, which is then indexed into the case library, allowing the system to learn.” (Ex. 1003 at 7:1-10).</p>  <pre> graph TD A([CURRENT NETWORK STATE]) --> B[RETRIEVE BEST MATCHED CASES] B --> C[ADAPT TO CURRENT SITUATION] C --> D[EVALUATE THROUGH COMPARISON TO OTHER CASES OR SIMULATION] D --> E[IMPLEMENT SOLUTION IN REAL WORLD] E --> F[EVALUATE RESULTS AND ADD TO CASE LIBRARY] F -- "NEEDS IMPROVEMENT" --> C </pre> <p>Figure 3: The case-based reasoning process.</p>
when a likely solution cannot be determined, saving a state of the computer system.	<p>Gurer discloses that</p> <p>“Once a fault is identified, the whole problem solving episode should be analyzed. The value of the tests (i.e., useful, not useful), the steps taken,</p>

US 5,944,839 Claim Language	Correspondence to Gurer
	<p>any circuitous paths or dead-ends taken, and the success of the analysis should be stored into a new case. This information, in addition to contextual information, comprise a new case which is then indexed into the case library.” (Ex. 1003 at 8:6-8).</p>
<p>8. The method of claim 6, wherein the determining step comprises the substeps of:</p> <p>inferring the likely solution to the problem from questions, actions, and rules contained in a knowledge database;</p>	<p>Gurer discloses that</p> <p>CBR problem solving can be depicted as a five-step process, as shown in Figure 3: (1) retrieval, (2) interpretation and adaptation, (3) evaluation and repair, (4) implementation, and (5) evaluation and learning. The first step is retrieving cases that best match the current situation or case. Thus it is crucial to use an appropriate indexing method, such as <i>decision trees</i> or nearest neighbor matching. Once a case is retrieved, it must be interpreted and then adapted. The interpretation process is a simple comparison between the retrieved cases and the current case. <i>Adaptation is a complicated, domain-dependent process that uses rules to adapt the current case to the problem situation and propose an initial solution, based on the similarities and differences.</i> The next step is an evaluation and repair cycle where the proposed solution is evaluated through comparisons to cases with similar solutions or through simulation, and the solution is modified accordingly. After the CBR system has found its best solution, the solution is implemented and the results are evaluated. (Ex. 1003 at 7:1-10).</p>

US 5,944,839 Claim Language	Correspondence to Gurer
	 <p data-bbox="857 657 1161 678">Figure 3: The case-based reasoning process.</p>
<p>wherein the AI engine utilizes the selected ones of the plurality of sensors to gather information when the knowledge database lacks information necessary to answer a question.</p>	<p>Gurer discloses</p> <p>“The filtering and correlation of alarms is the first step of fault diagnosis. The second step involves further analysis and identification of the exact cause of the alarms, or the fault. This process is an iterative one where <i>alarm data are analyzed and decisions are made whether more data should be gathered</i>, a finer grained analysis should be executed, or problem solving should be performed. <i>Gathering more data can consist of sending tests to network elements or requesting network performance data.</i>” (Ex. 1003 at 6: 18-22).</p> <p>Gurer notes earlier that “[t]ypically alarms are produced by . . . network elements (e.g., ATM switches, customer premise equipment).” (Id. at 1: 30-31).</p>
<p>14. The method of claim 6, wherein the detecting step comprises the steps of:</p> <p>periodically activating selected ones of the plurality of sensors to</p>	<p>Gurer discloses that</p> <p>“Gathering more data can consist of sending tests to network elements or requesting network performance data.” (Ex. 1003 at 6:18-22).</p> <p>Gurer notes earlier that</p>

US 5,944,839 Claim Language	Correspondence to Gurer
gather information about the computer system; and	<p>“[t]ypically alarms [or <i>sensors</i>] are produced by . . . network elements (e.g., ATM switches, customer premise equipment).” (<i>Id. at 1:30-31</i>).</p> <p>Further,</p> <p>“Faults are identified by analyzing the filtered and correlated alarms and by requesting tests and status updates from the <i>element managers</i>, which provide additional information for diagnosis.” (<i>Id. at 2:4-5</i>).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the fault management system of Gurer would periodically activate at least one of the alarms or element managers (another type of sensor) to gather information about the computer system, particularly when requesting status updates from the element managers. (<i>Ex. 1007 at ¶¶ 45-46</i>).</p>
analyzing the information to determine whether a problem exists.	As described with respect to claim 6, <i>supra</i> , Gurer teaches analyzing the information gathered from the plurality of sensors (or alarms) to determine whether a problem exists.
<p>15. A program storage device readable by a computer system, the program storage device tangibly embodying a program of instructions executable by the computer system to perform method steps for automatically maintaining the computer system, the program storage device holding instructions for:</p>	<p>Claim 15 is nearly coextensive in scope with claim 1, the only differences being (1) form over substance, <i>i.e.</i>, claim 1 recites a tool whereas claim 15 recites a program storage device for automatically maintaining a computer system, and (2) the limitation reciting “sensing information about the computer system by <i>at least one sensor</i>” in claim 15 versus a <i>plurality of sensors</i> in claim 1.</p> <p>The method of Gurer may be readily executed on a computer readable storage medium. Because the scope of claim 1 necessarily includes all of the limitations of claim 15, Petitioner’s arguments with respect to claim 1 apply equally to claim 15.</p> <p>Accordingly, for at least the same reasons as set forth with respect to claim 1, claim 15 is rendered obvious over</p>

US 5,944,839 Claim Language	Correspondence to Gurer
	Gurer under 35 U.S.C. § 103(a).
sensing information about the computer system by at least one sensor;	See claim 1, limitation reciting “a plurality of sensors stored in the memory and executing on the processor and adapted for gathering data about the computer system . . . ,” <i>supra</i> .
determining whether a computer problem exists from the sensed information;	See claim 1, limitation reciting “a plurality of sensors stored in the memory and executing on the processor and adapted for . . . detecting whether a computer problem exists from the data and the plurality of cases,” <i>supra</i> .
searching a plurality of cases with the sensed information to determine whether a likely solution to the computer problem exists;	See claim 1, limitation reciting “a knowledge database stored in the memory and holding a plurality of cases describing potential computer problems and corresponding likely solutions,” <i>supra</i> .
when additional information is needed to determine a likely solution to the computer problem, activating the at least one sensor to sense the additional information; and	See claim 1, limitation reciting “wherein when the knowledge database lacks data necessary to determine a likely solution to the computer problem, the AI engine activates a particular sensor in the plurality of sensors to gather the necessary data and store the data in the knowledge database,” <i>supra</i> .
when a likely solution to the computer problem does not exist, saving the sensed information as a new case of the plurality of cases.	See claim 1, limitation reciting “wherein when the knowledge database does not describe a likely solution to the computer problem, the AI engine saves the gathered data in the knowledge database as a new case,” <i>supra</i> .
17. The program storage device of claim 15, wherein the sensing step comprises the substep of:	Gurer discloses that “Gathering more data can consist of sending tests to network elements or requesting network performance data.” (<i>Ex. 1003 at 6:18-22</i>).

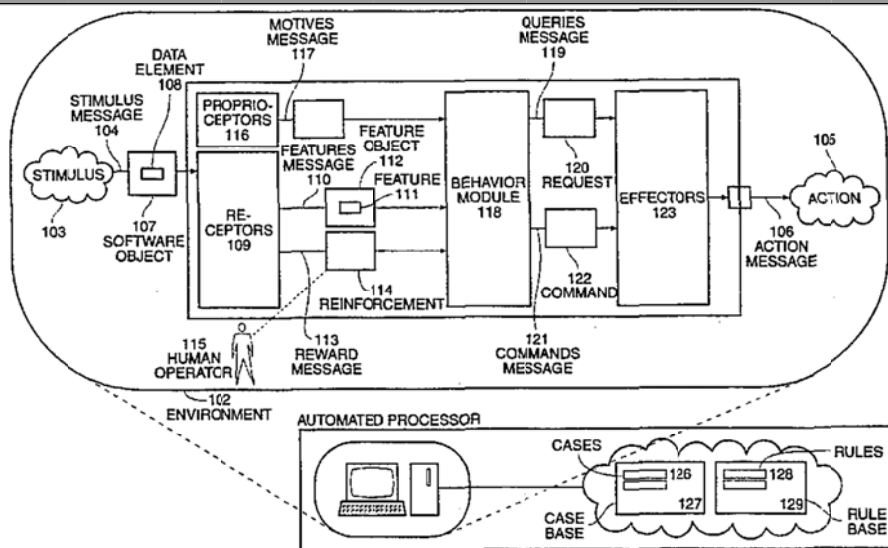
US 5,944,839 Claim Language	Correspondence to Gurer
<p>periodically activating at least one sensor to gather information about the computer system.</p>	<p>Gurer notes earlier that</p> <p>“[t]ypically alarms [or <i>sensors</i>] are produced by . . . network elements (e.g., ATM switches, customer premise equipment).” (<i>Id. at 1:30-31</i>).</p> <p>Further,</p> <p>“Faults are identified by analyzing the filtered and correlated alarms and by requesting tests and status updates from the <i>element managers</i>, which provide additional information for diagnosis.” (<i>Id. at 2:4-5</i>).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the fault management system of Gurer would periodically activate at least one of the alarms or element managers (another type of sensor) to gather information about the computer system, particularly when requesting status updates from the element managers. (<i>Ex. 1007 at ¶¶ 45-46</i>).</p>

**B. Claims 1, 2, 6, 8, 14, 15 and 17 are Rendered Obvious by
Allen ‘218 Under 35 U.S.C. § 103**

U.S. Patent No. 5,586,218 to Allen (“Allen ‘218,” Ex. 1004) was not considered during the original prosecution of the ‘839 patent, nor is it cumulative of any prior art considered by the Examiner. Allen ‘218 was filed on August 24, 1995 and issued on December 17, 1996. The earliest effective priority date of the ‘839 patent is March 19, 1997. Therefore, Allen ‘218 is available as prior art to the ‘839 patent under 35 U.S.C. § 102(e).

The following claim chart demonstrates, on a limitation-by-limitation basis, how claims 1, 2, 6, 8, 14, 15 and 17 of the ‘839 patent are rendered obvious by Allen ‘218 under 35 U.S.C. § 103(a). (*See Ex. 1007 at ¶¶ 49-65*).

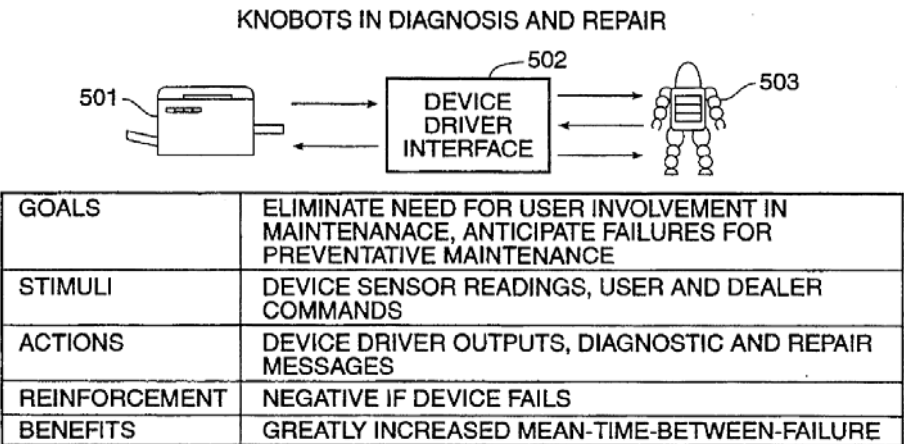
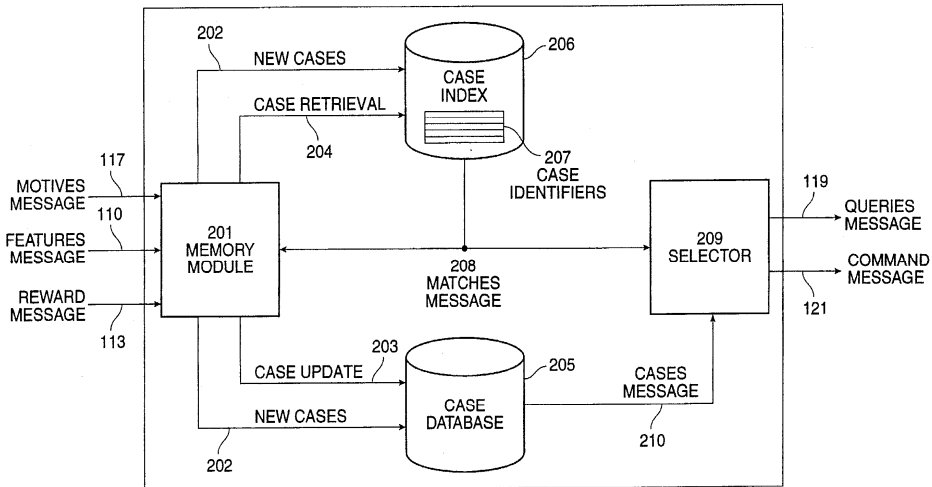
US 5,944,839 Claim Language	Correspondence to Allen ‘218
1. A tool for automatically maintaining a computer system having a processor and a memory, the tool comprising:	Allen ‘218 discloses “A software agent which performs autonomous learning in a real-world environment, implemented in a case-based reasoning system and coupled to a sensor for gathering information from, and to an effector for manipulating, its environment.” (<i>See Ex. 1004 at Abstract</i>). “In a preferred embodiment, the environment 102 may comprise either a software environment or a physical environment, or some combination thereof.” (<i>Id. at 4:5-7</i>).



In a preferred embodiment, the software agent **101** may be implemented with an automated processor **124**, which may execute a software inference engine **125** for reasoning using a set of cases **126** in a case base **127** and a set of rules **128** in a rule base **129**. In a preferred embodiment, the processor **124** may comprise a system having a processor, memory comprising a stored program, memory comprising data, and input/output devices, as is well known in the art. The operation and software structures of this system are described herein in terms of their functions and at a level of detail which would be clear to those of ordinary skill in the art. It would be clear to anyone of ordinary skill in the art, after perusal of the specification, drawings and claims herein, that modification and/or programming (using known programming techniques) of a processor of known design to achieve these functions would be a straightforward task and would not require undue experimentation.

(*Id.* at 5:8-24).

Furthermore, Allen '218 explicitly discusses the fact that the sensor gathers information from its environment, which, in the case of the preventative maintenance example illustrated in FIG. 5 of Allen '218, is the computer system within the office equipment, e.g., printer, photocopier, that it is monitoring. (*Id.* at 7:51-52). Allen '218 describes that the purpose of this example tool is "to anticipate failures of the device 501 so that

US 5,944,839 Claim Language	Correspondence to Allen '218										
	<p>preventative maintenance may be performed so as to reduce failures and to increase meantime-between-failure.” (<i>Id. at</i> 7:67-8:3).</p> <div data-bbox="527 457 1425 898"> <p style="text-align: center;">KNOBOTS IN DIAGNOSIS AND REPAIR</p>  <table border="1" data-bbox="527 646 1425 898"> <tr> <td>GOALS</td><td>ELIMINATE NEED FOR USER INVOLVEMENT IN MAINTENANACE, ANTICIPATE FAILURES FOR PREVENTATIVE MAINTENANCE</td></tr> <tr> <td>STIMULI</td><td>DEVICE SENSOR READINGS, USER AND DEALER COMMANDS</td></tr> <tr> <td>ACTIONS</td><td>DEVICE DRIVER OUTPUTS, DIAGNOSTIC AND REPAIR MESSAGES</td></tr> <tr> <td>REINFORCEMENT</td><td>NEGATIVE IF DEVICE FAILS</td></tr> <tr> <td>BENEFITS</td><td>GREATLY INCREASED MEAN-TIME-BETWEEN-FAILURE</td></tr> </table> </div> <p style="text-align: center;">FIG.5</p>	GOALS	ELIMINATE NEED FOR USER INVOLVEMENT IN MAINTENANACE, ANTICIPATE FAILURES FOR PREVENTATIVE MAINTENANCE	STIMULI	DEVICE SENSOR READINGS, USER AND DEALER COMMANDS	ACTIONS	DEVICE DRIVER OUTPUTS, DIAGNOSTIC AND REPAIR MESSAGES	REINFORCEMENT	NEGATIVE IF DEVICE FAILS	BENEFITS	GREATLY INCREASED MEAN-TIME-BETWEEN-FAILURE
GOALS	ELIMINATE NEED FOR USER INVOLVEMENT IN MAINTENANACE, ANTICIPATE FAILURES FOR PREVENTATIVE MAINTENANCE										
STIMULI	DEVICE SENSOR READINGS, USER AND DEALER COMMANDS										
ACTIONS	DEVICE DRIVER OUTPUTS, DIAGNOSTIC AND REPAIR MESSAGES										
REINFORCEMENT	NEGATIVE IF DEVICE FAILS										
BENEFITS	GREATLY INCREASED MEAN-TIME-BETWEEN-FAILURE										
<p>a knowledge database stored in the memory and holding a plurality of cases describing potential computer problems and corresponding likely solutions</p>	<p>Allen '218 describes a knowledge database (“case database 205”) stored in the memory and holding a plurality of cases describing potential computer problems and corresponding likely solutions.</p> <div data-bbox="511 1245 1437 1728">  </div> <p style="text-align: center;">FIG. 2</p>										

US 5,944,839 Claim Language	Correspondence to Allen ‘218
	<p>In a preferred embodiment, the case database 205 may comprise a limited number of stored cases 126, the contents of the case database 205 being determined by a genetic technique for producing, evaluating and selecting the cases 126, such as a genetic technique like that disclosed with FIG. 3. Thus, while the case database 205 might store less than all of the cases 126 which the autonomous agent 101 has encountered, it might maintain a set of cases 126 which provides a preferred model of the environment, such as those cases 126 which allow the agent 101 to distinguish variant problem scenarios and to act autonomously and intelligently in those variant problem scenarios.</p> <p>(See <i>Ex.1004 at 5:49-60</i>).</p> <p>Further,</p> <p>The case database 205 may comprise its current set of cases 126, each of which may comprise a features element 301, which may generally indicate when the case 126 may be useful, and an action element 302, which may indicate the action 105 to take and an evaluation 303 of that action 105.</p> <p>(<i>Id. at 6:11-15</i>).</p>
a plurality of sensors stored in the memory and executing on the processor and adapted for gathering data about the computer system, storing the data in the knowledge database, and detecting whether a computer problem exists from the data and the plurality of cases; and	<p>Allen ‘218 discloses “[a] software agent [101] which performs autonomous learning in a real-world environment, implemented in a case-based reasoning system and coupled to a sensor for gathering information from . . . its environment.”</p> <p>(See <i>Ex.1004 at Abstract</i>).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the software agent 101 of Allen ‘218 could be coupled to a plurality of sensors, instead of just a single sensor, for gathering information from the environment, as this represents nothing more than a simple substitution of one sensor with a plurality of sensors to predictably arrive at the claimed invention. (<i>Ex. 1007 at ¶¶ 51-53</i>).</p>
an AI engine stored in the memory and	<p>Allen ‘218 discloses an AI engine (“software agent 101 with software interference engine 125”) executing on the processor in response to detection of a computer problem.</p>

US 5,944,839 Claim Language	Correspondence to Allen '218
<p>executing on the processor in response to detection of a computer problem and [the AI engine] utilizing the plurality of cases to determine a likely solution to the detected computer problem,</p>	<p>Allen '218 further discloses</p> <p style="padding-left: 40px;">In a preferred embodiment, the software agent 101 may be implemented with an automated processor 124, which may execute a software inference engine 125 for reasoning using a set of cases 126 in a case base 127 and a set of rules 128 in a rule base 129. In a preferred embodiment, the processor 124 may comprise a system having a processor, memory comprising a stored program, memory comprising data, and input/output devices, as is well known in the art. The operation and software structures of this system are described herein in terms of their functions and at a level of detail which would be clear to those of ordinary skill in the art. It would be clear to anyone of ordinary skill in the art, after perusal of the specification, drawings and claims herein, that modification and/or programming (using known programming techniques) of a processor of known design to achieve these functions would be a straightforward task and would not require undue experimentation. (<i>See Ex.1004 at 5:8-24</i>).</p> <p>Allen '218 also teaches</p> <p style="padding-left: 40px;">“A device 501, such as a photocopier or printer, may be coupled to a device driver interface 502, which may be coupled to a control processor 503 including the autonomous agent 101 [or AI engine]. . .</p> <p style="padding-left: 40px;">The device driver interface 502 provides the stimulus message 104 to the control processor 503, which delivers the stimulus message 104 to the autonomous agent 101.” (<i>Id. at 7:53-61</i>).</p> <p>It is noted that</p> <p style="padding-left: 40px;">“A software agent 101 may be embedded in an environment 102 so that the agent 101 may receive a stimulus 103 from the environment 102 by receiving a stimulus message 104, and may perform an action 105 which affects the environment 102 by sending an action message.” (<i>Id. at 3:56-60</i>).</p>

US 5,944,839 Claim Language	Correspondence to Allen ‘218
	<p>Further,</p> <p>In a preferred embodiment, the case database 205 may comprise a limited number of stored cases 126, the contents of the case database 205 being determined by a genetic technique for producing, evaluating and selecting the cases 126, such as a genetic technique like that disclosed with FIG. 3. Thus, while the case database 205 might store less than all of the cases 126 which the autonomous agent 101 has encountered, it might maintain a set of cases 126 which provides a preferred model of the environment, such as those cases 126 which allow the agent 101 to distinguish variant problem scenarios and to act autonomously and intelligently in those variant problem scenarios.</p> <p>(<i>Id. at 5:49-60</i>).</p>
<p>wherein when the knowledge database lacks data necessary to determine a likely solution to the computer problem, the AI engine activates a particular sensor in the plurality of sensors to gather the necessary data and store the data in the knowledge database, and</p>	<p>Allen ‘218 discloses</p> <p>“A behavior module 118 may receive the motives message 117, the features message 110 and the reward message 113, and may generate a <i>queries message 119, which may comprise software objects 107 which describe a request 120 by the agent 101 for further information.</i>” (<i>Ex. 1004 at 4:56-60</i>).</p>
<p>wherein when the knowledge database does not describe a likely solution to the computer problem, the AI engine saves the gathered</p>	<p>Allen ‘218 discloses that “the autonomous agent may comprise a memory of cases, the contents of that memory being determined by a genetic technique for producing, evaluating and selecting cases. <i>New cases may be produced by inspection of scenarios from the environment . . .</i>” (<i>Ex.1004 at 2:63-67</i>).</p>

US 5,944,839 Claim Language	Correspondence to Allen ‘218
data in the knowledge database as a new case.	
<p>2. The tool of claim 1, wherein each case comprises:</p> <p>at least one question asking about a particular aspect of the computer system that can be answered by the data gathered by the plurality of sensors; and</p>	<p>The ‘839 patent discloses that “[q]uestions belong to one or four categories: Yes/No; Numeric; Text; and List. Yes/No questions are those that have Yes or No answers. Numeric questions are those having answers that are integers. Text questions have textual answers. Finally, List questions have answers selected from a list of legal answers.” (<i>See Ex. 1001 at 3:52-57</i>).</p> <p>Allen ‘218 discloses that</p> <p style="padding-left: 40px;">The case database 205 may comprise its current set of cases 126, each of which may comprise a features element 301, which may generally indicate when the case 126 may be useful, and an action element 302, which may indicate the action 105 to take and an evaluation 303 of that action 105. (<i>Ex. 1004 at 6:11-15</i>).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that each of the cases of Allen ‘218 would include at least one question about a particular aspect of the computer system, particularly when the autonomous agent 101 is determining when a case 126 may be useful, as discussed above. (<i>Ex. 1007 at ¶¶ 54-56</i>).</p>
at least one action describing a likely solution to a potential computer problem stored in the knowledge database.	<p>The ‘839 patent discloses that “[a]n action is a statement that indicates a solution to a computer problem.” (<i>See Ex. 1001 at 3:58-59</i>).</p> <p>Allen ‘218 discloses that</p> <p style="padding-left: 40px;">“The case database 205 may comprise its current set of cases 126, each of which may comprise a features element 301, which may generally indicate when the case 126 may be useful, and <i>an action element 302, which may indicate the action 105 to take</i> and an evaluation 303 of that action 105.” (<i>Ex. 1004 at 6:11-</i></p>

US 5,944,839 Claim Language	Correspondence to Allen ‘218
	15).
<p>6. A method of optimizing a computer system, the method comprising the steps of:</p>	<p>Allen ‘218 discloses optimizing a computer system. Allen ‘218 discloses</p> <p style="padding-left: 40px;">The invention provides a software agent which performs autonomous learning in a real-world environment. The autonomous agent may learn by reinforcement (including positive and negative, and delayed and sporadic, reinforcement), in addition to learning by example and learning by being told what to do. In a preferred embodiment, the autonomous agent may be implemented in a case-based reasoning system, which may be coupled to a sensor for gathering information from, and to an effector for manipulating, its environment (which may comprise a software environment, a physical environment, or some combination thereof). In addition to gathering a case base of experience in its environment, the autonomous agent may tune that case base in response to an evaluation of how well it is operating in that environment. The evaluation may be its own, or may (See Ex.1004 at 2:43-57).</p>
<p>detecting a problem in the computer system;</p>	<p>Allen ‘218 discloses “[a] software agent [101] which performs autonomous learning in a real-world environment, implemented in a case-based reasoning system and coupled to a sensor for gathering information from . . . its environment.” (See Ex.1004 at Abstract).</p> <p>Allen ‘218 further discloses that</p> <p style="padding-left: 40px;">“A software agent 101 may be embedded in an environment 102 so that the agent 101 may receive a stimulus 103 from the environment 102[, detected by the sensor,] by receiving a stimulus message 104, and may perform an action 105 which affects the environment 102 by sending an action message.” (Ex. 1004 at 3:56-60).</p>
<p>activating an AI engine in response to the problem detection;</p>	<p>Allen ‘218 discloses activating an AI engine (“software agent 101”) in response to the problem detection.</p> <p>Allen ‘218 discloses that</p> <p style="padding-left: 40px;">“A software agent 101[, or AI engine,] may be</p>

US 5,944,839 Claim Language	Correspondence to Allen ‘218
	<p><i>embedded in an environment 102 so that the agent 101 may receive a stimulus 103 from the environment 102 by receiving a stimulus message 104, and may perform an action 105 which affects the environment 102 by sending an action message.” (Ex. 1004 at 3:56-60).</i></p> <p>Allen ‘218 further discloses:</p> <p style="padding-left: 40px;">In addition to taking action on a problem based on its case base, the autonomous agent may take action to gather information so as to determine which cases are most appropriate to that problem. (<i>Id. at 2:59-62</i>).</p>
utilizing, by the AI engine, selected ones of a plurality of sensors to gather information about the computer system;	<p>Allen ‘218 discloses “[a] software agent [101] which performs autonomous learning in a real-world environment, implemented in a case-based reasoning system and coupled to a sensor for gathering information from . . . its environment.” (<i>Ex. 1004 at Abstract</i>).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the software agent 101 of Allen ‘218 could be coupled to a plurality of sensors, instead of just a single sensor, for gathering information from the environment, as this represents nothing more than a simple substitution of one sensor with a plurality of sensors to predictably arrive at the claimed invention. (<i>Ex. 1007 at ¶¶ 51-53</i>).</p>
determining, by the AI engine, a likely solution to the problem from the gathered information; and	<p>Allen ‘218 discloses activating an AI engine (“software agent 101”) in response to the problem detection.</p> <p>Allen ‘218 discloses that</p> <p style="padding-left: 40px;">“A software agent 101[, or AI engine,] may be embedded in an environment 102 so that the agent 101 may receive a stimulus 103 from the environment 102 by receiving a stimulus message 104, and <i>may perform an action 105 which affects the environment 102 by sending an action message.</i>” (Ex. 1004 at 3:56-60).</p>

US 5,944,839 Claim Language	Correspondence to Allen ‘218
	The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the software agent 101, or AI engine, is likely to perform an action 105 after it has determined a likely solution to the problem based on the stimulus message 104. (<i>Ex. 1007 at ¶¶ 57-59</i>).
when a likely solution cannot be determined, saving a state of the computer system.	Allen ‘218 discloses that when a likely solution cannot be determined, the state of the computer system is saved. “The autonomous agent may comprise a memory of cases, the contents of that memory being determined by a genetic technique for producing, evaluating and selecting cases. <i>New cases may be produced by inspection of scenarios from the environment . . .</i> ” (<i>Ex. 1004 at 2:63-67</i>).
<p>8. The method of claim 6, wherein the determining step comprises the substeps of:</p> <p>inferring the likely solution to the problem from questions, actions, and rules contained in a knowledge database;</p>	<p>Allen ‘218 discloses that</p> <p>“The case database 205 may comprise its current set of cases 126, each of which may comprise a features element 301, which may generally indicate when the case 126 may be useful, and an action element 302, which may indicate the action 105 to take and an evaluation 303 of that action 105.” (<i>Ex. 1004 at 6:11-15</i>).</p> <p>Allen ‘218 further discloses that</p> <p>“the software agent 101 may be implemented with an automated processor 124, which may execute a software inference engine 125 for reasoning using a set of cases 126 in a case base 127 and a set of rules 128 in a rule base 129.” (<i>Id. at 5:8-12</i>).</p>
wherein the AI engine utilizes the selected ones of the plurality of sensors to gather information when the knowledge database lacks information	<p>Allen ‘218 discloses “[a] software agent [101] which performs autonomous learning in a real-world environment, implemented in a case-based reasoning system and coupled to <i>a sensor</i> for gathering information from . . . its environment.” (<i>See Ex.1004 at Abstract</i>).</p> <p>Allen ‘218 further discloses</p> <p>“A behavior module 118 may receive the motives message 117, the features message 110 and the reward</p>

US 5,944,839 Claim Language	Correspondence to Allen ‘218
necessary to answer a question.	<p>message 113, and may generate a <i>queries message 119, which may comprise software objects 107 which describe a request 120 by the agent 101 for further information.</i>” (Id. at 4:56-60).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the request 120 by the agent 101 for further information would utilize a sensor to gather additional information. (Ex. 1007 at ¶¶ 60-61).</p>
<p>14. The method of claim 6, wherein the detecting step comprises the steps of:</p> <p>periodically activating selected ones of the plurality of sensors to gather information about the computer system; and</p>	<p>Allen ‘218 discloses “[a] software agent [101] which performs autonomous learning in a real-world environment, implemented in a case-based reasoning system and coupled to <i>a sensor</i> for gathering information from . . . its environment.” (See Ex.1004 at Abstract).</p> <p>Allen ‘218 further discloses</p> <p>“A behavior module 118 may receive the motives message 117, the features message 110 and the reward message 113, and may generate a <i>queries message 119, which may comprise software objects 107 which describe a request 120 by the agent 101 for further information.</i>” (Id. at 4:56-60).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the autonomous learning system of Allen ‘218 would periodically activate at least one sensor to gather further information about the environment, particularly when the agent 101 requests further information. (Ex. 1007 at ¶¶ 62-63).</p>
analyzing the information to determine whether a problem exists.	As described with respect to claim 6, <i>supra</i> , Allen ‘218 teaches analyzing the information gathered from sensor to determine whether a problem exists.
15. A program storage device readable by a computer system, the program	Claim 15 is nearly coextensive in scope with claim 1, the only differences being (1) form over substance, <i>i.e.</i> , claim 1 recites a tool whereas claim 15 recites a program storage device for automatically maintaining a computer system, and (2) the limitation reciting “sensing information about the computer

US 5,944,839 Claim Language	Correspondence to Allen ‘218
storage device tangibly embodying a program of instructions executable by the computer system to perform method steps for automatically maintaining the computer system, the program storage device holding instructions for:	<p>system by <i>at least one sensor</i>” in claim 15 versus a <i>plurality of sensors</i> in claim 1.</p> <p>The method of Allen ‘218 may be readily executed on a computer readable storage medium. Because the scope of claim 1 necessarily includes all of the limitations of claim 15, Petitioner’s arguments with respect to claim 1 apply equally to claim 15.</p> <p>Accordingly, for at least the same reasons as set forth with respect to claim 1, claim 15 is rendered obvious over Allen ‘218 under 35 U.S.C. § 103(a).</p>
sensing information about the computer system by at least one sensor;	See claim 1, limitation reciting “a plurality of sensors stored in the memory and executing on the processor and adapted for gathering data about the computer system . . . ,” <i>supra</i> .
determining whether a computer problem exists from the sensed information;	See claim 1, limitation reciting “a plurality of sensors stored in the memory and executing on the processor and adapted for . . . detecting whether a computer problem exists from the data and the plurality of cases,” <i>supra</i> .
searching a plurality of cases with the sensed information to determine whether a likely solution to the computer problem exists;	See claim 1, limitation reciting “a knowledge database stored in the memory and holding a plurality of cases describing potential computer problems and corresponding likely solutions,” <i>supra</i> .
when additional information is needed to	See claim 1, limitation reciting “wherein when the knowledge database lacks data necessary to determine a likely solution to the computer problem, the AI engine activates a particular

US 5,944,839 Claim Language	Correspondence to Allen ‘218
determine a likely solution to the computer problem, activating the at least one sensor to sense the additional information; and	sensor in the plurality of sensors to gather the necessary data and store the data in the knowledge database,” <i>supra</i> .
when a likely solution to the computer problem does not exist, saving the sensed information as a new case of the plurality of cases.	See claim 1, limitation reciting “wherein when the knowledge database does not describe a likely solution to the computer problem, the AI engine saves the gathered data in the knowledge database as a new case,” <i>supra</i> .
<p>17. The program storage device of claim 15, wherein the sensing step comprises the substep of:</p> <p>periodically activating at least one sensor to gather information about the computer system.</p>	<p>Allen ‘218 discloses “[a] software agent [101] which performs autonomous learning in a real-world environment, implemented in a case-based reasoning system and coupled to <i>a sensor</i> for gathering information from . . . its environment.” (<i>See Ex.1004 at Abstract</i>).</p> <p>Allen ‘218 further discloses “A behavior module 118 may receive the motives message 117, the features message 110 and the reward message 113, and may generate a <i>queries message 119, which may comprise software objects 107 which describe a request 120 by the agent 101 for further information.</i>” (<i>Id. at 4:56-60</i>).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the autonomous learning system of Allen ‘218 would periodically activate at least one sensor to gather further information about the environment, particularly when the agent 101 requests further information. (<i>Ex. 1007 at ¶¶ 62-63</i>).</p>

C. Claims 6, 8 and 14 are Rendered Obvious by Barnett Taken in View of Allen ‘664 Under 35 U.S.C. § 103

U.S. Patent No. 5,664,093 to Barnett et al. (“Barnett,” Ex. 1005) was not considered during the original prosecution of the ‘839 patent, nor is it cumulative of any prior art considered by the Examiner. Barnett was filed on July 25, 1996 and issued on September 2, 1997. The earliest effective priority date of the ‘839 patent is March 19, 1997. Therefore, Barnett is available as prior art to the ‘839 patent under 35 U.S.C. § 102(e).

Similarly, U.S. Patent No. 5,581,664 to Allen (“Allen ‘664,” Ex. 1006) was not considered during the original prosecution of the ‘839 patent, nor is it cumulative of any prior art considered by the Examiner. Allen ‘664 was filed on May 23, 1994 and issued on December 3, 1996. The earliest effective priority date of the ‘839 patent is March 19, 1997. Therefore, Allen ‘664 is available as prior art to the ‘839 patent under 35 U.S.C. § 102(e).

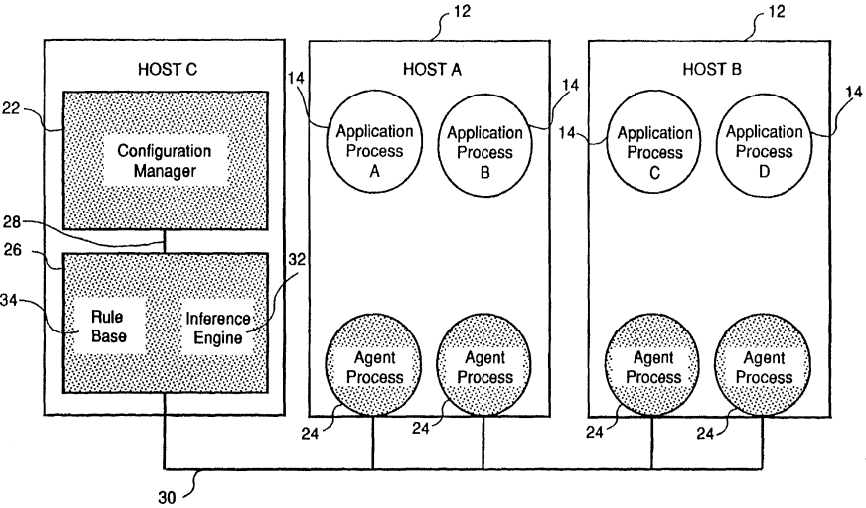
A skilled artisan would have been motivated to combine the teachings of Barnett and Allen ‘664 because the case-based reasoning system described by Allen ‘664, which is smoothly integrated into a rule-based reasoning system, could be easily substituted for the rule-based AI engine described in Barnett. It would be clear to a skilled artisan at that time that Allen ‘664’s hybrid case-based/rule-based approach would provide advantages over Barnett’s simple rule-based approach, e.g., it could handle cases that did not match specific rules, it could learn and adapt

its strategies over time, etc., which were well-documented in the many articles that had been published at that time on case-based reasoning. Hence this “upgrade” of the AI engine from rule-based to a hybrid case-based/rule-based approach would be an obvious step for improving Barnett’s system. (*Ex. 1007 at ¶ 71*).

The following claim chart demonstrates, on a limitation-by-limitation basis, how claims 6, 8 and 14 of the ‘839 patent are rendered obvious by Barnett in view of Allen ‘664 under 35 U.S.C. § 103. (*Id. at ¶¶ 66-76*).

US 5,944,839 Claim Language	Correspondence to Barnett in view of Allen ‘664
<p>6. A method of optimizing a computer system, the method comprising the steps of:</p>	<p>Barnett discloses</p> <p>“A system and method for managing faults in a distributed system. The fault management system includes a configuration manager that maintains configuration information of components used in the distributed system. A plurality of measurement agents obtain performance information from the components in the distributed system. A diagnostic system has a plurality of rules written according to the configuration information stored therein. The diagnostic system is coupled to the configuration manager and each of the plurality of measurement agents and identifies faults occurring in the distributed system and provides solutions for correcting the faults.” (<i>Ex. 1005 at Abstract</i>).</p>
<p>detecting a problem in the computer system;</p>	<p>Barnett discloses</p> <p>“The fault management system works as following for a fault being reported from a particular application. Initially, the fault is reported through the configuration manager 22 or from a measurement agent 24. The components that may be faulty are treated as objects.” (<i>Ex. 1005 at 5:24-</i></p>

US 5,944,839 Claim Language	Correspondence to Barnett in view of Allen ‘664
	27).
activating an AI engine in response to the problem detection;	<p>Barnett discloses</p> <p>“The fault management system works as following for a fault being reported from a particular application. Initially, the fault is reported through the configuration manager 22 or from a measurement agent 24. The components that may be faulty are treated as objects. The <i>diagnostic system 26 (or “AI engine”)</i> then asks the configuration manager what classes of objects depend from the application . . . The diagnostic system then asks for the explicit relationships about the given fault.” (Ex. 1005 at 5:24-37).</p>
utilizing, by the AI engine, selected ones of a plurality of sensors to gather information about the computer system;	<p>Barnett discloses that</p> <p>“Once a list of objects that are associated with the fault are received, the diagnostic system can now query the <i>measurement agents 24 (or “plurality of sensors”)</i> about the status of each instance of an object.” (Ex. 1005 at 5:47-50).</p> <p>Barnett further discloses that</p> <p>“Host computers A and B each run several application processes A and B and C and D, respectively. A <i>plurality of measurement agents 24</i> obtain performance information from the components and the processes in the hosts A and B.” (Id. at 2:56-60).</p>

US 5,944,839 Claim Language	Correspondence to Barnett in view of Allen '664
	 <p style="text-align: center;">FIG. 2</p>
<p>determining, by the AI engine, a likely solution to the problem from the gathered information; and</p>	<p>Barnett discloses that the diagnostic system (or “AI engine”) receives configuration information from the configuration manager and performance information from a plurality of measurement agents 24 (or “sensors”) and uses the information with the plurality of rules to identify faults and provide solutions for any faults. (<i>Ex. 1005 at 2:51 – 3:2</i>).</p> <p>Barnett further discloses that</p> <p style="padding-left: 40px;">system. Once a list of objects that are associated with the fault are received, the diagnostic system can now query the measurement agents 24 about the status of each instance of an object. In addition, the diagnostic system can ask about other objects that are required by the list of objects require. This allows the diagnostic system to learn the relationships between all components necessary for a functional system, as well as the status of each component. Using the rules in the rule base, the diagnostic system can identify performance problems and provide solutions for overcoming faults and sluggish performance of an application. The diagnostic (<i>Id. at 5:47-57</i>).</p>
<p>when a likely solution cannot be determined, saving a state of the</p>	<p>Barnett discloses that</p> <p style="padding-left: 40px;">“The diagnostic system receives the configuration</p>

US 5,944,839 Claim Language	Correspondence to Barnett in view of Allen ‘664
computer system.	<p>information from the configuration manager and the performance information from the plurality of measurement agents and uses the configuration and performance information with the plurality of rules to identify faults and provide solutions for the faults.” (<i>Ex. 1005 at 2:4-9</i>).</p> <p>Barnett, however, fails to teach or suggest the outcome when the diagnostic system cannot determine a likely solution.</p> <p>Allen ‘644 discloses that</p> <p>“In the case-matching step 202 or in the best-case step 203, the inference engine 111 may determine that there is no case 105 (<i>or solution</i>) which is a good match for the case template 312. The inference engine 111 may create a new case 105 which partly or fully copies the case template 312, and may ask the user 119 . . . what the prescribed action 309 for the case 105 should be . . .</p> <p>The inference engine 111 may also add new cases 105 to the case base 104 without asking the user 119. When the inference engine 111 determines that there is no case 105 which is a good match for the case template 312, <i>it may invoke a rule 103</i> or a procedural structure 117 which creases a new case 105 . . .” (<i>Ex. 1006 at 8:21-36</i>).</p> <p>The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the diagnostic system of Barnett would invoke a rule to save a state of the computer system when a likely solution to the identified faults cannot be determined, per the teachings of Allen ‘664. (<i>Ex. 1007 at ¶¶ 68-70</i>). The declaration of Prof. Mowry further sets forth the reasons why one skilled in the art would combine the Barnett and Allen ‘664 references</p>

US 5,944,839 Claim Language	Correspondence to Barnett in view of Allen ‘664
	in this manner. (<i>Id. at</i> ¶ 71).
<p>8. The method of claim 6, wherein the determining step comprises the substeps of:</p> <p>inferring the likely solution to the problem from questions, actions, and rules contained in a knowledge database;</p>	<p>Barnett discloses that</p> <p style="padding-left: 40px;">...the model can be constructed dynamically. The querying operation is performed by an inference engine 32. The diagnostic system also includes a rule base 34 comprising a plurality of rules for the various objects within the distributed application. By using the abstract relationships learned from the configuration manager and the rules from the rule base, the diagnostic system is able to monitor performance and diagnose several types of failures that are reported from any of the plurality of measurement agents. (<i>Ex. 1005 at</i> 4:6-14).</p>
<p>wherein the AI engine utilizes the selected ones of the plurality of sensors to gather information when the knowledge database lacks information necessary to answer a question.</p>	<p>Barnett discloses that “the diagnostic system 26 does not have information about the static or dynamic model, but can obtain all of the information from the configuration manager 22 and the measurement agents 24.” (<i>Ex. 1005 at</i> 4:25-28).</p> <p>Barnett further discloses:</p> <p style="padding-left: 40px;">...tools and provide solutions for any faults. There are several mechanisms which permit the diagnostic system to ask the configuration manager and the measurement agents for information. For example, there may be a coordinator which the diagnostic system uses to communicate with the configuration manager and the agents. The agents may in turn talk to other agents if they need to abstract and encapsulate information to the diagnostic system. (<i>Id. at</i> 3:2-9).</p>
<p>14. The method of claim 6, wherein the detecting step comprises the steps of:</p> <p>periodically activating selected ones of the plurality of sensors to gather information about the computer</p>	<p>Barnett discloses that</p> <p style="padding-left: 40px;">In addition, the fault and performance analysis of the present invention can be used for any system composed of hardware and software components, and even abstract components like actions, tasks, and deliverables. It can be integrated into network management applications and capacity planning tools. It can do load balancing and predictive fault analysis. In general it can be applied to any distributed application on a computer network that needs to react to changes in a dynamic or static environment.</p>

US 5,944,839 Claim Language	Correspondence to Barnett in view of Allen ‘664
system; and	<p data-bbox="662 285 976 321"><i>(Ex. 1005 at 7:7-15).</i></p> <p data-bbox="558 369 1437 789">The declaration of Prof. Mowry sets forth the reasons why one skilled in the art would understand that the diagnostic system of Barnett would periodically activate at least one of the measurement agents 24 to gather information about the computer system, i.e., performance information related to the components in the distributed system, particularly when the fault and performance analysis of Barnett is integrated into network management applications and capacity planning tools and can do load balancing and predictive fault analysis. <i>(Ex. 1007 at ¶¶ 74-75).</i></p>
analyzing the information to determine whether a problem exists.	<p data-bbox="558 800 1338 959">As described with respect to claim 6, <i>supra</i>, Barnett teaches analyzing the information gathered from the plurality of sensors (or measurement agents 24) to determine whether a problem exists.</p>

VIII. CONCLUSION

Substantial, new and noncumulative technical teachings have been presented for each of claims 1, 2, 6, 8, 14, 15 and 17 of the '839 patent, which are anticipated or rendered obvious for the reasons set forth above. There is a reasonable likelihood that Petitioner will prevail as to each of the claims. *Inter Partes* Review of claims 1, 2, 6, 8, 14, 15 and 17 of the '839 patent is accordingly requested.

The USPTO is authorized to charge any fees or credit any overpayments to Deposit Account No. 15-0030.

Respectfully submitted,
OBLON SPIVAK

Dated: December 21, 2012

/Greg H. Gardella/
Greg H. Gardella
Reg. No. 46,045

Scott A. McKeown
Reg. No. 42,866

Customer Number
22850
Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 02/10)

CERTIFICATE OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b) on the Patent Owner by Express Mail of a copy of this Petition for *Inter Partes* Review and supporting materials at the correspondence address of record for the '839 patent:

FENWICK & WEST LLP
SILICON VALLEY CENTER
801 CALIFORNIA STREET
MOUNTAIN VIEW CA 94041

Dated: December 21, 2012

By: /S/ /Greg H. Gardella/
Greg H. Gardella (Reg. No. 46,045)