Trials@uspto.gov
571-272-7822

Paper 9
Entered: May 17, 2013

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ORACLE CORPORATION
Petitioners,

v.

CLOUDING IP, LLC
Patent Owner.

Case IPR2013-00095 (JL)
Patent 5,944,839

Before JAMESON LEE, MICHAEL W. KIM, and RAMA G. ELLURU
Administrative Patent Judges.

LEE, *Administrative Patent Judge*.

DECISION
Institution of *Inter Partes* Review
37 C.F.R. § 42.108

I. INTRODUCTION

Oracle Corporation ("Oracle") filed a petition (Paper 1, "Pet.") requesting *inter partes* review of claims 1, 2, 6, 8, 14, 15, and 17 of Patent 5,944,839 (Ex. 1001, "the '839 patent"). Clouding IP, LLC ("Clouding"), the patent owner, filed a preliminary response on March 26, 2013. (Paper 6, "Prel. Resp.") We have jurisdiction under 35 U.S.C. § 314.

The standard for instituting an *inter partes* review is set forth in 35 U.S.C. § 314(a) which provides as follows:

> THRESHOLD -- The Director may not authorize an inter partes review to be instituted unless the Director determines that the information presented in the petition filed under section 311 and any response filed under section 313 shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.

Upon consideration of the petition and the patent owner's preliminary response, we determine that the information presented in the petition establishes that there is a reasonable likelihood that Oracle would prevail with respect to claims 6, 8, and 14 of the '839 patent. Accordingly, pursuant to 35 U.S.C. § 314, we authorize an *inter partes* review to be instituted on claims 6, 8, and 14. The petition for instituting *inter partes* review is ***granted***.

A. Related Proceedings

Oracle indicates that the '839 patent is involved in co-pending litigation captioned *Clouding IP, LLC v. Oracle Corp.*, Case No. 1:12-cv-00642 (D.Del.). (Pet. 3.)

B. The ’839 Patent

The ’839 patent is directed to a system and method for automatic maintenance of a computer system. (Ex. 1001, Title; Abstract:1-2.) The specification explains that, although there is a great need for technical support professionals who can diagnose, repair, and optimize personal computers, those professionals are often unavailable or too expensive to call when a computer problem occurs. (Ex. 1001, 1:22-26.) The specification further explains that pre-existing automated utility programs have deficiencies. (Ex. 1001, 1:28-29). One such program adopts a “canned” approach, which merely schedules a fixed set of tools to execute at a predetermined time, and thus is inflexible and performs too much maintenance. (Ex. 1001, 1:32-36). Other such programs often misdiagnose the computer problem and apply the wrong solution. (Ex. 1001, 1:37-42.)

According to the specification, there is a need for a computer utility that can learn from the results of prior diagnosis and solutions and apply that learned knowledge to solve future computer system problems. (Ex. 1001, 1:47-50.) The disclosed invention of the ’839 patent uses a set of sensors in combination with a case database to diagnose and solve computer problems automatically. (Ex. 1001, 1:56-58.) When one of the sensors detects a problem, it activates an artificial intelligence (“AI”) engine. (Ex. 1001, 1:62-64.) The AI engine includes a database, which includes information to diagnose and correct problems with the computer system, and the sensors provide inputs to the database. (Ex. 1001, 1:65 to 2:3.) The AI engine diagnoses the cause of the problem and determines a solution, and if the AI engine needs more information, it may request additional data from one or more sensors. (Ex. 1001, 2:4-8.)

As claimed, the invention includes the requirement that (a) when the knowledge base 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 (claim 1), (b) when a likely solution cannot be determined, saving a state of the computer system (claim 6), and (c) when a likely solution to the computer problem does not exist, saving the sensed information as a new case of the plurality of cases (claim 15). During initial examination, the patent owner distinguished its claims from the prior art references applied by the examiner on the basis that the references do not disclose such limitations. (Ex. 1002, at January 27, 1999 Amendment, pp. 6-7.)

C. Exemplary Claims

Of the challenged claims, claims 1, 6, and 15 are independent claims. Each of claims 1, 6, and 15 is reproduced below (emphasis added):

> 1. A tool for automatically maintaining a computer system having a processor and a memory, the tool comprising:
>
> a knowledge database stored in the memory and holding a plurality of cases describing potential computer problems and corresponding likely solutions;
>
> 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
>
> an AI engine stored in the memory and 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 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 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.

6. A method of optimizing a computer system, the method comprising the steps of:

detecting a problem in the computer system;

activating an AI engine in response to the problem detection;

utilizing, by the AI engine, selected ones of a plurality of sensors to gather information about the computer system;

determining, by the AI engine, a likely solution to the problem from the gathered information; and

when a likely solution cannot be determined, saving a state of the computer system.

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:

> sensing information about the computer system by at least one sensor;
>
> determining whether a computer problem exists from the sensed information;
>
> searching a plurality of cases with the sensed information to determine whether a likely solution to the computer problem exists;
>
> 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
>
> **when a likely solution to the computer problem does not exist, saving the sensed information as a new case of the plurality of cases**.

D. Prior Art Relied Upon by Petitioner

Oracle relies upon the following prior art references:

Allen '664	U.S. Patent 5,581,664	Dec. 3, 1996	(Ex. 1006)
Allen '218	U.S. Patent 5,586,218	Dec. 17, 1996	(Ex. 1004)
Barnett	U.S. Patent 5,664,093	Sep. 2, 1997	(Ex. 1005)
Gürer, D., Khan, I, and Ogier, R., An Artificial Intelligence Approach to Network Fault Management (**Gürer**)		Feb. 1996	(Ex. 1003)

E. The Asserted Grounds

Oracle alleges that the challenged claims are unpatentable based on the following grounds:

1. Claims 1, 2, 6, 8, 14, 15, and 17 as unpatentable under 35 U.S.C. § 103 for obviousness over Gürer;
2. Claims 1, 2, 6, 8, 14, 15, and 17 as unpatentable under 35 U.S.C. § 103 for obviousness over Allen ’218;
3. Claims 6, 8, and 14 as unpatentable under 35 U.S.C. § 103 for obviousness over Barnett and Allen ’664.

II. ANALYSIS

A. Claim Construction

As a first step in our analysis for determining whether to institute a review, we determine the meaning of the claims. In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b). Under the broadest reasonable construction standard, claim terms are presumed to be given their ordinary and customary meaning as would be understood by one of ordinary skill in the art at the time of the invention. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc). An inventor may rebut that presumption by providing a definition of the term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). Therefore, the words of the claim will be given their plain meaning unless the plain meaning is inconsistent with the specification. *In re Zletz*, 893 F.2d 319, 321 (Fed. Cir. 1989). In that regard, we must be careful not to read a particular embodiment appearing in the written description into the claim if the claim language is broader than the embodiment. *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993).

Solution to the problem / Solution to the computer problem

Claims 1 and 15 each refers to "solution to the computer problem"; and claim 6 first recites "detecting a problem in the computer system," and then refers to "solution to the problem." The plain meaning of "solution," in the absence of special definition, requires specific information on problem resolution or elimination and is not satisfied by mere identification of the problem. The specification of the '839 patent does not set forth a special definition for the term "solution," and does not include an indication that mere identification of the problem is or provides a solution. We construe "solution to the problem" and "solution to the computer problem" such that they would not be satisfied by mere identification of the problem. Mere identification of the problem is insufficient.

Sensors

"[S]ensors" are defined in the specification of the '839 patent as software programs that gather information from a computer system. (Ex. 1001, 3:16-17.) We construe the term according to that definition provided in the specification.

Artificial Intelligence (AI) Engine

The term "AI Engine" is not defined in the specification of the '839 patent. It is not a term of common parlance in the English language, but a technical term of art. Both Oracle and Clouding attempt to set forth its desired meaning, but both proposed meanings have deficiencies. Oracle states (Pet. 16:2-6):

> 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*).

According to Oracle's technical witness Professor Todd C. Mowry, Ph.D., both the sensors and the AI engine are pieces of software. (Ex. 1007, ¶ 24:3.) Professor Mowry further states (Ex. 1007, ¶ 24:7-11): "it would be obvious to a skilled artisan that the software that performs the sensor functionality and the software that performs the AI engine functionality might be integrated within the same larger software application such that the distinction in functionality between the sensors and the AI engine is no longer existent."

Clouding, on the other hand, states the following regarding the meaning of "AI engine" (Prel. Resp. 4:12-14):

> Under a broadest reasonable construction the term "AI engine" means a software program for processing information in a database according to case-based scenarios. *Id*. at col. 4, ll. 12-20. [Citing to the specification of the '893 patent.]

The letters "A" and "I" in "AI" stand for Artificial Intelligence. We agree with the parties that in light of the specification of the '893 patent, one with ordinary skill would understand that "AI engine" is a computer program. However, it is not just any computer program. It is a computer program used for decision making or problem solving, based on reasoning or inferences.

Under the rule of broadest reasonable interpretation, nothing in the specification precludes the AI engine from being a part of the same computer

program which implements one or more sensors. Therefore, Oracle's assertion in that regard is correct. However, the underlying basis for that construction is not what would have been obvious to one with ordinary skill, as argued by Oracle.

Clouding's contention that "AI engine" is limited to that which processes database information in a case-based scenario is unpersuasive. The fact that the specification of the '893 patent discloses, as a preferred embodiment, an AI engine which employs case-based reasoning and uses a database for storing case information, does not limit the AI engine recited in the claims to a particular kind or type. The rule of broadest reasonable interpretation precludes importation into the claims of an "extraneous limitation" from the specification. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). A limitation is extraneous if there is no need for its inclusion in the claim for the claim to have a reasonable meaning. *See id*.; *E.I. du Pont de Nemours & Co. v. Phillips Petroleum Co.*, 849 F.2d 1430, 1433 (Fed. Cir. 1988). In patent law, "the name of the game is the claim." *In re Hiniker Co.*, 150 F.3d 1362, 1369 (Fed. Cir. 1998).

Clouding merely has pointed to the type of AI engine in a preferred embodiment described in the specification of the '893 patent. Clouding neither has shown that there is no other type of AI engine because all AI engines employ case-based reasoning, nor has explained why the term "AI engine" should be interpreted as covering only AI engines which use or employ case-based reasoning. For instance, it is unexplained why an AI engine cannot be based on neural networks or rule-based expert systems. *See* Gürer (Ex. 1003, 3:6-26.)

B. Claims 1, 2, 6, 8, 14, 15, 17 – Obvious over Gürer

Oracle contends that claims 1, 2, 6, 8, 14, 15, and 17 are unpatentable under 35 U.S.C. § 103 as obvious over Gürer. In support of that contention, Oracle relies on a declaration of Professor Todd C. Mowry. (Ex. 1007.) Upon review of Oracle's analysis and supporting evidence, we determine that Oracle has not demonstrated that there is a reasonable likelihood that it would prevail with respect to claims 1, 2, 6, 8, 14, 15, and 17 on the ground that those claims are unpatentable for obviousness over Gürer.

Gürer is titled "An Artificial Intelligence Approach to Network Fault Management." In the Abstract, at lines 1-2, it is stated that traditionally, network management activities, such as fault management, have been performed with direct human involvement, but that these activities are becoming more demanding and data intensive. Gürer further states (Ex. 1003, Abstract:3) that it is becoming necessary to automate network management activities. According to Gürer (Ex. 1003, Abstract:3-5), artificial intelligence technologies play an important role in the problem solving and reasoning techniques that are employed in fault management. In that regard, Gürer notes that "expert systems" are not sufficient for today's evolving network needs, and proposes a hybrid Artificial Intelligence (AI) solution that employs both neural networks and case-based reasoning techniques for the fault management of heterogeneous distributed informational networks. (Ex. 1003, Abstract:5-7.)

Gürer's disclosed approach includes an initial stage for collecting system alarms, a subsequent stage for diagnosing a fault, and then another stage to correct

the diagnosed fault. (Ex. 1003, 1:30 to 2:7; Figure 1.) Figure 1 of Gürer is reproduced below:

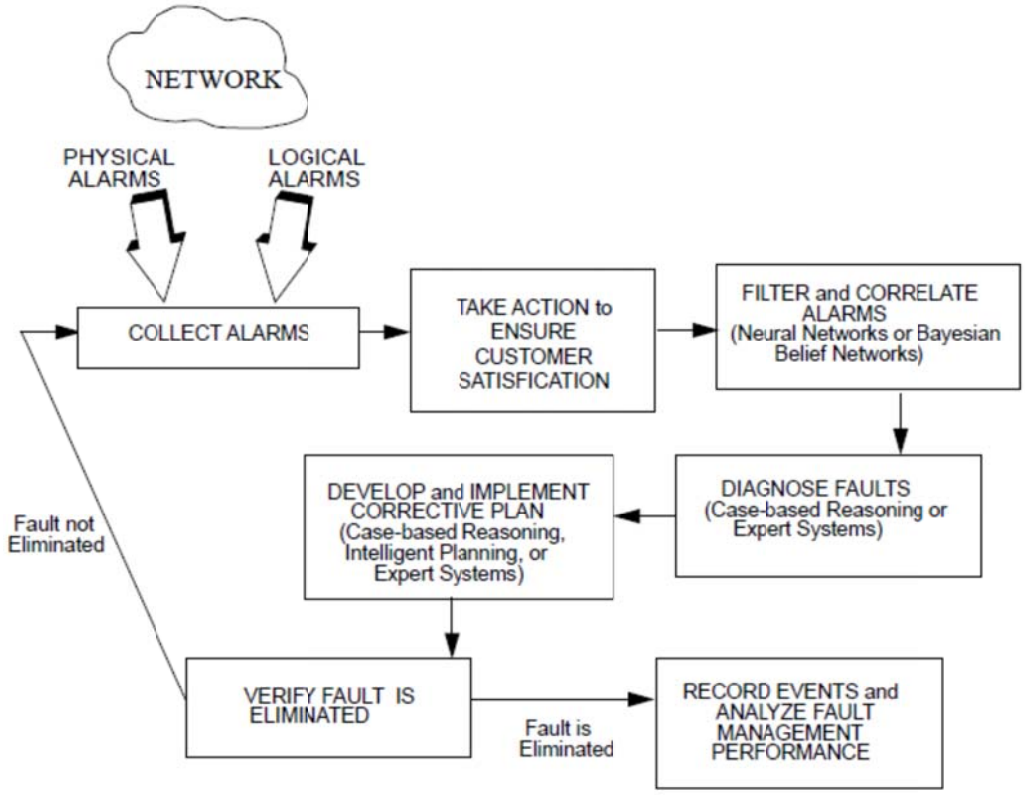

Figure 1: The fault management process and possible AI technologies.

For **fault diagnosis**, Gürer describes that it is appropriate to train a Neural Network or construct a Bayesian Belief Network to correlate alarms and recognize basic fault patterns, and then use symbolic processing such as case-based reasoning to further analyze the data, run tests, and pinpoint the exact identity and location of the fault. (Ex. 1003, 3:35-37.) For **fault correction** after fault diagnosis, Gürer recommends using case-based reasoning. Specifically, Gürer states (Ex. 1003, 9:11-14):

> Additionally, CBR [Case Based Reasoning] systems will be able to use previous experiences in formulating a new fault correction strategy. For example, when a device is unresponsive to manual correction, similar cases (i.e., previous fault correction solutions) that had the same problem can be retrieved from the case library and their solutions analyzed. Drawing analogies from those cases, the CBR system could come up with an appropriate solution within the context of the current problem.

As discussed above, in Gürer's disclosed approach, both fault identification and fault correction use case-based reasoning relying on a plurality of prior cases stored in a case database. In that regard, Gürer's disclosed system accesses cases from the case library for diagnosis purposes, and also cases from the case library for fault correction purposes, as illustrated in Figure 4 reproduced below (Ex. 1003, Figure 4):

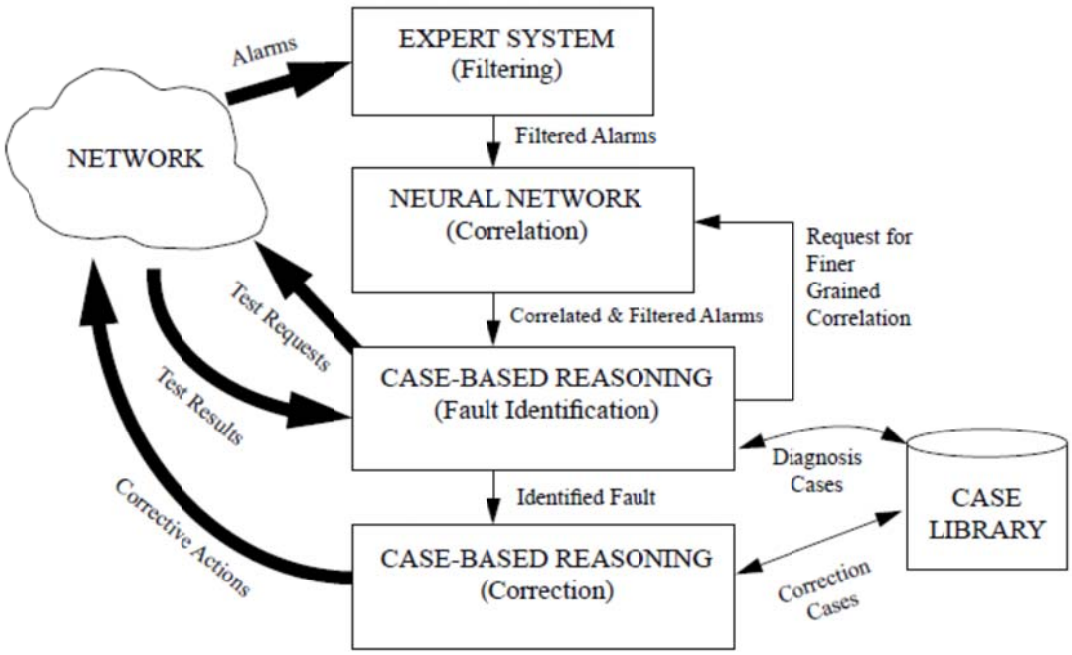

Figure 4: A hybrid AI system for automated network fault management.

For the limitation in independent claim 1, "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," Oracle cites to the following text in Gürer as satisfying the recited feature (Ex. 1003, 8:6-8):

> 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.

Oracle also cites to the same above-quoted text of Gürer to meet this claim feature of independent claim 6: "when a likely solution cannot be determined, saving a state of the computer system." Oracle further cites to the same above-quoted text of Gürer to meet this claim feature of independent claim 15: "when a likely solution to the computer problem does not exist, saving the sensed information as a new case of the plurality of cases."

We are unpersuaded that the cited disclosure of Gürer meets the above-noted limitations of claims 1, 6, and 15. That disclosure is within the "**Fault diagnosis**" portion of the disclosure from page 3, line 30, to page 8, line 9, and pertains solely to problem diagnosis. It does not teach determining a "solution" for correcting the identified fault. The portion of Gürer's disclosure relating to determining a solution of the diagnosed problem begins on page 8, line 10, subsequent to the text quoted by Oracle. Note that the quoted text begins with the "[o]nce the fault is identified," not "once a solution to the fault is identified."

In any event, even assuming that the quoted text relates to problem correction and not mere problem identification, the quoted disclosure still does not meet the above-identified features of independent claims 1, 6, and 15. All of those features require the saving of information when a likely solution cannot be found. In that regard, claims 1 and 15 refer to the saving of gathered or sensed data as a new case, and claim 6 refers to the saving of the state of the computer system. In each case, the premise is that a likely solution to the problem cannot be determined. That concept is absent from the quoted text in Gürer. Rather, the contrary is true. It indicates that once a fault is identified, all information including any circuitous paths or dead-ends taken on the way to problem identification, as well as the eventual success of the analysis, are stored into a new case. The pre-condition to the storing of a new case is that the analysis ultimately or eventually is successful. Only in that event are circuitous paths and dead-ends taken stored. That comes from a plain reading of the quoted text. Oracle does not explain why that is not the case.

For the foregoing reasons, we determine that Oracle has not shown a reasonable likelihood that it would prevail in demonstrating unpatentability of claims 1, 6, and 15 under 35 U.S.C. § 103 as obvious over Gürer. Claims 8 and 14 each depend on claim 6, and claim 17 depends on claim 15. Similarly, we determine that Oracle has not shown a reasonable likelihood that it would prevail in demonstrating unpatentability of claims 8, 14, and 17 under 35 U.S.C. § 103 as obvious over Gürer.

C. Claims 1, 2, 6, 8, 14, 15, 17 – Unpatentable Over Allen '218

Allen '218 is directed to a case-based reasoning system which performs autonomous learning in a real-world environment. (Ex. 1004, 1:18-20.) In Allen '218, it is stated that it would be advantageous if an automated reasoning system dynamically could create its own case base without substantial human intervention, or at least with only occasional human intervention. (Ex. 1004, 2:19-24.)

Allen '218 discloses a software Agent 101, implemented on a 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. (Ex. 1004, 5:8-12.) In pertinent part, Allen '218 states (Ex. 1004, 5:49-60):

> 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.

For the limitation in independent claim 1, "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," Oracle cites to the following text in Allen '218 as satisfying the recited feature (Ex. 1004, 2:63-3:1):

> In a preferred embodiment, 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. New cases may be produced by inspection of scenarios from the environment, by mutation of old cases . . . .

Oracle also cites to the same above-quoted text of Allen '218 to meet this feature of independent claim 6: "when a likely solution cannot be determined, saving a state of the computer system." Oracle further cites to the same above-quoted text of Allen '218 to meet similar feature of independent claim 15: "when a likely solution to the computer problem does not exist, saving the sensed information as a new case of the plurality of cases."

The cited portion of Allen '218 is inadequate to support Oracle's contentions regarding the above-noted features of independent claims 1, 6, and 15. Aside from citing the above-quoted text, Oracle provides no explanation. We find nothing in the above-quoted text from Allen '218, which indicates any action or operation which is triggered or caused by not finding a likely solution to a computer problem or any problem. The cited text describes no particular circumstance in which a new case is created or saved into the case database. It is speculative to deduce or infer that when a likely solution cannot be determined, the sensed information or the gathered data is saved in the case database as a new case, or that the state of the computer system is saved.

Furthermore, while the above-noted limitations of claims 1, 6, and 15 each refer to detecting a computer problem or a problem in a computer system, Oracle has not addressed adequately how the disclosure of Allen '218 accounts for such a computer problem. For instance, while Oracle refers to "the computer system"

within office equipment 501 illustrated in Figure 5 of Allen ’218 (Pet. 35:19-25), which is a printer or photocopier being monitored, it identifies no disclosure that any detected problem or failure of device 501 is from a computer system in device 501. Oracle also identifies no description that refers to any computer system within office equipment device 501.

For the foregoing reasons, we determine that Oracle has not shown a reasonable likelihood that it would prevail in demonstrating unpatentability of claims 1, 6, and 15. Claims 8 and 14 each depend on claim 6, and claim 17 depends on claim 15. Similarly, we determine that Oracle has not shown a reasonable likelihood that it would prevail in demonstrating unpatentability of claims 8, 14, and 17 under 35 U.S.C. § 103 as obvious over Allen ’218.

D. Claims 6, 8, 14 – Unpatentable over Barnett and Allen ’664

Claim 6 is independent, and each of claims 8 and 14 depend on claim 6. The arguments and evidence presented by Oracle with regard to claims 6, 8, and 14 are persuasive. For instance, claim 6 requires that “when a likely solution cannot be determined, saving a state of the computer system.” Oracle sufficiently has accounted for that requirement for each of claims 6, 8, and 14.

Barnett discloses a fault management system including a plurality of measuring agents, which obtain performance information from the system being monitored, and a diagnostic system having a plurality of rules stored therein. (Ex. 1005, 1:59-67.) The diagnostic system receives performance information from the measuring agents and uses the stored rules to identify faults and provide solutions for the faults. (Ex. 1005, 2:4-9.) The system being monitored is a distributed

system including a plurality of interconnected computers 12. (Ex. 1005, 2:32-37.) As shown in Figure 2, Barnett's diagnostic system includes a rule based AI engine 26 having a rules database 34:

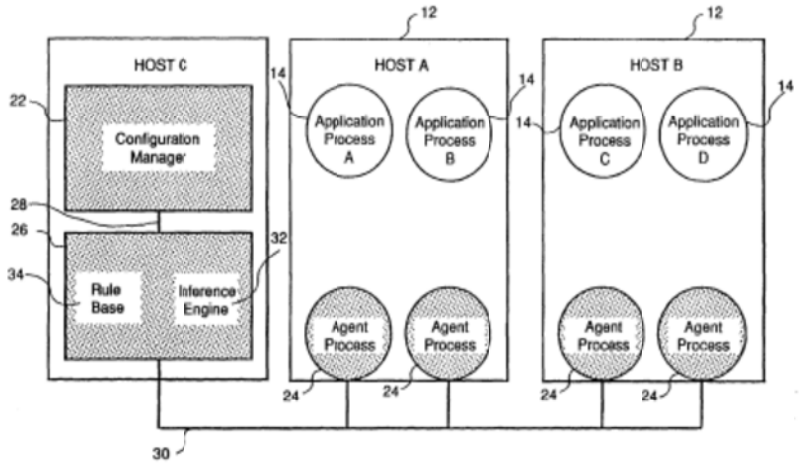

FIG. 2

Figure 2 illustrates a block diagram of a disclosed embodiment of the fault management system of Barnett

Oracle acknowledges that Barnett does not describe what action to take when its system, specifically its rule based AI engine, cannot determine a likely solution to a detected problem. (Pet. 51:2:8-10; Ex. 1007, ¶ 68:5-7.) However, Oracle persuasively argues that in light of Barnett's disclosure, it would have been obvious to one with ordinary skill to save the state of the computer system having the problem, when a likely solution to the problem cannot be determined. In that regard, Professor Mowry testifies in paragraph 69 of his declaration as follows (Ex. 1007, ¶ 69):

> (69) Although rule-based systems do not automatically update their rules when they fail, Barnett does teach that the precision of the AI engine can be improved by adding additional rules. (*Ex. 1005 at 7:1-6*). Hence, consistent with claim 6 of the '839 patent which recites, in part, "when a likely solution cannot be determined, saving a state of the computer system.," it would be clear to a person of ordinary skill in the art that in order for the programmer or user to understand how and when these additional rules should be added, a state of the computer system should be saved when the AI engine fails to determine a likely solution, so that the programmer or user can later determine the context when the failure took place. Since actions such as this are implemented via rules in a rule-based system, it would be natural and obvious to a skilled artisan to add a rule to the rule base that states that when a likely solution cannot be found, such a state of the computer system should be saved.

We credit the above-quoted testimony of Professor Mowry. We agree with Oracle's contention that in light of the fact that it is desirable to create new rules to handle new situations, it would have been obvious to save the state of the computer system having a problem, when application of the current set of stored rules fail to provide a likely solution to the problem.

However, Barnett discloses a rule-based inference system, not a system employing case-based reasoning as is required by claims 6, 8, and 14. Oracle relies on a combination of Barnett and Allen '664 to make up for that deficiency.

Allen'664 discloses a hybrid AI engine in the form of a case-based reasoning system, which is integrated into a rule-based reasoning system. (Ex. 1006, Abstract:1-2.) The disclosed system of Allen '664 may select the case which is the best match for the encountered problem, but may act

differently from the precise action prescribed for that case. (Ex. 1006. Abstract:8-10.) Allen '664 describes dynamically adapting its case database to the problems encountered, where additional cases may be created for consideration in the resolution of future problems and old or obsolete cases are removed. (Ex. 1006, Abstract:11-16.) Allen '664 also describes that the system may solicit human advice for treating problems which are not well-treated by the case database. (Ex. 1006, Abstract:16-20.)

Professor Mowry explains (Ex. 1007, ¶ 71);

> (71) A skilled artisan would be 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.

We credit the above-quoted testimony. For the reasons indicated in the testimony of professor Mowry, we agree with Oracle that in light of the disclosure of Allen '664, one with ordinary skill in the art would have known to "upgrade" the system of Barnett to use a hybrid case-based and rule-based AI engine for diagnosing problems and providing a solution to the problems. That would include

the addition of a case database for storing sample cases, in addition to the rules database for storing rules.

The inclusion of case-based reasoning and a case database to the rule-based system of Barnett does not affect the above-discussed reasoning regarding why one with ordinary skill in the art would have known to save the state of the computer system having a problem, if a likely solution to the problem cannot be determined. The rules database still exists and rule-based reasoning is still employed. Neither is eliminated by the addition use of case-based reasoning and the addition of a case database. Furthermore, Allen '664 also contemplates that when the existing cases in the database are inadequate to provide a solution, the system may ask the user what action to take. (Ex. 1006, 8:21-28.) In light of that disclosure, and as in the case of Barnett as discussed above, one with ordinary skill in the art would have known to store the state of the computer system having the problem, so that the user may consider the situation and provide an appropriate response.

Clouding argues that while the need for additional rules may be recognized by one with ordinary skill in the art, there is nothing to suggest that the state of the computer would be included as a part of these new rules. (Prel. Resp. 16:9-13.) The argument is misplaced. Oracle does not contend that the state of the computer would somehow be incorporated into a new rule. Rather, the position is simply that the state of the computer should be stored so that a user or operator can examine the situation to try to formulate a solution to be included in a new rule.

For all of the foregoing reasons, we determine that Oracle has demonstrated a reasonable likelihood that it would prevail in demonstrating that claims 6, 8, and

14 are unpatentable under 35 U.S.C. § 103 for obviousness over Barnett and Allen ’664.

III. CONCLUSION

For the forgoing reasons, we determine that the information presented in the petition establishes that there is a reasonable likelihood that Oracle would prevail in demonstrating unpatentability of claims 6, 8, and 14 of the ’839 patent, but not claims 1, 2, 15, and 17 of the ’839 patent.

IV. ORDER

Accordingly, it is

ORDERED that pursuant to 35 U.S.C. § 314, an *inter partes* review is hereby instituted as to claims 6, 8, and 14 of the ’839 patent on the ground of unpatentability under 35 U.S.C. § 103 for obviousness over Barnett and Allen ’664;

FURTHER ORDERED that *inter partes* review is not instituted with respect to claims 1, 2, 6, 8, 14, 15, and 17, either on the ground of obviousness over Gürer or on the ground of obviousness over Allen ’218;

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(d) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial; the trial is commencing on the entry date of this decision; and

FURTHER ORDERED that an initial conference call with the Board is scheduled for 3:00 PM Eastern Time on June 6, 2013; the parties are directed to

the Office Trial Practice Guide[1] for guidance in preparing for the initial conference call, and should come prepared to discuss any proposed changes to the Scheduling Order entered herewith and any motions the parties anticipate filing during the trial.

[1] *Office Patent Trial Practice Guide*, 77 *Fed. Reg*. 48756, 48765-66 (Aug. 14, 2012).

Case IPR2013-00095
Patent 5,944,839

For PETITIONER:

Greg Gardella
Scott A. McKeown
OBLON SPIVAK
cpdocketgardella@oblon.com
cpdocketmckeown@oblon.com

For PATENT OWNER

Tarek N. Fahmi
Amy J. Embert
Fahmi, Sellers & Embert
tarek.fahmi@tnfip.com
amy.embert@fseip.com