

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION, EMC INTERNATIONAL US. HOLDINGS, INC.,
AND VMWARE, INC.,
Petitioner,

v.

CLOUDING CORP.
Patent Owner.

Case IPR2014-01309
Patent 5,944,839 A

Before JAMESON LEE, RAMA G. ELLURU, and
MIRIAM L. QUINN, *Administrative Patent Judges*.

ELLURU, *Administrative Patent Judge*.

FINAL WRITTEN DECISION ON REMAND
35 U.S.C. § 144 and 37 C.F.R. § 42.5(a)

BACKGROUND

This Decision addresses the United States Court of Appeals for the Federal Circuit’s remand in *EMC Corp., EMC Int’l. U.S. Holdings, Inc., VMware, Inc. v. Clouding Corp.*, No. 2016-1999, 686 F. Appx. 857 (Fed. Cir. May 1, 2017) (not precedential).

EMC Corporation, EMC International U.S. Holdings, Inc., and VMware, Inc. (collectively “Petitioner”) filed a Petition (Paper 1 (“Pet.”)) requesting an *inter partes* review of claims 1, 2, 6, 8, 14, 15, and 17 of U.S. Patent No. 5,944,839 A (Ex. 1001, “the ’839 patent”) owned by Clouding Corp. (“Patent Owner”). We instituted an *inter partes* review of claims 6, 8, and 14. Paper 14 (“Inst. Dec.”), 16.

Patent Owner filed a Response (Paper 22 (“PO Resp.”)) and Petitioner filed a Reply (Paper 25 (“Pet. Reply”)) to Patent Owner’s Response. After briefing and oral argument, we issued a Final Decision (Paper 34 (“Final Dec.”), 15) finding that claims 6, 8, and 14 had not been shown to be unpatentable over the combination of Barnett¹ and Allen ’664². Petitioner appealed our decision to the Federal Circuit.

On May 1, 2017, the Federal Circuit issued a decision, vacating and remanding our Final Decision. *EMC*, 686 F. Appx. 858. The Federal Circuit determined we erred, and in particular determined that Petitioner properly argued, and Patent Owner did not dispute, that the combination of Barnett and Allen ’664 teaches the condition “when a likely solution [to a problem in the computer system] cannot be determined.” *Id.* at 860–62.

¹ U.S. Patent 5,664,093 (Ex. 1005).

² U.S. Patent 5,581,664 (Ex. 1006).

The Federal Circuit further determined that Patent Owner admitted that there would have been motivation to combine Barnett and Allen '664. *Id.* at 863. Of relevance to this decision, the Federal Circuit determined that Patent Owner's only argument to the Board in support of nonobviousness was that Petitioner's proposed combination of Barnett and Allen '664 would not yield a solution that involves "saving a state of a computer system," as the second half of claim 6, step 5 requires. *Id.* Because there is no claim of secondary considerations, the Federal Circuit remanded for us to answer only one question: "whether Barnett and Allen, considered in combination, would [have] suggest[ed] to a person of ordinary skill in the art 'saving a state of the computer system.'" *Id.* The mandate issued on June 7, 2017.

After considering the entire record, we conclude that Petitioner has shown, by a preponderance of the evidence, that claims 6, 8, and 14 are unpatentable over the combination of Barnett and Allen '664 because the combination teaches or suggests "saving a state of the computer system."

DISCUSSION

A. *The '839 Patent*

The '839 patent is directed to a system and method for automatic maintenance of a computer system. Ex. 1001, Abstract. According to the specification, there is a need for a computer utility that can learn from the results of prior diagnoses and solutions and apply that learned knowledge to solve future computer system problems. *Id.* at 1:47–50. The disclosure of the '839 patent is directed to a "maintenance tool" that uses a set of sensors in combination with a case database to diagnose and solve computer problems automatically. *Id.* at 1:56–58. When one of the sensors detects a

computer problem, it activates an artificial intelligence (“AI”) engine. *Id.* at 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. *Id.* at 1:65. 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. *Id.* at 2:4–8. If the maintenance tool has gathered all possible data and still does not include a solution to the computer problem, then the AI engine has failed to find a solution in the knowledge database, and the failure represents a computer problem that is new to the tool. *Id.* at 4:66–5:3. Accordingly, if the maintenance tool does not find a solution in the knowledge database, the tool preferably saves the state of the computer system and knowledge database to a location where the state and the database can be examined by a human computer expert. *Id.* at 4:66–5:8.

B. The Claims Involved on Remand

Of the involved claims on remand, claim 6 is independent and reproduced below, with numbered limitations and the limitation at issue italicized. Each of claims 8 and 14 depends from claim 6.

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

- [1] detecting a problem in the computer system;
- [2] activating an AI engine in response to the problem detection;
- [3] utilizing, by the AI engine, selected ones of a plurality of sensors to gather information about the computer system;

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

[5] when a likely solution cannot be determined, *saving a state of the computer system*.

C. Claim Construction

In the Final Decision, we did not explicitly construe any claim term. Final Dec. 5. Furthermore, the Federal Circuit's remand decision does not warrant a claim construction for purposes of this decision.

D. The Asserted Prior Art

1. Overview of Barnett

Barnett discloses a fault management system that includes a configuration manager that maintains configuration information of components used in the distributed system. Ex. 1005, Abstract. The fault management system includes a plurality of rules written according to the configuration information stored therein. *Id.* The fault management system also includes 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. *Id.* at 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. *Id.* at Abstract, 2:4–9. Barnett's diagnostic system includes a rule based AI engine having a rules database. *Id.* at Fig. 2. According to Barnett, a primary objective of the invention is to provide a management system that understands abstract relationships between components (i.e., processes) and has rules written according to these abstract relationships. *Id.* at 1:50–54. Another objective of the invention is to provide a management system that

understands the meta-model and model of the distributed system and has rules based on the meta-model relationship. *Id.* at 1:55–58.

2. *Overview of Allen '664*

Allen '664 describes a case-based-reasoning system for an application field that includes a rule base of inferential rules suited to that field and a case base of exemplary cases, which are notable in that field, and a database of relevant problem data for that field. Ex. 1006 at 2:45–49. For example, in a case-based-reasoning system for loan approval, the rule base may include rules about the creditworthiness of the customer, the case base may comprise exemplary cases that are known to match common fact scenarios, and the database may comprise relevant problem data such as loan payment histories, results of loan-officer interviews and credit-check data. *Id.* at 2:51–59.

E. *Claim 6*

In remanding this case, the Federal Circuit explained that we are to answer only one question: whether Barnett and Allen, considered in combination, would have taught or suggested to a person of ordinary skill in the art “saving a state of the computer system,” as the second half of claim step 5 of independent claim 6 requires. *EMC*, 686 F. Appx. 859, 863–64. We determine that the combination of Barnett and Allen '664 teaches or suggests this limitation.

Petitioner acknowledges that Barnett is silent on what happens when its rule-based AI engine fails to determine a likely solution to the fault. Pet. 25; Pet. Reply 5. In its Reply, Petitioner argues that “it is both inherent and obvious that *Barnett* teaches saving a state of the computer system when a likely solution to a computer problem cannot be determined.” Pet. Reply 4

(emphasis in original). Specifically, Petitioner argues that “it was well known at the time of the ’839 Patent that saving the state of the computer system at the point when a failure occurs is an effective means to debug and improve software and it would have been obvious to include this capability when implementing Barnett.” Pet. 25 (citing Ex. 1007 ¶¶ 73–74 (sic) (Petitioner erroneously cites to ¶ 784)).

Patent Owner characterizes Barnett as using “learned configuration and performance information to identify faults and provide solutions for those faults.” PO Resp. 6. According to Barnett, “[t]he configuration of any computer can be learned because the information is known to the computer.” Ex. 1006, 6:50–52. Thus, states Barnett, “this fault management system can be used to analyze the performance of any computer system, and make recommendations on ways to improve the performance of the system without requiring any modifications of the rules.” *Id.* at 6:50–56. Based on this disclosure, Patent Owner argues that Barnett does not “teach or suggest [the outcome] when the [diagnostic] system cannot determine a likely solution to a fault.” PO Resp. 6. Patent Owner avers that

this would appear to be a situation that is not truly contemplated by *Barnett* inasmuch as the system described by *Barnett* is purportedly capable of learning any necessary configuration information to analyze the performance of the subject computer system, and making recommendations on ways to improve the performance of that system without requiring any modifications of its rules.

Id. at 6–7 (citing Ex. 1005, 6:50–56) (emphasis in original).

We have reviewed Patent Owner’s argument and evidence and determine that Petitioner has shown by a preponderance of the evidence that Barnett teaches the limitation at issue. As an initial matter, Patent Owner

does not submit expert testimony as to how a skilled artisan would understand Barnett. *Id.* In addition, as Petitioner argues, “*Barnett* explicitly contemplates an embodiment in which the diagnostic system *would initially fail to determine a likely solution to a fault*, but in which its configuration manager nonetheless remains capable of introducing ‘new component types, which require new models and meta-models.’” Pet. Reply 6 (citing Ex. 1005, 6:31–37) (emphasis added). Specifically, Barnett states “[s]ince the model of the distributed system is maintained by the configuration manager, and because the rules in the diagnostic system apply to classes of components, it is possible to introduce new component types which require new models and meta-models.” Ex. 1005, 6:31–35. Dr. Sacerdoti, Petitioner’s expert, states that a skilled artisan “would understand this citation to mean that, absent the encoding of models and meta-models describing the new component types, the diagnostic system would be unable to derive a precise diagnosis.” Ex. 1024 ¶ 18. Dr. Sacerdoti further opines that a skilled artisan “would know that introducing new component types would be facilitated by saving the state of the distributed system when a likely solution cannot be found.” *Id.*

Barnett also describes another embodiment:

For example, the fault and performance analysis can be customized to particular applications and provide precise isolation of the faulty component by adding additional relationships in the abstract, or meta-model, and by adding additional rules related to these new classes of components. Therefore, this invention can identify performance problems to a coarse level with little effort, and to a fine level with additional customization.

Ex. 1005, 6:66–7:6. According to Dr. Sacerdoti, “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 [] 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 in which the failure took place.” Ex. 1007 ¶ 75. Dr. Sacerdoti further states that “[because] 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.” *Id.* With regard to saving a state of the computer system, Dr. Sacerdoti, also opines as follows:

It is known in the art that an effective means of enabling the debugging and improvement of virtually any software program is to save the state of the computer system at the point when a failure occurs. If an embodiment of the Barnett patent were not to have such a capability, it would be obvious to a skilled artisan to include this capability when implementing the invention disclosed in Barnett.

Id. ¶ 74.

We are persuaded by Dr. Sacerdoti’s testimony and determine that Petitioner has demonstrated that it was known that an effective means of enabling the debugging and improvement of almost any software program was to save the state of the computer system when a fault (i.e., failure) occurred, and a skilled artisan would have applied this teaching to Barnett’s fault management system. Accordingly, we find that Barnett alone teaches or suggests “saving a state of the computer system.”

Petitioner also refers to Allen '664 for the “saving a state of the computer” limitation. Allen '664 teaches that a new case is created when there is no case that is a good match for the case template, i.e., solution. Specifically, Allen '664 teaches that

[i]n the case-matching step **202** or in the best-case step **203**, the inference engine **111** may determine that there is no case **105** [or solution] 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.... [¶] 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**, it may invoke a rule **103** or a procedural structure **117** which creates a new case **105**.

Ex. 1006, 8:21–36. Petitioner contends that a skilled artisan “would have understood [that] Barnett’s diagnostic system, as modified by Allen '664, 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.” Pet. 57, n.16 (citing Ex. 1007 ¶¶ 94–97).

Patent Owner disputes that Allen '664 teaches or suggests “saving a state of the computer system.” According to Patent Owner, Allen '664 “teaches the person of ordinary skill in the art how to integrate rule-based systems (such as that described by *Barnett*) with case-based systems.” PO Resp. 7. Thus, concludes Patent Owner, a skilled artisan would adopt the strategy taught by Allen '664 when no likely solution exists in its case base, which is adding a rule that “crea[tes] a new case using an existing case template and, optionally, user-prescribed information.” *Id.* at 8 (citation omitted), 11–12. The “new case,” according to Patent Owner, “may include

attribute-value pair data from the case template as well as data received from a user.” *Id.* at 8. Patent Owner avers that Allen ’664 does not suggest that “computer state” information is part of the user-provided information or part of the attribute-value information prescribed in the case template. *Id.* at 8–9 (citation omitted); *see id.* at 11–12. According to Patent Owner, the template information is “case criteria.” *Id.* at 9. Patent Owner concludes that “any combination of *Barnett* and *Allen* ’664 would, at best, save case criteria and user-supplied information as a new case and would not save computer state, as required by claim 6.” *Id.*

We have reviewed Patent Owner’s argument and evidence and determine that Petitioner has shown by a preponderance of the evidence that the combination of Allen ’664 teaches and Barnett teaches the limitation at issue. Patent Owner’s argument is not persuasive because it is based improperly on what Allen ’664 teaches individually, and not on the teachings of the combined references. As the Federal Circuit has explained, “[t]he test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference Rather, the test is what the combined teachings of those references would have suggested to those of ordinary skill in the art.” *In re Keller*, 642 F.2d 413, 425 (CCPA 1981).

In the combined system of Barnett and Allen ’664 proposed by Petitioner, a case template would be created, “when there is a fault in the computer system, to search for a case whose symptoms (i.e., the state of the computer system) match the fault.” Pet. Reply 12 (citing Ex.1024 ¶¶ 22–23, 25; Ex. 1006, 8:31–40). As Petitioner reasons, Allen ’664 teaches “saving of the *case template of a problem*,” and thus, the case template would

comprise facts about the state of the faulty computer system. *Id.* at 12–13 (citation omitted) (emphasis added); *see id.* at 13 (Petitioner argues that a skilled artisan would understand that “the case retrieval step in the standard case-based reasoning cycle entails comparing a digital model of the presenting situation with the digital model of each of a collection of cases, and determining if there is a match.”). Petitioner avers, “[t]o match a problem to the cases in the case base [in the combined Barnett and Allen ’664 system], the case template may be constructed for the problem with attribute-value pairs which correspond to notable parameters of the problem.” Pet. Reply 13 (citing Ex. 1006, 5:2–6). Allen ’664 discloses that “[e]ach case **105** may comprise a set of attributes **301**, each of which has a value **302**. Attributes **301** and values **302** are typically manipulated as an attribute-value pair **303**.” Ex. 1006, 4:32–36. Dr. Sacerdoti states that a skilled artisan “would understand that the attributes whose values describe a case template are the same attributes that describe a case that is stored in the case database.” Ex. 1024 ¶ 27. Allen ’664 further discloses that “[t]o match a problem **311** to the cases **105** in the case base **104**, a case template **312** may be constructed for the problem **311** with attribute-value pairs **303**[,] which correspond to notable parameters of the problem **311**.” Ex. 1006, 5:2–6. Thus, we agree with Petitioner that when applying the teachings of Allen ’664 to Barnett, which teaches a fault management for any computer system, the case template includes the “information about the state of the faulty computer system.” *See id.* at 12; Ex. 1024 ¶¶ 22–26. Furthermore, Allen ’664 teaches saving the “case template” of a problem, as a “new case,” for which it cannot find a solution. Specifically, Allen ’664 teaches that:

[T]he inference engine **111** may determine that there is no case **105** 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** (by means of the user interface **118**) what the prescribed action **309** for the case **105** should be. The inference engine **111** may add the new case **105** to the case base **104** and proceed to another problem **311** with an augmented case base **104**.

Ex. 1006, 8:22–30. Thus, we further agree that in the combined Barnett and Allen '664 system, when Allen '664's case template is saved as a “new case,” the “state of the computer system” would also be saved. See Pet. Reply 13–14. Accordingly, we further find that the combination of Barnett and Allen '664 teaches or suggests “saving a state of the computer system.”

Patent Owner further argues that “designing rule-based or hybrid rule- and case-based fault management systems [such as the combined Barnett and Allen '664 system] is not akin to debugging a computer program.” PO Resp. 9. According to Patent Owner, tasks that are suited to debugging programs would not be applied in the context of a rule-based reasoning system. *Id.* at 10. Patent Owner further adds that designing rule-based reasoning systems is unlike “prosaic programming tasks,” is “counter intuitive,” involves rules “expressed in a nonprocedural way, not like programming code,” and “may be beyond the ability of many application programmers.” *Id.* at 10–11. Patent Owner concludes that “there is no suggestion that saving state would, in any way, be a modification that a person of ordinary skill in the art would apply to *Barnett*, especially when *Barnett* never even contemplates a situation where a solution to a problem cannot be found and when *Allen '664* teaches an alternative solution.” *Id.* at 11. Petitioner replies that Patent Owner's “argument is a *non sequitur*

rooted in pure speculation, and reflects an incorrect understanding of how expert-based AI systems are designed.” Pet. Reply 8 (citing Ex. 1024 ¶¶ 32–37).

We are not persuaded by Patent Owner’s argument. As an initial matter, Patent Owner’s arguments concerning what the combined teachings would have suggested to one with ordinary skill in the art, are mere attorney argument unsupported by evidence, such as expert testimony. See PO Resp. 9–12. We are further sufficiently persuaded by Petitioner’s argument that saving the “state of a problem” is “at least” as important for expert systems, whether they are case-based or rule-based systems, such as Barnett, as it is for conventional systems. Pet. Reply 8 (citing Ex. 1024 ¶¶ 13, 32).

Petitioner’s argument is supported by Dr. Sacerdoti’s testimony, which was formed from his “own extensive experience in managing and teaching the development of rule-based expert systems, and from what [he was] aware was the state of the art at the time of the invention of the ’839 patent.” Ex. 1024 ¶ 32. For example, Dr. Sacerdoti opines that “it was widely known in the art that a library of saved test cases was a key component of an expert system development project” and “[b]ecause the test cases for Barnett are states of the computer system, one of skill in the art would know that it was obvious and/or inherent in the development of a system such as Barnett’s to maintain a library of states of the computer system.” *Id.* ¶ 36.

CONCLUSION

We are sufficiently persuaded that the combination of Barnett and Allen ’664 teaches or suggests “saving a state of the computer system.” Specifically, as discussed above, Barnett alone, or the combination of Barnett and Allen ’664, teaches or suggests the aforementioned limitation.

Accordingly, given the Federal Circuit's directive, because we find that this limitation is taught or suggested by the asserted prior art, we determine that independent claim 6, and dependent claims 8 and 14, are unpatentable.

ORDER

Accordingly, it is

ORDERED that claims 6, 8, and 14 are *unpatentable*; and

FURTHER ORDERED that this is a final written decision under 35 U.S.C. § 318(a). Parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Don Daybell
Bas de Blank
Monte M. F. Cooper
Jason Yu
ORRICK, HERRINGTON, & SUTCLIFFE LLP
d2ptabdocket@orrick.com
m2bptabdocket@orrick.com
m26ptabdocket@orrick.com
j2yptabdocket@orrick.com

PATENT OWNER:

Tarek N. Fahmi
ASCENDA LAW GROUP, PC
tarek.fahmi@ascendalaw.com