

NOTE: This disposition is nonprecedential.

**United States Court of Appeals
for the Federal Circuit**

ORACLE AMERICA, INC.,
Appellant

v.

GOOGLE, INC.,
Appellee

2014-1351

Appeal from the United States Patent and Trademark
Office, Patent Trial and Appeal Board in Reexamination
No. 95/001,548.

Decided: March 20, 2015

MARC A. HEARRON, Morrison & Foerster LLP, Wash-
ington, DC, argued for appellant. Also represented by
MEHRAN ARJOMAND, Los Angeles, CA; DEANNE MAYNARD,
Washington, DC; CHRISTOPHER B. EIDE, Palo Alto, CA.

DARYL JOSEFFER, King & Spalding LLP, Washington,
DC, argued for appellee. Also represented by ROBERT T.
NEUFELD, Atlanta, GA; ADAM CONRAD, Charlotte, NC;
SCOTT THOMAS WEINGAERTNER, New York, NY; BRIAN C.
BANNER, Austin, TX .

Before DYK, O'MALLEY, and TARANTO, *Circuit Judges*.

O'MALLEY, *Circuit Judge*.

Google, Inc. ("Google") filed a request for an *inter partes* reexamination of U.S. Patent No. 6,910,205 ("the '205 patent"), arguing, *inter alia*, that the '205 patent is invalid under § 102 as anticipated by Partial Translation, Swedish Institute of Computer Science Technical Report (T93.5), October 1993 ("Magnusson"). The Patent Trial and Appeal Board ("the Board") affirmed the examiner's rejection of the '205 patent as anticipated by Magnusson. Oracle America, Inc. ("Oracle"), the owner of the '205 patent, appeals the Board's rejection. For the reasons explained below, we *affirm-in-part*, *reverse-in-part*, *vacate-in-part*, and *remand*.

I. BACKGROUND

A. The '205 Patent

The Java computer language allows software developers to create programs that are capable of running on any device that has a Java virtual machine ("JVM") installed. This ensures that, once the program is created, it can run on a multitude of different devices—e.g., Windows computers, Apple computers, tablets, smartphones. When a program is written in Java, it is compiled into a series of virtual machine instructions, also referred to in the patent as "bytecodes." Those virtual machine instructions can be distributed to any device that has a JVM installed. The JVM installed on each device interprets those virtual machine instructions to achieve the desired output. Although this is an advantage for interoperability, the JVM must interpret the virtual machine instructions while the program is running. This typically causes the execution to be slower than a program that is compiled directly into native machine instructions.

To help increase the execution speed of Java-like programs—programs that use a virtual machine to interpret virtual machine instructions—the '205 patent discloses a method of replacing some virtual machine instructions with native machine instructions. This allows the JVM to skip the interpretation step for those instructions and purports to speed up the execution of the program. Figure 5 is exemplary:

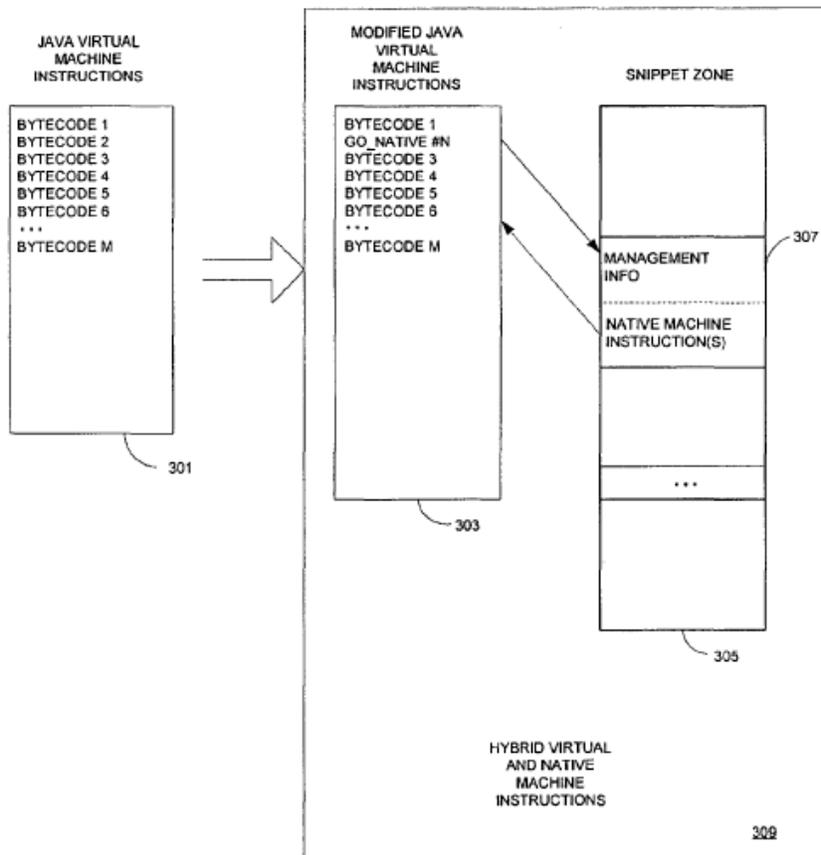


FIG. 5

'205 patent Fig. 5.

The virtual machine instructions **301** are modified such that native instructions replace some virtual machine instructions in the hybrid instructions **309**. In this example, the virtual machine instruction BYTECODE 2 is replaced by the virtual machine instruction “GO_NATIVE #N” in the modified virtual machine instructions **303**. When the JVM reaches the new instruction, rather than interpret BYTECODE 2, it will execute the new GO_NATIVE instruction. This will cause the execution of the corresponding native code in the snippet zone **305**. In this example, the snippet executes native machine instructions that perform the same function as BYTECODES 2–5 (as if the JVM had interpreted those virtual machine instructions). As a result, once the snippet is executed, the program will continue by interpreting BYTECODE 6.

Claims 1 and 2 are indicative of the issues on appeal:

1. In a computer system, a method for increasing the execution of virtual machine instructions at runtime, the method comprising:

receiving a first virtual machine instruction;

generating, at runtime, a new virtual machine instruction that represents or references one or more native instructions that can be executed instead of said first virtual machine instruction; and

executing said new virtual machine instruction instead of said first virtual machine instruction.

2. The method of claim 1, further comprising *overwriting a selected virtual machine instruction with a new virtual machine instruction*, the new virtual machine instruction specifying execution of the at least one native machine instructions.

'205 patent col. 13 ll. 44–58 (emphasis added).

B. The Reexamination Proceedings

On February 17, 2011, Google requested this *inter partes* reexamination, arguing, *inter alia*, that Magnusson anticipated the claims of the '205 patent. The examiner decided that Magnusson presented a substantial new question of patentability and granted the reexamination. After amendment and briefing, the examiner rejected all of the challenged claims as anticipated by Magnusson. On November 27, 2013, the Board affirmed the examiner's rejection. Although the examiner did not construe any claims, the Board construed "overwriting" as "the act of replacing some information in a computer file with new information, rather than literally writing over an existing information." *Google, Inc. v. Oracle Am., Inc.*, No. 2013-010321, 2013 WL 6217845, at *3 (Patent Tr. & App. Bd. Nov. 27, 2013) ("*Board Decision*"). Based on that construction, the Board concluded that Magnusson anticipated the claims that contained the "overwriting" limitation—claims 2–4, 15, 16, and 18–21. The Board also affirmed the examiner's determination that Magnusson was an enabling prior art references for all claims, including the claims that do not have the "overwriting" limitation—claims 1 and 8.

Oracle timely appealed. Because this is an appeal from a final decision of the Board, we have jurisdiction under 28 U.S.C. § 1295(a)(4)(A) (2012).

II. DISCUSSION

We review the Board's legal conclusions, e.g., claim construction, de novo.¹ *In re NTP, Inc.*, 654 F.3d 1268,

¹ After oral argument in this case, the Supreme Court decided *Teva Pharms. USA, Inc. v. Sandoz, Inc.*, 135 S.Ct. 831 (Jan. 20, 2015), which changed our standard of review of a district court's factual determinations in construing a claim. Because the Board did not base its

1273 (Fed. Cir. 2011). We review the Board’s factual findings for “substantial evidence.” *In re Morsa*, 713 F.3d 104, 109 (Fed. Cir. 2013). Substantial evidence is “less than the weight of the evidence but more than a mere scintilla of evidence.” *Id.* “[W]here two different, inconsistent conclusions may reasonably be drawn from the evidence in [the] record, an agency’s decision to favor one conclusion over the other is the epitome of a decision that must be sustained upon review for substantial evidence.” *Id.* (quoting *In re Jolley*, 308 F.3d 1317, 1329 (Fed. Cir. 2002)).

Whether or not a piece of prior art is enabled is a question of law based on underlying factual findings. *Id.* (citing *In re Antor Media Corp.*, 689 F.3d 1282, 1287, 1292 (Fed. Cir. 2012)).

A. Claim Construction

On reexamination, claims are to be “given their broadest reasonable interpretation consistent with the specification.” *In re Yamamoto*, 740 F.2d 1569, 1571 (Fed. Cir. 1984). Claims must be read in light of the specification as it would be interpreted to a person having skill in the art. *In re Am. Academy of Sci. Tech. Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004) (collecting cases).

Although the examiner did not construe any terms, the Board construed “overwriting” to mean “the act of replacing some information in a computer file with new information, rather than literally writing over an existing information.” *Board Decision*, 2013 WL 6217845, at *3. The Board explained that its construction is warranted because “only bytecode 2 has been ‘overwritten’ within the

construction in this case on any findings of fact, and instead only considered intrinsic evidence, we need not consider whether *Teva* also changes our standard of review on appeals from the Board.

modified [JVM] instructions 303, despite the fact that bytecodes 3–5 are also being replaced by one or more native machine instructions.” *Id.*

We conclude that the Board’s construction is erroneous. Although claims are given their “broadest reasonable construction” on reexamination, that construction must be “consistent with the specification.” *Yamamoto*, 740 F.2d at 1571. Based on the clear language in the claims and the specification, “overwriting” means “replacing information in a particular memory location with new information in that location.”

In the example from Figure 5—which the Board relies upon for its construction—although the native machine instructions execute instead of BYTECODES 2–5, only BYTECODE 2 is overwritten or replaced by a new virtual machine instruction—GO_NATIVE #N. The specification explains that only “the initial virtual machine instruction”—in this example, BYTECODE 2—is overwritten. The new virtual machine instruction—GO_NATIVE #N—replaces the old virtual machine instruction in the same memory location as the old virtual machine instruction. ’205 patent col. 8 ll. 2–5. This ensures that the virtual machine will interpret GO_NATIVE #N instead of BYTECODE 2 when it reaches that instruction. *Id.* col. 8 ll. 27–30 (“When the interpreter executes the go_native bytecode, the interpreter will look up the snippet in the snippet zone specified by the go_native bytecode and then activate the native machine instructions in the snippet.”). BYTECODES 3–5, on the other hand, remain in memory, and are, therefore, not overwritten with new information—i.e., a new virtual machine instruction. Indeed, the specification refers to only BYTECODE 2 as being overwritten, not BYTECODES 3–5. *Id.* col. 8 ll. 2–5 (“The interpreter generates modified Java virtual machine instructions 303 by *overwriting bytecode 2* with a go_native virtual machine instructions.” (emphasis added)).

This example is consistent with the remainder of the specification, which consistently discloses that only the first bytecode in the sequence is actually overwritten or replaced. *Id.* col. 7 ll. 27–28; *see id.* col. 7 ll. 36–40 (“At step 209, the virtual machine instruction *at the beginning of the selected portion* of the function is overwritten with a new virtual machine instruction that specifies the execution of one or more native machine instructions of the snippet.” (emphasis added)). The new `go_native` instruction replaces the old bytecode at the same location in memory so that the virtual machine will interpret the `go_native` instruction instead of the old bytecode. *See id.* col. 8 ll. 27–30. While this replacement could occur either by actually writing over the existing information with the new information or by deleting the existing information and inserting the new information, the existing and new information must both appear in the same memory location. *Id.*

Based on the language in the claims and specification, accordingly, we reject the Board’s construction and construe “overwriting” to mean “replacing information in a particular memory location with new information in that location.”² Because the Board’s finding of anticipation was based on an erroneous claim construction, we vacate the Board’s finding that Magnusson anticipates the overwriting claims of the ’205 patent—claims 2–4, 15, 16,

² Although we use the term “information” in our construction, we note that the claims at issue discuss “overwriting a selected virtual machine instruction with a new virtual machine instruction.” ’205 patent col. 13 ll. 54–56. Because neither party addresses the additional limitations in the claims at issue, we do not address them here. The Board, however, will need to address these limitations on remand.

and 18–21—and remand for proceedings consistent with this opinion.

B. Enablement of Magnusson

Because the Board considered enablement based on an erroneous construction of overwriting, we vacate the Board’s enablement determination with respect to the overwriting claims—claims 2–4, 15, 16, and 18–21—and remand for the Board to make the first determination of enablement with respect to those claims. Oracle, however, also argues that Magnusson is not enabled for claims 1 and 8, which do not include the overwriting limitation. In determining whether a prior art reference is enabled, we ask “whether a person of ordinary skill in the art could make or use the claimed invention without undue experimentation based on the disclosure *of that particular document.*” *Morsa*, 713 F.3d at 110 (emphasis in original) (citing 35 U.S.C. § 112).

Oracle argues that, because Magnusson does not explain how the “TRANSLATED” instruction is introduced, it is not enabled with respect to the “generating” and “representing” steps in claims 1 and 8, respectively. *See* ’205 patent col. 13, ll. 48–51 (“[G]enerating, at runtime, a new virtual machine instruction that represents or references one or more native instructions that can be executed instead of said first virtual machine instruction”); *id.* col. 14 ll. 23–25 (“[R]epresenting said at least one native machine instruction with a new virtual machine instruction that is executed after the compiling of the function”). The examiner,³ however,

³ The Board did not provide any analysis of this issue in its opinion; it merely credited the examiner’s analysis. *Board Decision*, 2013 WL 6217845, at *5 (“We have similarly considered the record and we find that the arguments and evidence submitted by [Oracle] are not

extensively reviewed the record and found person of ordinary skill would have known how to introduce the “TRANSLATED” instruction to meet the generating and representing steps. J.A. 31–32. Oracle fails to provide any explanation as to how these findings are not supported by substantial evidence.

Oracle’s remaining arguments are also unpersuasive. Contrary to Oracle’s assertions, the examiner did not base his finding of enablement on a comparison between the disclosure in Magnusson and the disclosure in the ’205 patent. Instead, the examiner merely noted that the specification of the ’205 patent did not even have the level of detail that Oracle was arguing was necessary in Magnusson. *See In re Epstein*, 32 F.3d 1559, 1568 (Fed. Cir. 1994) (“Rather the Board’s observation that appellant did not provide the type of detail in his specification that he now argues is necessary in prior art references supports the Board’s finding that one skilled in the art would have known how to implement the features of the references and would have concluded that the reference disclosures would have been enabling.”). Furthermore, the mere fact that the expert witnesses for both sides were persons of *extraordinary* skill, does not preclude the examiner’s finding that a person of ordinary skill would recognize the errors in Magnusson. Even though Oracle’s expert may have disagreed, we cannot say that the examiner’s determination of the knowledge of a person of ordinary skill is not supported by substantial evidence.

We therefore conclude that Magnusson is an enabling reference with respect to claims 1 and 8 and affirm the Board’s holding that these claims are anticipated.

sufficient to overcome the presumption of enablement relied upon by the Examiner.”).

III. CONCLUSION

For the foregoing reasons, we reverse the Board's construction of "overwriting" and vacate its anticipation and prior art enablement conclusions based on its incorrect construction—for claims 2–4, 15, 16, and 18–21. We remand this case for proceedings consistent with this opinion. With respect to claims 1 and 8, which do not have the "overwriting" limitation, we agree with the Board that Magnusson is an enabling prior art reference. We therefore affirm the Board's holding that these two claims are anticipated.

**AFFIRMED IN PART, REVERSED IN PART,
VACATED IN PART, REMANDED**