

The opinion in support of the decision being entered today was *not* written for publication and is *not* binding precedent of the Board.

Paper No. 17

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte VIKTORS BERSTIS

Appeal No. 2002-1774
Application No. 09/159,955

ON BRIEF

Before BARRETT, GROSS, and BARRY, *Administrative Patent Judges*.
BARRY, *Administrative Patent Judge*.

DECISION ON APPEAL

A patent examiner rejected claims 1, 4-11, 14-24, 26, and 27. (Examiner's Answer at 2.) The appellant appeals therefrom under 35 U.S.C. § 134(a). We reverse.

BACKGROUND

The appellant's invention concerns the execution of Java application programs.

alternately decodes and interprets bytecodes. The bytecodes are designed to be easy to interpret on any computer and easily translated on the fly into native machine code.

(Id.)

A just-in-time (“JIT”) compiler increases the performance of a Java program by compiling methods into native machine code before the code is executed. *(Id.)*

Although native machine code provides for increased execution speeds, the appellant explains that the code consumes more storage than Java bytecodes. *(Id. at 4.)* He adds that the JIT compiler takes time to compile programs and consumes storage itself.

(Id.)

In contrast, the appellant’s invention collects statistics on the frequency that code paths in the bytecode of class methods are executed and uses the statistics to select paths that are pre-compiled into native machine code. The combination of bytecodes for little executed paths and native machine code for frequently executed paths are sent to a client for execution. According to the appellant, the need for a JIT compiler on a client is reduced or eliminated thereby saving storage space and JIT compiling time.

to execute at near native speeds while offering reduced code size for infrequently executed code. (*Id.*)

A further understanding of the invention can be achieved by reading the following claim:

1. A method in a distributed data processing system for optimizing execution of a class, the method comprising:

identifying a frequency of execution of paths in interpreted instructions for methods for the class within the distributed data processing system;

compiling a portion of the paths into precompiled native machine code for the data processing system using the frequency of execution of paths; and

sending the class to a client in the distributed data processing system.

Claims 1, 4-11, 14-20, 24, 26, and 27¹ stand rejected under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 6,044,220 ("Breternitz"). Claims 21-23 stand rejected

under § 103(a) as obvious over Breternitz in view of U.S. Patent No. 5,809,512 ("Kato").²

OPINION

Rather than reiterate the positions of the appellant or examiner *in toto*, we address the main point of contention therebetween. The appellant argues, "[b]ecause *Breternitz* is directed to **translation** of machine instructions from a first instruction set to a second instruction set, *Breternitz* does not teach or suggest compiling a portion of the paths into precompiled native machine code for the data processing system using the frequency of execution of paths." (Reply Br. at 5.) The "[e]xaminer disagrees with Appellant's argument that there is no teaching of *Breternitz* that a portion of paths are compiled into a precompiled native code for a data processing system being a compiled portion of the paths that are executed more frequently than other portions of the paths." (Examiner's Answer at 8.) He makes the following assertions.

Breternitz clearly disclose this limitation as "combine interpretive execution with translation. This combined approach **uses interpretive execution for low-frequency instructions** and translates natively those instruction sequences that take up most of the execution time. The combined approach achieves the low overhead in code size while allowing for the

the frequency exceeds a threshold . . . the idiom is translated into its optimized native code. . ." at line 62 of col. 5 to line 13 of col. 6.

(*Id.*)

"Analysis begins with a key legal question -- *what* is the invention *claimed*?" *Panduit Corp. v. Dennison Mfg. Co.*, 810 F.2d 1561, 1567, 1 USPQ2d 1593, 1597 (Fed. Cir. 1987). In answering the question, "words of the claim are generally given their ordinary and accustomed meaning. . . ." *In re Paulsen*, 30 F.3d 1475, 1480, 31 USPQ2d 1671, 1674 (Fed. Cir. 1994) (citing *Carroll Touch, Inc. v. Electro Mechanical Sys., Inc.*, 15 F.3d 1573, 1577, 27 USPQ2d 1836, 1840 (Fed. Cir. 1993)).

Here, independent claims 1, 11, 26, and 27 specify in pertinent part the following limitations: "identifying a frequency of execution of paths in interpreted instructions" and "compiling a portion of the paths into precompiled native machine code. . . ." Similarly, independent claims 10, 20, and 24 specify in pertinent part the following limitations: "compiling the portion of the paths in the bytecodes. . . ." Also similarly, independent claim 21 specifies in pertinent part the following limitations: "compiles the portion of the

meaning, the limitations require translating a portion of paths of a computer program **expressed in a high-order language** into its machine language equivalent.

Having determined what subject matter is being claimed, the next inquiry is whether the subject matter is obvious. "In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness." *In re Rijckaert*, 9 F.3d 1531, 1532, 28 USPQ2d 1955, 1956 (Fed. Cir. 1993)(citing *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992)). "A *prima facie* case of obviousness is established when the teachings from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art." *In re Bell*, 991 F.2d 781, 783, 26 USPQ2d 1529, 1531 (Fed. Cir. 1993) (quoting *In re Rinehart*, 531 F.2d 1048, 1051, 189 USPQ 143, 147 (CCPA 1976)).

Here, although Breternitz uses the word "translation," col. 1, l. 59, the examiner fails to show that the reference's invention translates a computer program expressed in a high-order language into its machine language equivalent. To the contrary, the

translates machine instructions “from [a] foreign architecture into a sequence of native,”
col. 1, l. 24-25, machine instructions.

The examiner fails to allege, let alone show, that the addition of Kato cures the deficiency of Breternitz. Absent a teaching or suggestion of translating a portion of paths of a computer program expressed in a high-order language into its machine language equivalent, he fails to present a *prima facie* case of obviousness as to the limitations of compiling interpreted instructions into native machine code. Therefore, we reverse the obviousness rejections of claims 1, 4-11, 14-24, 26, and 27.

CONCLUSION

In summary, the rejections of claims 1, 4-11, 14-24, 26, and 27 under § 103(a) are reversed.



REVERSED

LEE E. BARRETT
Administrative Patent Judge

ANITA PELLMAN GROSS
Administrative Patent Judge

LANCE LEONARD BARRY
Administrative Patent Judge

)
)
)
)
)
) BOARD OF PATENT
) APPEALS
) AND
) INTERFERENCES
)
)
)
)

Appeal No. 2002-1774
Application No. 09/159,955

Page 9

DUKE W YEE
CARSTENS YEE & CAHOON
PO BOX 802334
DALLAS, TX 75380