

The opinion in support of the decision being entered today was not written for publication and is not binding precedent of the Board.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. 25

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte GLEN S. BURKE and GAIL ZACHARIAS

Appeal No. 1997-2467
Application No. 08/509,753

ON BRIEF

Before FLEMING, GROSS, and BARRY, Administrative Patent Judges.
BARRY, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the rejection of claims 1-4 and 11-18. We reverse.

BACKGROUND

The invention at issue in this appeal relates to computer programming language constructs. Many computer languages used to develop programs are static languages. Programs written in such languages cannot be modified without recompilation. Dynamic languages, in contrast, allow certain actions to be

taken at runtime that could not be accomplished with a static language.

Common Lisp (CL) is a dynamic language. It features three types of functional objects: generic functions, ordinary functions, and methods. An ordinary function cannot be added to a generic function. A method cannot be invoked outside a generic function, except in the very restricted context of method combination, and then only using specialized syntax. Besides these restrictions, however, ordinary functions and methods provide the same functionality to a user. The distinction therebetween complicates CL.

The appellants' invention aims to allow a functional object in a dynamic language to be used as both a directly-invoked ordinary function and as a method in a generic function. Their method object contiguously comprises a header, a method-information field, a function prolog field, and a method body. When called as a component of a generic function, an entry point is at the method body or at the method-information section. When called directly, another entry point exists at the function prolog field. By combining

the concepts of ordinary functions and methods, their
construct simplifies language semantics.

Claim 15, which is representative for our purposes,
follows:

15. An arrangement for using a method object in
accordance with a dynamic object-oriented
programming language of a computer, said arrangement
comprising:

a memory for storing the method object, the method
object comprising:

a header field having a starting memory address;

a method-information field located at a first offset
from the starting memory address, said method-information
field containing a predetermined address value;

a function-prolog field of code located at a second
offset from the starting memory address; and

a method body field located at said predetermined
address value from the starting memory address, said
method body field containing program code for execution
by said computer, and

a processor for invoking said method object in
response to entering said method object at one of a first
entry point and a second entry point, wherein said first
entry point is different from said second entry point.

The prior art applied in rejecting the claims follows:

Patton et al. (Patton) 4,868,745 Sep. 19, 1989

Keene, Object-Oriented Programming in Common LISP 8, 9,
66-115 (1989).

Claims 1-4 and 11-18 stand rejected under 35 U.S.C. § 103 as being obvious over Patton in view of Keene. Rather than reiterate the arguments of the appellants or examiner in toto, we refer the reader to the brief and answer for the respective details thereof.

OPINION

After considering the record, we are persuaded that the examiner erred in rejecting claims 1-4 and 11-18.

Accordingly, we reverse. We begin by noting the following principles from

In re Rijckaert, 9 F.3d 1531, 1532, 28 USPQ2d 1955, 1956 (Fed. Cir. 1993).

In rejecting claims under 35 U.S.C. § 103, the examiner bears the initial burden of presenting a prima facie case of obviousness. 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, 782, 26 USPQ2d 1529, 1531 (Fed. Cir. 1993) (quoting In re Rinehart, 531 F.2d 1048, 1051, 189 USPQ 143, 147 (CCPA 1976)).

With these principles in mind, we consider the examiner's rejection and the appellants' argument.

The examiner asserts, "Patton et al teach entering said object at one of a first entry point to directly invoke said body in response to execution of the ordinary function call (column 4, lines 60 - 65) and a second entry point to invoke said method body in the context of a generic function dispatch in response to execution of the generic function call (column 4, lines 55 - 60), wherein said first entry point is different from said second entry point" (Examiner's Answer at 4.) The appellants argue, "the combination of Patton and Keene neither discloses nor suggests an apparatus or method for entering a method object at one of **two different** entry points when invoking the method object." (Appeal Br. at 9.)

Claims 1-3 and 18 specify in pertinent part the following limitations: "entering said method object at one of a first

entry point to directly invoke said method body in response to execution of the ordinary function call and a second entry point to invoke said method body in the context of a generic function dispatch in response to execution of the generic function call, wherein said first entry point is different from said second entry point" Similarly, claims 4 and 11-13 specify in pertinent part the following limitations: "(c) directly invoking the method object by entering the method object at a first entry point at the function-prolog field in response to execution of the ordinary function call ... and (d) invoking the method object in the context of a generic function dispatch by entering the method object at a second entry point at the method-information field in response to execution of the generic function call" Also similarly, claim 14 specifies in pertinent part the following limitations: "means for directly invoking said method object by entering the method object at a first entry point at the function-prolog field in response to execution of the ordinary function call ... and means for invoking said method object in the context of a generic function dispatch by entering the method object at a second entry point at said method -

information field in response to execution of the generic function call" Further similarly, claims 15-17 specify in pertinent part the following limitations: "a processor for invoking said method object in response to entering said method object at one of a first entry point and a second entry point, wherein said first entry point is different from said second entry point." Accordingly, claims 1-4 and 11-18 require entering a method object at one of two different entry points when invoking the object.

The examiner fails to show a teaching or suggestion of the limitations in the prior art of record. Patton teaches that "objects are provided with a uniform structure so that all objects can be indifferently executed, either directly or indirectly. Objects of different types are still uniformly structured. Each comprises at least a prologue address and a body. These two parts are contiguous in memory. The prologue address addresses a prologue which describes execution of the object. The body is data." Col. 1, ll. 62-68. Although the reference discloses an object that can be executed directly and executed indirectly, the two types of executions do not

enter the object at different entry points. To the contrary, the object is entered at the same entry point, viz., the prologue address thereof. Specifically, "[i]n the direct execution of an object, an interpreter pointer points to the prologue address of an object.... In indirect execution of an object the interpreter pointer points to an object pointer which in turn points to or addresses the prologue address of an object" Col. 2, 11.

6-13. Relying on Keene to "teach[] that a method object is a type of object for the purpose of performing a function[,]" (Examiner's Answer at 5), the examiner fails to show that the secondary reference cures the deficiency of Patton.

Because Patton points to the prologue address of an object for both direct and indirect execution thereof, we are not persuaded that the reference discloses or would have suggested the limitations of "entering said method object at one of a first entry point to directly invoke said method body in response to execution of the ordinary function call and a second entry point to invoke said method body in the context of a generic function dispatch in response to execution of the

generic function call, wherein said first entry point is different from said second entry point[;]" "(c) directly invoking the method object by entering the method object at a first entry point at the function-prolog field in response to execution of the ordinary function call ... and (d) invoking the method object in the context of a generic function dispatch by entering the method object at a second entry point at the method-information field in response to execution of the generic function call[;]" "means for directly invoking said method object by entering the method object at a first entry point at the function-prolog field in response to execution of the ordinary function call ... and means for invoking said method object in the context of a generic function dispatch by entering the method object at a second entry point at said method information field in response to execution of the generic function call[;]" and "a processor for invoking said method object in response to entering said method object at one of a first entry point and a second entry point, wherein said first entry point is different from said second entry point." Therefore, we reverse the rejection of

claims 1-4 and 11-18 as being obvious over Patton in view of Keene.

CONCLUSION

In summary, the rejection of claims 1-4 and 11-18 under § 103 is reversed.

REVERSED

| | | |
|-----------------------------|---|-----------------|
| MICHAEL R. FLEMING |) | |
| Administrative Patent Judge |) | |
| |) | |
| |) | |
| |) | |
| |) | BOARD OF PATENT |
| ANITA PELLMAN GROSS |) | APPEALS |
| Administrative Patent Judge |) | AND |
| |) | INTERFERENCES |
| |) | |
| |) | |
| |) | |
| LANCE LEONARD BARRY |) | |
| Administrative Patent Judge |) | |

LLB/dal

Appeal No. 1997-2467
Application No. 08/509,753

Page 11

CHARLES J. BARBAS
CESARI AND MCKENNA
30 ROWES WHARF
BOSTON, MA 02110