

THIS OPINION WAS NOT WRITTEN FOR PUBLICATION

The opinion in support of the decision being entered today
(1) was not written for publication in a law journal and
(2) is not binding precedent of the Board.

Paper No. 10

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte BRUCE E. CHAPA

Appeal No. 1997-3104
Application 08/221,124¹

ON BRIEF

Before BARRETT, FLEMING and GROSS, **Administrative Patent
Judges**.

FLEMING, **Administrative Patent Judge**.

DECISION ON APPEAL

¹ Application for patent filed March 30, 1994.

Appeal No. 1997-3104
Application 08/221,124

This is a decision on appeal from the final rejection of claims 1 through 6, all the claims pending in the present application.

The invention relates to remote application primary interface (API) processing methods using network protocols in a network which includes personal computers and a server. In particular, Appellant discloses on page 1 of the specification that an API is a collection of commands which enables a programmer to obtain services from an application. Appellant discloses further that networks typically include personal computers which are connected to a server. The server may handle tasks, such as electronic mail and electronic faxing for the network. Personal computers have limited resources and, as a result, such personal computers are not capable of executing applications that require large amounts of extended memory. On page 2 of the specification, Appellant discloses that it would be desirable to provide a method for networking personal computers in which memory-intensive tasks are off-

Appeal No. 1997-3104
Application 08/221,124

loaded to a server which will have the required amount of large extended memory.

On page 3 of the specification, Appellant discloses that figure 4 is a flow diagram of the method of the present invention. On page 7 of the specification, Appellant discloses the method of the present invention referring to figure 4. On pages 7 through 9 of the specification, Appellant discloses in detail the method steps in which the personal computer is able to off-load API requests generated by an application program to a server.

Independent claim 1 is reproduced as follows:

1. A method for processing an API request generated by an application program that is executing in a client terminal by a server computer networked to the client terminal comprising the steps of:

(a) allocating a block of operating memory by the client terminal;

(b) placing data defining the client terminal API request in the block of operating memory by the client terminal;

Appeal No. 1997-3104
Application 08/221,124

(c) sending the data and instructions for creating the block of memory to the server computer by the client terminal;

(d) allocating a second block of operating memory like the first block of operating memory by the server computer using the instructions received from the client terminal;

(e) storing the data from the client terminal in the second block of operating memory by the server computer;

(f) processing the client terminal API request and storing API response data in the second block of memory by an API service routine executed by the server computer;

(g) sending the API response data and instructions for creating the second block of memory to the client terminal by the server computer;

(h) storing the API response data from the server computer in the one block of operating memory by the client terminal; and

(i) processing the API response data from the one block of operating memory by the application program.

The reference relied on by the Examiner is as follows:

Moeller et al. (Moeller)	5,473,777	Dec. 5,
1995		

Claims 1 through 6 stand rejected under 35 U.S.C. § 102 as being anticipated by Moeller.

Appeal No. 1997-3104
Application 08/221,124

Rather than repeat the arguments of Appellant or the Examiner, we make reference to the brief and answer for details thereof.

OPINION

After careful review of the evidence before us, we do not agree with the Examiner that claims 1 through 6 are anticipated by Moeller.

It is axiomatic that anticipation of a claim under § 102 can be found only if the prior art reference discloses every element of the claim. **See *In re King***, 801 F.2d 1324, 1326, 231 USPQ 136, 138 (Fed. Cir. 1986) and ***Lindemann Maschinenfabrik GMBH v. American Hoist & Derrick Co.***, 730 F.2d 1452, 1458, 221 USPQ 481, 485 (Fed. Cir. 1984). "Anticipation is established only when a single prior art reference discloses, expressly or under principles of inherency, each and every element of a claimed invention." ***RCA Corp. v. Applied Digital Data Sys.***,

Appeal No. 1997-3104
Application 08/221,124

Inc., 730 F.2d 1440, 1444, 221 USPQ 385, 388 (Fed. Cir.),
cert. dismissed, 468 U.S. 1228 (1984), *citing Kalman v.*
Kimberly-Clark Corp., 713 F.2d 760, 772, 218 USPQ 781, 789
(Fed. Cir. 1983), *cert. denied*, 465 U.S. 1026 (1984).

Furthermore, "[t]o establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.'" *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) *citing Continental Can Co. v. Monsanto Co.*, 948 F.3d 1264, 1268, 20 USPQ2d 1746, 1749 (Fed. Cir. 1991). "Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." *Id.* at 1269, 20 USPQ2d at 1749 (*quoting In re Oelrich*, 666 F.2d 578, 581, 212 USPQ 323, 326 (CCPA 1981)).

Appeal No. 1997-3104
Application 08/221,124

On page 5 of the brief, Appellant argues that Moeller does not disclose a method of remotely processing an API request

by a server rather than a client terminal. Appellant argues that the term "server" clearly refers to a computer which services an API request instead of the client terminal that executes the application which produces the API request. On pages 7 and 8 of the brief, Appellant further argues that Moeller does not inherently teach a method of remotely processing an API request by a server rather than a client terminal.

We note that Appellant's claim 1 recites the method for processing an API request generated by an application program that is executed in a client terminal. Furthermore, we note that Appellant's claim 1 recites: allocating a block of operating memory by the client terminal; placing the data and defining the client terminal API request in the block of operating memory by the client terminal; sending the data and instructions for creating the block of memory to the server

Appeal No. 1997-3104
Application 08/221,124

computer by the client terminal; allocating a second block of operating memory by the server computer using the instructions received from the client terminal; storing the data from the client terminal in a second block of operating memory by the server computer; processing the client terminal API request; sending the API response data and instructions to the client terminal by the server computer;

storing the API response data from the server computer in the one block of operating memory by the client terminal; and processing the API response data from the one block of operating memory by the application program. We further note that Appellant claims similarly in the other remaining independent claim, claim 6, a method for processing an API request generated by an application program that is executing on a first computer by a second computer networked to the first computer. Therefore, we find that the Appellant's claims recite a method of remotely processing an API request generated by an application program that is executing on a

Appeal No. 1997-3104
Application 08/221,124

client terminal by a server coupled to the client terminal using network protocols that can be transported between various operating systems.

Upon a careful review of Moeller, we find that Moeller is directed to an apparatus for enabling an object-oriented application to access in an object-oriented manner a procedural operating system having a naive procedural interface. See Abstract. Furthermore, Appellant discloses that figure 1 illustrates a block diagram of a computer platform in which a wrapper of the present invention operates. See column 4, lines 34 through 37. Moeller further discloses that figure 1

illustrates a block diagram of a computer platform 102 which includes a single central processing unit (CPU) 106. See column 5, lines 16 through 23. Moeller further teaches that it should be noted that the CPU 106 may represent a single processor, but preferably represents multiple processors operating in parallel. See column 5, lines 23 through 25.

Appeal No. 1997-3104
Application 08/221,124

However, we fail to find that Moeller teaches a method for processing an API request generated by an application program that is executing in a client terminal by a server as recited in Appellant's claims 1 through 6.

In the Examiner's answer, the Examiner argues that the method claimed by the Appellant is inherent to Moeller. However, to establish inherency, the Examiner must show that the missing descriptive matter is necessarily present in the system described in the reference. We fail to find that the Examiner has met this burden. In particular, Moeller's system may be operated with the processors in parallel but we fail to find the method in which one processor will interchange memory data and instructions with another processor in the particular method as recited in Appellant's claim. Furthermore, we fail to find that Moeller must operate in this way to perform his disclosed wrapper for enabling an object-oriented application to maintain virtual memory using procedural function calls.

Appeal No. 1997-3104
Application 08/221,124

In view of the foregoing, the decision of the
Examiner rejecting claims 1 through 6 is reversed.

REVERSED

	LEE E. BARRETT)	
	Administrative Patent Judge)	
)	
)	
)	BOARD OF
PATENT)	
	MICHAEL R. FLEMING)	APPEALS AND
	Administrative Patent Judge)	
INTERFERENCES)	
)	
)	
)	
	ANITA PELLMAN GROSS)	
	Administrative Patent Judge)	

MRF:psb

Appeal No. 1997-3104
Application 08/221,124

Paul W. Martin
Law Department, ECD-2
101 W. Schantz Avenue
Dayton, OH 45479