

The opinion in support of the decision being entered today was not written for publication and is not binding precedent of the Board.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte DANIEL OTTO BECKER

Appeal No. 2006-0051
Application No. 09/583,519

ON BRIEF

Before JERRY SMITH, BLANKENSHIP, and MACDONALD, Administrative Patent Judges.

BLANKENSHIP, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the examiner's final rejection of claims 1-5, 7, 9, 10, 12, 13, 15-20, 22, and 24, which are all the claims remaining in the application.

We affirm-in-part.

BACKGROUND

The invention relates to distributed data processing systems and in particular to client-server data processing systems. Commands within a messaging system are given a name which matches the filename for the class file implementing the command. A command name may be passed to a server and used to dynamically load and instantiate a message command by loading a class of the same name. Claim 1 is reproduced below.

1. A method for generating a command in a computer messaging system, comprising the steps of:

receiving a command name at a server via said messaging system, said command name being sent by a client to initiate a particular one of multiple server functions identified by an executable having a name synonymous with said command name;

utilizing said command name to automatically load a class file having a name including said command name; and

dynamically executing functions on said server associated with said class file.

The examiner relies on the following reference:

Tyra et al. (Tyra)	US 6,442,565 B1	Aug. 27, 2002 (filed Aug. 13, 1999)
--------------------	-----------------	--

Claims 1-5, 7, 9, 10, 12, 13, 15-20, 22, and 24 stand rejected under 35 U.S.C. § 102 as being anticipated by Tyra.

We refer to the Final Rejection (mailed Oct. 29, 2003) and the Examiner's Answer (mailed Aug. 26, 2004) for a statement of the examiner's position and to the

Appeal No. 2006-0051
Application No. 09/583,519

Brief (filed Jun. 1, 2004) and the Reply Brief (filed Sep. 27, 2004) for appellant's position with respect to the claims which stand rejected.

OPINION

Appellant submits (Brief at 3) that the claims stand or fall together. Appellant contends that instant claim 1 is an "exemplary" claim (id. at 4), and presents arguments with respect to the language of that claim. Our initial focus will thus be on claim 1. We note that we have considered only appellant's arguments presented in the briefs, notwithstanding the indication that arguments not included in the briefs have been incorporated by reference (e.g., id.). See 37 CFR § 1.192(a) (2003) ("Any arguments or authorities not included in the brief will be refused consideration by the Board of Patent Appeals and Interferences, unless good cause is shown."); 37 CFR § 41.37(c)(1)(vii) (Sep. 13, 2004) ("Any arguments or authorities not included in the brief or a reply brief filed pursuant to § 41.41 will be refused consideration by the Board, unless good cause is shown.").

The examiner sets forth, at pages 4 and 5 of the Answer, how Tyra is deemed to meet the limitations of representative claim 1. According to appellant, however, Tyra's server operates solely as a repository of data, which transmits the data back to the requesting client and other systems on the network. Tyra's transmission method does not execute functions of a class file at the server. The "executing functions" language in

Appeal No. 2006-0051
Application No. 09/583,519

claim 1 refers to implementation of the function performed as a characteristic of the class file, as opposed to operations by the server to locate the class file. (Brief at 4-5.)

The examiner responds that Tyra's system executes functions on the server associated with the class file, such as retrieving the identifier from the request, manipulating the identifier, and comparing the manipulated identifier with available class files to locate the class file. The located class file is then loaded, executed, or implemented to achieve the results -- executing a function to perform an update for a particular stock price -- which are then transmitted to the client. The examiner refers to column 2, lines 61 through 67 and column 17, lines 25 through 65 of the reference. The examiner submits that a "function" is defined as "a software routine or procedure, which performs a special task or operation." Further, the examiner responds that the feature of "executing functions" referring specifically to executing the source code associated with the class file, as opposed to performing the server operations of locating and/or manipulating the class file, is not recited in the rejected claims. (Answer at 9-10.)

Appellant responds, in turn, that the claim language (e.g., "functions") must be read in the context of the claims. Instant claim 1, for example, recites initiating a particular server function "identified by an executable having a name synonymous with said command name" received at the server. In Tyra, in contradistinction, the name is a file name and the processing operations are performed on that file name, rather than executing the underlying or intrinsic function attributed to the executable process associated with that file name. (Reply Brief at 2-3.)

The statement of the rejection indicates (Answer at 4-5) that the argued limitations are disclosed by Tyra in the Abstract, Figure 29 and its corresponding text, and material at columns 2 and 17.

Tyra describes an apparatus and method whereby files in a network are downloaded, along with changes in a format of files. A request for a file including an identifier for the file is received. A search for the file is performed by manipulating the identifier and comparing the manipulated identifier with available files. A located file and associated changes in a format of the file are subsequently downloaded. Tyra col. 2, ll. 59-67. In particular, a client application makes use of a class, either with an explicit reference or a class name (Fig. 29, step 2901). If JVM (JAVA virtual machine) interpreter 2801 (Fig. 28) determines that a particular class has not been loaded, it makes a load request to JOD (JAVA On Demand) class loader 2803. JOD class loader 2803 may search for the class and, if found, load the class into memory (Fig. 29, step 2910), and the class is delivered to the application (step 2911). In either case (i.e., including the case that the class is presently loaded in the JVM interpreter), the class is returned to the application (Step 2911). Tyra col. 17, ll. 25-64.

In our understanding of the teachings of Tyra, we do not find disclosure of a command name sent by a client "to initiate a particular one of multiple server functions identified by an executable having a name synonymous with said command name," as required by instant claim 1. In Tyra, the command name sent by the client refers to, the examiner notes (Answer at 5 and 9), the update of particular stock prices or quotes.

The search for the class in the reference is not described as being identified by an executable having a name synonymous with the class name. Rather, the class name is the target of the search, and thus unrelated to initiation of the functions required in performing the search for the class.

We are not persuaded by appellant (e.g., Brief at 4), however, that the claim 1 recitation of “dynamically executing functions on said server associated with said class file” requires that functions identified by an executable having a name synonymous with the command name be performed on the server. The claim 1 language may be read as dynamically executing functions on the server that is associated with the class file. That is, the language “associated with said class file” may relate to the “server,” rather than the “functions” as appellant seems to argue. Claim 1 does not recite, “dynamically executing functions, on said server, associated with said class file.” Cf. Margaret Shertzer, The Elements of Grammar, MacMillan Publishing Co. 1986 (p. 85, Rule 22(5) “Use commas to set off descriptive phrases following the noun they modify.”).

Instant claim 5 recites that “said selected command triggers an activation and execution at said network system of functions associated with said class. . . .” The rejection (Answer at 7) relates the limitation to the searching and transmitting of the file, but we do not find disclosure of “loading/executing” of the file in the indicated portions of the reference. The rejection (id. at 8) submits that claim 7 is rejected “under the same rationale” as claims 1 and 2. The rejection thus does not show, as recited in claim 7, “locating a class file having an executable class filename that is substantially similar to

said command name,” and “loading and instantiating said class file and functions provided thereby. . . .”

Instant claim 9 requires means for, inter alia, “instantiating functions of said class file on said server.” The rejection (Answer at 8) refers to the rejection of claims 3 through 5. Instant claim 4 recites, “instantiating functions of said class file on said server.” The rejection of claim 4 (Answer at 6) relates the limitation to Figure 28 and column 17, lines 1 through 24 of Tyra. We do not find disclosure of the limitation in the indicated portions of the reference. Further, the rejection of claim 22 (Answer at 8) fails to show wherein “said command name triggers an execution of a related function of said class when received at a recipient computer system. . . .”

We find, however, that appellant’s arguments in support of “exemplary” claim 1 are not commensurate with the scope of independent claims 13 and 17. Instant claim 17 recites a computer program product with instructions “for utilizing said command name to automatically load a class file having a name including said command name” (e.g., Tyra’s disclosure of request for, and loading of, a class by name). The claim further recites instructions for “dynamically executing functions associated with said class file after said class file is loaded.” The claim does not specify where the functions may be executed, and thus does not distinguish over the functions being performed in a client system that requested the class file. Instant claim 13 recites means for “dynamically executing functions associated with said class file after said class file is loaded,” which again does not specify where the functions may be executed. Tyra

Appeal No. 2006-0051
Application No. 09/583,519

discloses that, in addition to files, software modules may be distributed by the system (e.g., col. 16, ll. 48-67). We therefore sustain the rejection of claims 13 and 17, and of depending claims 15, 16, and 18 through 20, as the claims are not separately argued by appellant.

For the foregoing reasons, we thus sustain the § 102 rejection of claims 13 and 15 through 20 over Tyra, but do not sustain the § 102 rejection of claims 1 through 5, 7, 9, 10, 12, 22, and 24.

CONCLUSION

The rejection of claims 1-5, 7, 9, 10, 12, 13, 15-20, 22, and 24 under 35 U.S.C. § 102 as being anticipated by Tyra is affirmed with respect to claims 13 and 15-20, but reversed with respect to claims 1-5, 7, 9, 10, 12, 22, and 24.

Appeal No. 2006-0051
Application No. 09/583,519

No time period for taking any subsequent action in connection with this appeal may be extended under 37 CFR § 1.136(a). See 37 CFR § 1.136(a)(1)(iv).

AFFIRMED - IN - PART

JERRY SMITH)	
Administrative Patent Judge)	
)	
)	
)	
)	BOARD OF PATENT
HOWARD B. BLANKENSHIP)	APPEALS
Administrative Patent Judge)	AND
)	INTERFERENCES
)	
)	
)	
ALLEN R. MACDONALD)	
Administrative Patent Judge)	

Appeal No. 2006-0051
Application No. 09/583,519

DILLON & YUDELL LLP
8911 NORTH CAPITAL OF TEXAS HIGHWAY
SUITE 2110
AUSTIN, TX 78759