

The opinion in support of the decision being entered today was not written for publication and is not binding precedent of the Board.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. 22

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte ROBERT PAPPAS

Appeal No. 2002-1868
Application 09/059,033¹

ON BRIEF

Before HAIRSTON, BARRETT, and NAPPI, Administrative Patent Judges.

BARRETT, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the final rejection of claims 1-19.

We affirm-in-part.

¹ Application for patent filed April 13, 1998, entitled "Copying a Sequence of Commands to a Macro."

BACKGROUND

The invention relates to producing a macro file for use in an application program by transferring user commands from a log file containing a history of user commands to the macro file.

Claim 1 is reproduced below.

1. A method of producing a macro file for use in an application program comprising:

storing each of a plurality of user commands, applied to a data file opened in the application program, into a log file, the log file maintaining a real time sequential history of each of said plurality of user commands; and

transferring under user control at least one of the stored user commands from the log file to the macro file.

THE REFERENCES

The examiner relies on the following references:

Yuasa et al. (Yuasa)	5,448,736	September 5, 1995
Dwyer et al. (Dwyer)	5,706,457	January 6, 1998
Kurtenbach	5,867,163	February 2, 1999 (filed October 6, 1997)
Clark et al. (Clark)	5,970,064	October 19, 1999 (filed June 12, 1997)

THE REJECTIONS

Claims 1-3 and 10-14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Yuasa, Dwyer, and Clark.

Claims 4-9 and 15-19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Yuasa, Dwyer, Clark, and Kurtenbach.

We refer to the final rejection (Paper No. 14) (pages referred to as "FR__") and the examiner's answer (Paper No. 20)

Appeal No. 2002-1868
Application 09/059,033

for a statement of the examiner's rejection, and to the brief (Paper No. 19) (pages referred to as "Br__") for a statement of appellant's arguments thereagainst.

OPINION

Claims 1-3 and 12-14

The examiner finds that Yuasa teaches a method of producing a macro file for use in an application program, but does not disclose "transferring under user control at least one of the stored user commands from the log file to the macro file" (FR3). The examiner finds that Dwyer discloses this limitation because each icon 30 launches a set of linked macro functions that initializes the modem, controls data transfer, formats the received image data, etc. (FR3). The examiner further finds (FR3) that the combination of Yuasa and Dryer does not disclose "the log file maintaining a real time sequential history of each of said plurality of user commands." The examiner finds that Clark discloses the limitation for admission control in a communications network, referring to the abstract and column 2, lines 5-59, and column 17, lines 7-50 (FR3). The examiner concludes that it would have been obvious to modify Yuasa to apply transfer of user commands to a macro file as taught by Dwyer and to add a log file maintaining a real time sequential history as taught by Clark because "the system would provide the better reliable tool to end users by preformat [sic] prior

Appeal No. 2002-1868
Application 09/059,033

transferring linked macro of log files's command based on user's command in term of real time factor" (FR4).

Appellant argues that "[w]hile arguable Yuasa teaches some type of storing of commands and conventional macro generation, Yuasa does not teach the combination of storing commands in a log file and transferring the commands under user control from a log file to a macro" (Br6-7) and that the examiner admits that Yuasa does not describe transferring commands under user control (Br7). It is argued that Dwyer's "operator interaction" only launches macro functions and does not move commands from a log file into a macro file (Br7). It is argued that the examiner's reliance on Clark for teaching a log file maintaining a real-time sequential history of user commands is unreasonable because storing policy data in telecommunications networks would not lead one to provide a log file maintaining a real-time sequence of user commands for the claimed purpose of transferring commands under user control to the macro file (Br7). Appellant argues that the rejection fails to provide motivation for combining the references to render obvious the claimed invention (Br7-8).

The examiner expands on the teachings of the references (EA18-22), but we do not consider these responses to address appellant's arguments.

Dwyer and Clark do not support the examiner's rejection. Dwyer discloses a system and method for acquiring and archiving

images. The system receives and stores digitized images (abstract). A key aspect of the system is that the operator is allowed to perform only a predetermined set of image processing functions that correspond to the graphical icons 30, where each of the icons 30 launches a set of linked macro functions that perform a variety of tasks (abstract; col. 4, lines 15-30). We agree with appellant that the operator in Dwyer merely launches macros using the icons 30 and does not move commands from a log file to a macro file as found by the examiner. We also do not understand the examiner's reasoning regarding Clark as applied to the limitation of "the log file maintaining a real time sequential history of each of said plurality of user commands." Clark does mention the words "real time" in the title, but Clark is a machine system for controlling admission of communications connection to a communications network (col. 2, lines 30-35). We find no description of user commands, much less maintaining a sequential history of user commands. The network controller in Clark is a machine, not a human, and the admission control policy data is data stored in a switch, not user commands. The examiner states that "[t]he policy data is input into connection algorithms and admission algorithms operated by the switch which implement in real time connection (see column 8, lines 5-19) and that is a log file maintaining a real-time sequential history of user commands" (EA22). The examiner's reasoning is obscure and

does not point out the sequential history of user commands. Lastly, we do not understand how the stated motivation that "the system [of the combination] would provide the better reliable tool to end users by preformat [sic] prior transferring linked macro of log files's command based on user's command in term of real time factor" (FR4) results in the claimed invention or find any support for the motivation in the record.

Nevertheless, despite the examiner's findings to the contrary, it appears that claim 1 would have been obvious over Yuasa alone. Yuasa discloses a "command recall function" where the recorded key and/or mouse operations are called "macros," which are programs (col. 1, lines 10-52). The command recall function in various languages have problems, such as it being impossible to directly specify an arbitrary command or value, the commands record any operator errors, the chaining relationships between commands are not attained, etc. (col. 2, line 27 to col. 3, line 3). Yuasa provides a method of generating a program (macro) in which the user may use a portion of a group of commands which have been inputted by the operator (col. 3, line 36 to col. 4, line 46). In the Lisp programming language a command is called an S-expression and an execution of a command is expressed as an evaluation of an S-expression and a result of an execution of a command is called a value (col. 6, lines 29-32). The operator display 301 in Fig. 12 has a listener

Appeal No. 2002-1868
Application 09/059,033

portion 303 for the user to enter an S-expression, i.e., "user commands," a talker 302, which is a window for sequentially presenting the S-expressions inputted in the listener 303 and values obtained by evaluating the S-expression, and a finder 305 which is a window for displaying the inputted S-expressions in the form of icons (col. 6, lines 49-57). An example is described based on the employee list database 2500 in Fig. 24, which corresponds to the claimed "data file opened in the application program." S-expressions are entered in the listener 303 and when the carriage return (CR) key 315 is depressed, the S-expression with an identifier ("@" plus a number) appears in the talker 302 and an icon is displayed in the finder 305 (Figs. 26 & 27; col. 15, lines 29-52). Objects are generated in the computer (col. 15, lines 63-65). More S-expressions are inputted, displayed with identifiers in the talker, evaluated, and displayed as icons (e.g., Figs. 28-46). The talker 302 maintains the sequential history of S-expressions (commands noted by identifiers "@__") applied to the database. The sequence of S-expressions (commands) entered by the user and stored in the talker 302 meets the step of "storing each of a plurality of user commands, applied to a data file opened in the application program, into a log file, the log file maintaining a real time sequential history of each of said plurality of user commands."

Yuasa discloses several embodiments for "transferring under user control at least one of the stored user commands from the log file to the macro file." In embodiment 1, a program is generated having functions equivalent to the entire sequence of the command strings (col. 21, lines 58-60; col. 22, lines 48-52); in embodiment 2, a program based on a portion of the sequence of command strings is generated by specifying an end point to the chaining relationship (col. 22, lines 52-55; col. 23, lines 1-7); in embodiment 3, an intermediate portion of a sequence of commands may be generated as a program by specifying change points (col. 25, lines 2-6; col. 25, line 45 to col. 26, line 56); in figure 5, the operator can arbitrarily specify a plurality of execution results to automatically generate a program equivalent to sequences of command series (col. 30, lines 48-54). Yuasa discloses that "[i]tems such as commands previously inputted by the operator can be specified thereafter as commands or operands in a subsequent operation by use of identifiers, icons, function keys, or the like" (col. 34, lines 13-16). While it appears that claim 1 is anticipated by Yuasa, the rejection is maintained under § 103.

Thus, we sustain the rejection of claims 1-3 and 12-14 based on Yuasa alone.

Claim 10

Claim 10 recites: "The method of claim 1, wherein applying commands further comprises: storing information for the command that is used to produce a macro corresponding to the command."

The examiner admits that Yuasa does not disclose this limitation (FR6; EA7-8). The examiner finds that Dwyer teaches this limitation because the transfer DAT tape to album icon 33d calls a macro that performs the function (FR6; EA8) and concludes that it would have been obvious to store information for the command that is used to produce a macro in Yuasa because, "[b]y doing so, the system would enhance and provide a friendly tool to end users in terms of information storage while performing [sic, performing] another task" (FR7; EA8).

Appellant argues that the examiner admits that Yuasa does not describe claim 10 and Dwyer's discussion pertains to a tool that transfers images in digitized format to an album and "[t]his does not suggest information used to form a macro" (Br8).

The examiner expands on the teachings of Dwyer (EA23-24), but we do not consider this to address appellant's argument.

We do not understand the examiner's rationale or how the macro in Dwyer corresponds to the claim limitation of storing information that is used to produce a macro. The statement of motivation does not provide motivation for the specific modification, but is merely a general benefit to be achieved.

Appeal No. 2002-1868
Application 09/059,033

Given the statement of motivation and Dwyer, we do not see how one of ordinary skill in the art would arrive at the claimed invention. The motivation is not supported by the record.

Nevertheless, despite the examiner's finding to the contrary, we do not see how claim 10 distinguishes over Yuasa. Yuasa allows transfer under user control of all or some of the commands used to produce the command recall (macro) function. Information for the commands must inherently be stored at least during the time the user is creating the command recall (macro) function from the commands; it is certainly stored somewhere so it can be displayed on talker 302 for selection by the operator. And, the macro commands in Yuasa are stored for subsequent operation (col. 34, lines 13-16). Thus, we sustain the rejection of claim 10 over Yuasa alone.

Claim 11

Claim 11 recites: "The method of claim 1, wherein transferring commands further comprises: adding the macro data corresponding to the transferred command to the macro file."

The examiner admits that Yuasa does not disclose this limitation (FR7; EA8-9). The examiner finds that Dwyer teaches this limitation because it is determined whether or not a text file is added to the album 38 (FR7; EA9). The examiner concludes that it would have been obvious to modify Yuasa so that the step

Appeal No. 2002-1868
Application 09/059,033

of transferring commands would include adding the macro data corresponding to the transferred command because, "[b]y doing so, the system would enhance and provide better tools to end users in terms of building macro file structure" (FR7-8; EA9).

Appellant argues that the examiner admits that Yuasa does not describe claim 11 and Dwyer's discussion pertains to a tool that transfers images in digitized format to an album and "[t]his does not suggest adding the macro data corresponding to the transferred command to the macro file" (Br8).

The examiner expands on the teachings of Dwyer (EA24-25), but we do not consider this to address appellant's argument.

We do not understand the examiner's rationale or how determining whether a text file is added to the album in Dwyer corresponds to the claim limitation at issue. The statement of motivation is also unpersuasive.

Nevertheless, we do not see how claim 11 distinguishes over Yuasa. Yuasa allows transfer under user control of all or some of the commands used to produce the command recall (macro) function. When the commands are transferred to create the command recall (macro) file, macro data corresponding to the command must be added to the macro file to create the macro. Thus, we sustain the rejection of claim 11 over Yuasa alone.

Claims 4-9 and 15-18

Claim 4 recites: "The method of claim 1 wherein the log file is represented as a history palette that stores commands."

The examiner finds that the combination of Yuasa, Dwyer, and Clark do not disclose the limitation (FR8; EA10). The examiner finds that Kurtenbach discloses a log file represented as a history palette (FR8; EA10). The examiner concludes that it would have been obvious to include a log file represented as a history palette in the combination of Yuasa, Dwyer, and Clark because "[b]y doing so, the system would provide the better improved GUI that preserves the intuitive and simplicity of the tool shelf/palette paradigm to end users" (FR9; EA10).

Appellant argues that Kurtenbach does not suggest a log file represented as a history palette (Br9)

The examiner expands on the teachings of Kurtenbach (EA26).

Kurtenbach discloses a method of defining and automatically executing a sequence of commands (abstract). The user places (drags and drops) desktop icons from a tool box into a sequence tool shelf in an order in which the commands are to be executed (abstract). It is difficult to understand the examiner's rationale for combining the references since the examiner finds that Yuasa does not even teach a log file. Thus, the examiner is modifying a reference which has already been modified by two references, which suggests hindsight. While the tool shelf in

Appeal No. 2002-1868
Application 09/059,033

Kurtenbach seems to correspond to a history palette because the icons represent commands that will be executed in sequence, and while the tool palette and tool shelf in Kurtenbach appear to correspond to the tool palette 252 and history palette 256, respectively, in appellant's Fig. 13, this teaching is difficult to apply to Yuasa. The commands in Yuasa have to be specified in a particular way, partly because of the nature of the Lisp language, and thus do not lend themselves to being selected from a tool palette or represented as a history palette. The rejection fails to present reasons to establish a prima facie case of obviousness. The rejection of claims 4-9 and 15-18 is reversed.

Claim 19

Claim 19 recites: "The computer program of claim 12 wherein when the commands are transferred, they are transferred to an action palette that builds the macro file."

The examiner finds that the combination of Yuasa, Dwyer, and Clark do not disclose the limitation (FR13; EA16). The examiner finds that Kurtenbach discloses icons representing commands being dragged to an action (FR13; EA16). The examiner concludes that it would have been obvious to include icons representing commands being dragged to an action palette in the combination of Yuasa, Dwyer, and Clark because "[b]y doing so, the system would enhance

Appeal No. 2002-1868
Application 09/059,033

and provide better friendly technique which preserves the intuitive and simplicity of the tool shelf/palette paradigm to end users" (FR14; EA17).

Appellant argues that the examiner errs because the rejection relies on the same reasoning and analysis as for claim 4 even though claim 4 recited a history palette and claim 19 recites an action palette (Br9).

The examiner expands on the teachings of Kurtenbach and notes that Kurtenbach not only teaches a history palette, but also teaches an action palette (EA26).

We reverse the examiner's rejection for similar reasons to those stated with respect to claim 4. The rejection is modifying a reference which has already been modified by two references, which suggests hindsight. Moreover, it is not exactly clear how the references are being combined. While the tool shelf in Kurtenbach seems to correspond to a history palette (or log) because the icons represent commands that will be executed in sequence, there is no teaching of transferring commands from a history log to an action palette. The rejection fails to establish a prima facie case of obviousness. The rejection of claim 19 is reversed.

Appeal No. 2002-1868
Application 09/059,033

CONCLUSION

The rejection of claims 1-3 and 10-14 is sustained.

The rejection of claims 4-9 and 15-19 is reversed.

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

AFFIRMED-IN-PART

KENNETH W. HAIRSTON)	
Administrative Patent Judge)	
)	
)	
)	
)	BOARD OF PATENT
LEE E. BARRETT)	APPEALS
Administrative Patent Judge)	AND
)	INTERFERENCES
)	
)	
)	
ROBERT NAPPI)	
Administrative Patent Judge)	

Appeal No. 2002-1868
Application 09/059,033

FISH & RICHARDSON P.C.
3300 DAIN RAUSCHER PLAZA
MINNEAPOLIS, MN 55402