

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opinion in support of the decision being entered today (1) was not written for publication in a law journal and (2) is not binding precedent of the Board.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. 18

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte DONALD V. ALECCI,
ROSS J. HILBERT,
and RICHARD J. SMOLUCHA

Appeal No. 1997-1654
Application No. 08/431,307¹

ON BRIEF

Before JERRY SMITH, BARRETT, and BARRY, Administrative Patent Judges.

BARRY, Administrative Patent Judge.

¹ The application was filed on April 28, 1995. It is a continuation of Application Serial No. 08/164,616, which was filed on December 9, 1993, and is now abandoned. The latter application was a continuation of Application Serial No. 08/004,488, which was filed on January 14, 1993 and is now abandoned. The latter application was a continuation of Application Serial No. 07/742,149, which was filed on August 1, 1991 and is now abandoned. The latter application was a continuation of Application Serial No. 07/468,435, which was filed on January 22, 1990 and is now abandoned.

DECISION ON APPEAL

This is a decision on the appeal under 35 U.S.C. § 134 from the final rejection of claims 26-51. We affirm-in-part.

BACKGROUND

The invention at issue in this appeal relates to a computer program that creates data objects associated with computer graphic windows. Such a program can create many objects. A conventional computer that stores one copy of default attributes and one copy of overriding attributes per object accordingly requires a substantial amount of memory to store attribute values beyond whatever amount is needed for other storage. An improved, conventional computer reduces storage requirements by enabling a computer program to operate effectively while maintaining fewer copies of the default attributes. Even the improved computer, however, stores multiple copies of the overriding attributes.

In contrast, the invention enables a computer program to operate while storing only one copy of overriding attributes for multiple objects. Because it can be used with known

methods for reducing the number of copies of default attributes, the invention substantially reduces storage requirements for a computer program that use data objects with default attributes and overriding attributes.

Claim 26, which is representative for our purposes, follows:

26. In a computer system, a method of reducing internal memory requirements of a program during execution of the program, said method comprising the steps of:

specifying a set of default attributes prior to execution of the program;

specifying a set of one or more overriding attributes prior to execution of the program, said overriding attributes corresponding to selected default attributes such that at least one default attribute has no corresponding overriding attribute;

starting execution of the program;

storing values for each of the default attributes in a first internal memory of the computer system during execution of the program;

storing values for each of the overriding attributes in a separate second internal memory of the computer system during execution of the program; and

building a set of final attributes from the default attributes and the overriding attributes during execution of the program.

The references relied on in rejecting the claims follow:

David S. H. Rosenthal, "Going For Baroque", pp. 71-79, June 1988

Beatrice Lam, "The NewWave Office", Vol. 40, No. 4, pp. 23-31, August 1989.

Claims 26-51 stand rejected under 35 U.S.C. § 103 as obvious over Rosenthal in view of Lam. Rather than repeat the arguments of the appellants or examiner in toto, we refer the reader to the briefs and answer for the respective details thereof.

OPINION

In reaching our decision in this appeal, we considered the subject matter on appeal and the rejection and evidence advanced by the examiner. Furthermore, we duly considered the arguments of the appellants and examiner. After considering the totality of the record, we are not persuaded that the examiner erred in rejecting claims 26-29, 31-39, 41-47, and

49-51. We are persuaded, however, that the examiner erred in rejecting claims 30, 40, and 48. Accordingly, we affirm-in-part. Our opinion addresses the grouping and obviousness of the claims.

Grouping of the Claims

37 C.F.R. § 1.192(c)(7), as amended at 60 Fed. Reg. 14518 (Mar. 17, 1995), was controlling when the appeal brief was filed. Section 1.192(c)(7) stated as follows.

For each ground of rejection which appellant contests and which applies to a group of two or more claims, the Board shall select a single claim from the group and shall decide the appeal as to the ground of rejection on the basis of that claim alone unless a statement is included that the claims of the group do not stand or fall together and ... appellant explains why the claims of the group are believed to be separately patentable. Merely pointing out differences in what the claims cover is not an argument as to why the claims are separately patentable.

In addition, claims that are not argued separately stand or fall together. In re Kaslow, 707 F.2d 1366, 1376, 217 USPQ 1089, 1096 (Fed. Cir. 1983). When the patentability of dependent claims in particular is not argued separately, the claims stand or fall with the claims from which they depend.

In re King, 801 F.2d 1324, 1325, 231 USPQ 136, 137 (Fed. Cir. 1986); In re Sernaker, 702 F.2d 989, 991, 217 USPQ 1, 3 (Fed. Cir. 1983).

The appellants fail to explain why dependent claims 27, 28 and 31-37 and independent claim 47, which are subject to the same rejection as independent claim 26, are believed to be separately patentable; why dependent claims 41-46, which are subject to the same rejection as independent claim 38, are believed to be separately patentable; or why dependent claims 50 and 51, which are subject to the same rejection as independent claim 49, are believed to be separately patentable. Therefore, claims 26-28, 31-37, and 47 stand or fall together, with claim 26 as the representative claim of the group; claims 38 and 41-46 stand or fall together, with claim 38 as the representative claim of the group; and claims 49-51 stand or fall together, with claim 49 as the representative claim of the group. Next, we address the obviousness of the claims.

Obviousness of the Claims

We begin by finding that the references represent the level of ordinary skill in the art. See In re GPAC Inc., 57 F.3d 1573, 1579, 35 USPQ2d 1116, 1121 (Fed. Cir. 1995) (finding that the Board of Patent Appeals and Interference did not err in concluding that the level of ordinary skill in the art was best determined by the references of record); In re Oelrich, 579 F.2d 86, 91, 198 USPQ 210, 214 (CCPA 1978) ("[T]he PTO usually must evaluate ... the level of ordinary skill solely on the cold words of the literature."). Of course, every patent application and reference relies on the knowledge of persons skilled in the art to complement its disclosure. In re Bode, 550 F.2d 656, 660, 193 USPQ 12, 16 (CCPA 1977). Such persons must be presumed to know something about the art apart from what the references teach. In re Jacoby, 309 F.2d 513, 516, 135 USPQ 317, 319 (CCPA 1962).

We address the obviousness of the claims in the following groups: claims 26-28, 31-37, and 47; claims 38 and 41-46; claims 29 and 39; claims 30, 40, and 48; and claims 49-51.

Claims 26-28, 31-37, and 47

Regarding claims 26-28, 31-37, and 47, the appellants make three arguments. We address these seriatim. First, the appellants argue, "Rosenthal teaches neither separate default attributes and overriding attributes" (Appeal Br. at 14.) The examiner replies, "it is implicit that a subobject containing default and overriding attributes as per Rosenthal will acquire these attributes from separate sources." (Examiner's Answer at 4.) We agree with the examiner.

Claims 26-28, 31-37, and 47 each specify in pertinent part the following limitations:

- specifying a set of default attributes ...;
- specifying a set of one or more overriding attributes ...;
- ...
- storing values for each of the default attributes in a first internal memory ...;
- storing values for each of the overriding attributes in a separate second internal memory
-

Giving the limitations their broadest reasonable interpretation, the claims each recite separate default attributes and overriding attributes.

The appellants err in considering Rosenthal in less than its entirety. A reference must be considered as a whole for what it reveals "to workers in the art." Panduit Corp. v. Dennison Mfg. Co., 810 F.2d 1561, 1566, 1 USPQ2d 1593, 1595 (Fed. Cir. 1987). Such workers, moreover, must be presumed to know something about the art apart from what the references teach. Jacoby, 309 F.2d at 516, 135 USPQ at 319.

Here, the appellants admit that Rosenthal "teaches a way to specify default values." (Appeal Br. at 7.) This teaching would have suggested "specifying a set of default attributes" They also admit that the reference recognizes "that the ability to override defaults is a worthy goal." (Id.) Rosenthal, however, teaches much more.

Specifically, the reference teaches a "'Hello, World,'" string in the text of a computer program. Fig. 7. Rosenthal also teaches creating "a Label Widget to display the string, overriding the defaults database to set the Label's value to the string to display." P. 78. Because the string is used to

override the defaults, the teachings would have suggested "specifying a set of one or more overriding attributes"

Furthermore, workers in the art would have known that when a computer program is run on a computer, the computer loads the program into its internal memory. Lam evidences this knowledge, for example, by teaching that although a computer "code is made up of multiple code segments ... only those segments in use are kept in memory. As other segments are needed, they are read from [a] disc." P. 24. Accordingly, when Rosenthal's computer program is run on a computer, the computer loads the program into its internal memory. Because the reference's program specifies "a set of default attributes" and "a set of one or more overriding attributes" as aforementioned, the computer stores the values of these sets in its internal memory when it loads the program. Workers in the art also would have known that each line of the program is stored in a separate memory location. Accordingly, when Rosenthal's computer program is loaded, the computer stores the default attributes and the overriding attributes in separate memory locations. These teachings

would have suggested "storing values for each of the default attributes in a first internal memory" and "storing values for each of the overriding attributes in a separate second internal memory"

Second, the appellants argue that Rosenthal does not teach "the claimed building step." (Appeal Br. at 14.) The examiner makes the following reply.

Certain default attributes are defined when a toplevel object is created and other overriding attributes are defined at the creation of a subordinate instance of that toplevel object. Thus, ... Rosenthal indeed "'requires' the building step of claim 26" when new subobjects are launched. (Examiner's Answer at 4.)

We agree with the examiner. Claims 26-28, 31-37, and 47 each specify in pertinent part the "building [of] a set of final attributes from the default attributes and the overriding attributes"

The appellants again err in considering Rosenthal in less than its entirety. As aforementioned, Rosenthal creates a Label Widget to display the "Hello, World" string. The

reference overrides "the defaults database to set the Label's value to the string to display." P. 78. It "creates an X11 window for each Widget, setting its attributes from the data in the Widget." P. 78. More specifically, "[t]he default attributes for the Label Widget specify that the text is centered, and the default mechanism supplies a suitable font." PP. 78-79. "In the same way, the default mechanism supplies background and foreground colors for the Widget." P. 79. In summary, Rosenthal creates a window by superimposing the overriding "Hello, World" string on default fonts and colors. Workers in the art would have known that the window is represented by a set of final attributes. These teachings would have suggested "building a set of final attributes from the default attributes and the overriding attributes"

Third, the appellants argue, "Lam fails to teach a method for maintaining a single copy of data, such as a single copy of the overriding attribute values." (Appeal Br. at 14.) More specifically, they assert that Lam fails to teach maintaining only a single copy of the overriding attribute values. The examiner replies, "The 'data' of the claimed

'attributes' functions to achieve an effect within a program object in a manner reading on Lam's 'code', which also determines properties, or attributes, of a NewWave Office object." (Examiner's Answer at 4-5.) We disagree with the appellants.

The appellants err by attempting to read limitations from the specification into the claims. "In the patentability context, claims are to be given their broadest reasonable interpretations. Moreover, limitations are not to be read into the claims from the specification." In re Van Geuns, 988 F.2d 1181, 1184, 26 USPQ2d 1057, 1059 (Fed. Cir. 1993) (internal citations omitted).

Here, the claims do not recite the limitation of maintaining only a single copy of the overriding attribute values. Accordingly, the appellants' arguments concerning the limitation are immaterial. Therefore, we affirm the rejection of claims 26-28, 31-37, and 47. Next, we address the obviousness of claims 38 and 41-46.

Claims 38 and 41-46

Regarding claims 38 and 41-46, the appellants rely on the arguments advanced for claims 26-28, 31-37, and 47. We rejected these arguments for the aforementioned reasons. The pertinent limitations of claims 38 and 41-46 are substantially similar to those of claims 26-28, 31-37, and 47. Accordingly, we reject the arguments as applied to claims 26-28, 31-37, and 47 for the same reasons.

Further regarding claims 38 and 41-46, the appellants make the following argument.

Rosenthal does not state that membership in an attribute set is frozen once the program begins running, and the use of a dynamically accessed database creates the possibility that membership in a set of attributes could change while the program runs. By contrast, the Constant Membership Claims expressly require no change in membership of the set of overriding attributes during program execution. (Appeal Br. at 17.)

The examiner replies, "a Rosenthal user need not cause any 'attribute' modifications while executing the widget, and a set of 'overriding attributes' thereby left unchanged, or 'constant', is sufficient to read upon the claimed invention." (Examiner's Answer at 6.) We agree with examiner.

Claims 38 and 41-46 each specifies in pertinent part that "membership in the set of overriding attributes remains constant during execution of the program." Although we agree with the appellants that "the use of a dynamically accessed database creates the *possibility* that membership in a set of attributes *could* change while the program runs," (Appeal Br. at 17 (emphasis added)), such a change is not required. Because the possibility is not a certainty, the reference would have suggested "membership in the set of overriding attributes remains constant during execution of the program." Therefore, we affirm the rejection of claims 38 and 41-46. Next, we address the obviousness of claims 29 and 39.

Claims 29 and 39

Regarding claims 29 and 39, the appellants argue, "the references of record do not teach the claimed modification of source code to assign predetermined values to overriding attributes." (Appeal Br. at 15.) The examiner makes the following reply.

However, as is particularly evident in Lam, a modified collection of shared code segments, which impart "attributes" to a NewWave object instance, is, specified when the object is created and executed. Similarly, the "code" which serves as a "source" for defining a Rosenthal widget is necessarily modified when a subordinate instance of the widget is formed, using "overriding attributes". (Examiner's Answer at 5.)

We agree with the examiner.

Claims 29 and 39 each merely specifies that "said step of assigning a predetermined value to each of the overriding attributes comprises modifying a source code corresponding to the program." Rosenthal teaches a "Hello, World" computer program. P. 71. Insertion of lines of code specifying overriding attributes, i.e., the "Hello, World" string, into the program modifies the original program. This insertion would have suggested that "said step of assigning a predetermined value to each of the overriding attributes comprises modifying a source code corresponding to the program." Therefore, we affirm the rejections of claims 29 and 39. Next, we address the obviousness of claims 30, 40, and 48.

Claims 30, 40, and 48

Regarding claims 30, 40, and 48, the appellants argue, "The Office Action fails to identify any specific text, source code, or other portion of the cited references which discusses scratch memory." (Appeal Br. at 16.) The examiner makes the following reply.

The 'specific text' appellant might refer to appears at Lam, page 24, col 2: [t]he code is made up of multiple code segments, and only those segments in use are kept in memory. As other segments are needed, they are read in from disc. This explicit teaching of downloading needed items from separate storage into a temporary, or 'scratch memory' as they are called for then combines with the 'default'/'overriding attribute' arrangement of Rosenthal, in which given subobjects must of necessity assemble final 'attribute' sets before they can be executed. (Examiner's Answer at 6.)

We agree with the appellants.

Claims 30, 40, and 48 require "a scratch memory area" that is distinct from the memory locations where the claimed default attributes and overriding attributes are stored. The examiner errs in interpreting the content of Lam. Although he refers to Lam's memory, as aforementioned, the memory is the main memory area where a computer program is loaded. Lam, p.

24. The reference does not teach the scratch memory as claimed. Rosenthal does not cure this deficiency. Therefore, we reverse the rejections of claims 30, 40, and 48. Next and last, we address the obviousness of claims 49-51.

Claims 49-51

Regarding claims 49-51, the appellants rely on the arguments advanced for claims 26-28, 31-37, and 47. We rejected these arguments for the aforementioned reasons. The pertinent limitations of claims 49-51 are substantially similar to those of claims 26-28, 31-37, and 47. Accordingly, we reject the arguments as applied to claims 49-51 for the same reasons. Therefore, we affirm the rejection of claims 49-51.

We end by noting that the aforementioned affirmances are based only on the arguments made in the briefs. Arguments not raised therein are not before us, are not at issue, and are thus considered waived.

CONCLUSION

To summarize, the examiner's rejection of claims 26-29, 31-39, 41-47, and 49-51 under 35 U.S.C. § 103 is affirmed. His rejection of claims 30, 40, and 48 under 35 U.S.C. § 103, however, is reversed.

No period for taking subsequent action concerning this appeal may be extended under 37 C.F.R. § 1.136(a).

AFFIRMED-IN-PART

JERRY SMITH)	
Administrative Patent Judge)	
)	
)	
)	
)	BOARD OF PATENT
LEE E. BARRETT)	APPEALS
Administrative Patent Judge)	AND
)	INTERFERENCES
)	
)	
)	
LANCE LEONARD BARRY)	
Administrative Patent Judge)	

LLB/kis
John W. Ogilivie
COMPUTER LAW++
8 East Broadway
Suite 725
Salt Lake City, UT 84111