Apple, Inc. v. Motorola, Inc. et al

Doc. 255 Att. 2

Exhibit 34

Dockets.Justia.com

[redacted]

Jcabe U.S. PTO
09/140523
08/26/98

Subclass	5
Class	717
ISSUE CLASSIFICATION	

PATENT NUMBER
627598

BEST COPY

U.S. UTILITY PATENT APPLICATION

O.I.P.E.	PATENT DATE
RM 2 SCANNED Sm Q.A. Sw	AUG 14 2001

SECTOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
	~~395~~ 717	~~706~~ 5	~~2755~~ 2122	[illegible]

FILED WITH: ☐ DISK (CRF) ☐ FICHE
(Attached in pocket on right inside flap)

[redacted]

Certificate 2782
DEC 08 2009
of Correction

PREPARED AND APPROVED FOR ISSUE

ISSUING CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)					
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
717	5						
INTERNATIONAL CLASSIFICATION							
G06F 9/45							

☐ Continued on Issue Slip Inside File Jacket

☑ TERMINAL DISCLAIMER 09/140,523	DRAWINGS			CLAIMS ALLOWED	
	Sheets Drwg.	Figs. Drwg.	Print Fig.	Total Claims	Print Claim for O.G.
	17	17	1	22	1
☐ a) The term of this patent subsequent to ________ (date) has been disclaimed.	John Q. Chavis (Assistant Examiner) 3-14-01 (Date)			NOTICE OF ALLOWANCE MAILED 3-23-01	
☑ b) The term of this patent shall not extend beyond the expiration date of U.S Patent. No. 5,379,432	[illegible] KAKALI CHAKI PRIMARY EXAMINER (Primary Examiner) 3-21-01 (Date)			ISSUE FEE	
				Amount Due 1240.00	Date Paid [illegible]
☐ c) The terminal ____ months of this patent have been disclaimed.	Kim Parrell (Legal Instruments Examiner) 3/23/01 (Date)			ISSUE BATCH NUMBER D89	

WARNING:
The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A
(Rev. 10/97)

(LABEL AREA)

Formal Drawings (shts) set____

________(FACE)

983FH001

#16 C

Docket No. 3048-7035US1 7-5-00

P.2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

TC 2700 MAIL ROOM

JUL -5 2000

RECEIVED

Applicant(s):	Orton et al.	OTLC Docket:	P-046.63
Serial No.:	09/140,523	Group Art Unit:	2762
Filed:	August 26, 1998	Examiner:	J. Chavis
For:	OBJECT-ORIENTED OPERATING SYSTEM		

O I P E JC17 JUN 2 9 2000 PATENT & TRADEMARK OFFICE

PRELIMINARY AMENDMENT

Honorable Assistant Commissioner
of Patents and Trademarks
Washington, D.C. 20231
Sir:

In response to the Office Action of February 1, 2000, please amend the application as follows:

IN THE CLAIMS:

Please AMEND the claims as follows:

Please cancel claim 11 without prejudice.

Please amend claim 12 as follows:

2 ~~12~~. (AMENDED) The computer system of claim [11] ~~16~~ 1, wherein the procedural program logic code further comprises:

D1

procedural program logic code portions specific to each object-oriented method to issue one or ore procedural function calls compatible with the native interface to control the native system services performed by the hardware environment to correspond to the native system services required by the object-oriented method.

63

C

983FH178

Please cancel claims 13 - 15 without prejudice.

Please amend claim 16 as follows:

Sub C1

16. (Amended) A computer system, comprising:

computer hardware for performing native system services;

a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services;

object oriented methods requiring native system services;

procedural program logic code, responsive to invocations of the object-oriented methods, for causing the procedural operating system to control the computer hardware to perform the required native system services;

C2

[The computer system of claim 11, further comprising:]

executable program memory associate with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code;

means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory; and

a runtime loader, responsive to the determinations, to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods.

Please cancel claim 21 without prejudice.

Please amend claim 22 as follows:

C3 22. (AMENDED) A method for operating a computer system, comprising the steps of:

Sub C2

executing a procedural operating system on computer hardware, the procedural operating system including a native interface, responsive to procedural function calls, for providing native system services;

issuing calls, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services;

C3

[The method of claim 21, further comprising the steps of:]

determining if object-oriented methods to be invoked during runtime execution are present in executable program memory associated with the computer hardware; and

selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded.

Please amend claim 25 as follows:

10 ~~25.~~ (AMENDED) The method of claim [21] 7 ~~22~~, wherein the step of issuing calls, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services, further comprises the step of:

C4

adapting the native services provided by the procedural operating system to be compatible with the native system services required by the associated object-oriented method.

Please amend claim 27 as follows:

C5 27. (AMENDED) A ~~method for~~ operating a computer system, comprising the steps of:

[illegible] 23.

64

C

executing a procedural operating system, based on Windows or Unix operating systems, on a Unix or IBM compatible computer hardware environment;

providing an object-oriented interface, executing on the computer hardware environment, and responsive to object-oriented programming, for instantiating objects from object-oriented classes, encapsulating data for exclusive use with each object, and invoking object-oriented methods in the objects for operating on the encapsulated data; [and]

providing procedural programming logic code, responsive to selected ones of said invoked object-oriented methods requiring native system services, for issuing procedural calls, compatible with a native interface of the procedural operating system, to cause the hardware environment to provide the native system services in response to the object-oriented methods[.] ;

and

loading the methods during runtime before invocation thereof;

whereby a choice of which system implementation to use can be deferred to run-time.

REMARKS

Status Of The Claims

Claims 11, 13 - 15, 21 are cancelled without prejudice.

Claims 12, 16-20, 22-30, as amended, remain in the case remain in the case.

Rejection Of The Claims Over The Prior Art

Claims 11-30 are rejected under 35 U.S.C §103 as being unpatentable over the Schmidt, "Systems Programming With C++ Wrappers", C++ Report, October 1992.

Applicants' Response

In response, the Applicant states that Schmidt fails to disclose or suggest loading the method during runtime before invocation thereof, as claimed by the Applicant. The examiner cites the Schmidt reference to show that the applications built using Schmidt's libraries ultimately make system calls at run-time. This is true of all applications, not merely Schmidt's. Ultimately, the actual calls on system functionality get made at runtime in all cases. What the Applicants are claiming is that the claimed invention can defer the decision about which system implementation to use until run time. The Applicant's claimed invention loads the method during runtime just before invocation thereof. Adding the claim element of "loading the method during runtime before invocation thereof" means that (unlike Schmidt) it is possible to wait until the program is running before the particular library is chosen and used by the program. There is nothing in Schmidt that even suggests this claimed feature.

In the claimed invention, the application can be written and compiled, and only when it is actually running does the particular library get linked to it to specify which actual code (including the code with system calls specific to this platform) would be used. In the case of Schmidt, the developer makes the decision which library to use at development time, not run time. Schmidt then specifies a particular library with which to link (still at development time) and the resulting application is now hard-coded to work on only one particular system (and then, of course, the actual system calls eventually occur at run-time). Thus, in the claimed invention, the choice of which system implementation to use can be deferred to run-time, whereas in Schmidt's disclosed system it is determined prior to run-time and once determined can no longer be changed at run time".

Schmidt's repeated references to using "inline functions" is proof that his the technique is limited to compile-time. Schmidt also refers repeatedly to the benefit of his technique as allowing error checking at "compile-time" rather than "run-time". He also makes repeated references to "stronger type-checking", "type-safe operations", and "type mismatch detection" which are compile time features. Schmidt also talks about developers "designing and implementing" their own "wrappers", stating it "is an effective way for programmers to learn C++". Schmidt fails to disclose running or executing at run-time, as claimed by the Applicant.

The Applicant's claimed invention allows writing a program once, compiling it once, and simply by the differences in the platform on which it runs, use the appropriate wrapper. Schmidt does provide for different system implementations for the same programming API, but he requires that the choice be specified by the developer, rather than waiting until it is determined upon which system the program is actually running. Thus in claimed invention, the object oriented statements using a wrapper are located by the system at run-time while running or executing, whereas in Schmidt the locating is completed at development time. Schmidt fails to disclose or even suggest the Applicant's claimed invention.

By the above remarks, the Applicant believes all of the issues raise by the Examiner have been resolved. Accordingly, the Applicant respectfully requests the Examiner's reconsideration of the claims, allow the claims and pass the case to issue.

The Assistant Commissioner is hereby authorized to charge any additional fees which may be required for the timely consideration of this amendment under 37 C.F.R. §§ 1.16 , or credit any overpayment to Deposit Account No. 13-4503, Order No. 3048-7035.

Respectfully submitted,
MORGAN & FINNEGAN, L.L.P.

Dated: 6/28/2000

By: [illegible]
John E. Hoel
Registration No. 26,279
202-857-7887 - Telephone
202-857-7929 - Facsimile

SENDER'S ADDRESS:
Morgan & Finnegan L.L.P.
1775 Eye Street, N.W. Suite 400
Washington, D.C. 20006

C

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

[illegible]

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/140,523	08/26/98	ORTON D	P-046.63

LMC1/0802

JOHN E. HOEL
MORGAN & FINNEGAN, LLP
345 PARK AVENUE
NEW YORK NY 10154

EXAMINER	
CHAVIS, J	
ART UNIT	PAPER NUMBER
2762	17

DATE MAILED: 08/02/00

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

Office Action Summary	Application No. 09/140,523	Applicant(s) Orton et al.
	Examiner John Chavis	Group Art Unit 2762

☒ Responsive to communication(s) filed on *Jun 29, 2000*

☐ This action is **FINAL.**

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle* 1935 C.D. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire 3 month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claim

☒ Claim(s) *12, 16-20, and 22-30* is/are pending in the applicat

Of the above, claim(s) ______ is/are withdrawn from consideration

☐ Claim(s) ______ is/are allowed.

☒ Claim(s) *12, 16-20, and 22-30* is/are rejected.

☐ Claim(s) ______ is/are objected to

☐ Claims ______ are subject to restriction or election requirement.

Application Papers

☐ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The drawing(s) filed on ______ is/are objected to by the Examiner.

☐ The proposed drawing correction, filed on ______ is ☐ approved ☐ disapproved.

☐ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

☐ All ☐ Some* ☒ None of the **CERTIFIED** copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) ______.

☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: ______

☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

☒ Notice of References Cited, PTO-892

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s). ______

☐ Interview Summary, PTO-413

☐ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Notice of Informal Patent Application, PTO-152

— SEE OFFICE ACTION ON THE FOLLOWING PAGES —

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 11-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schmidt (C++ Report 1992). The applicant claims a system and method of enabling an object oriented program to access a procedural operating system using procedural function calls. The applicant further claims that his system checks that objects are loaded prior to invocation and if not loading the objects via a runtime loader. The features of the applicant's claims are now presented in a side by side manner with the teachings of Schmidt.

Claims	Schmidt
Clm 16. A computer system, comprising: computer hardware for performing native system services;	see page 50 and fig. 1, OS Kernel Services.
a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services;	See again fig. 1 as indicated above and the Existing C System Call API, Which indicates that native services Are provided for.

object oriented methods requiring native system services; and	See the user application of fig. 1.
procedural program logic code, responsive to invocations of the object oriented methods, for causing the procedural operating system to control the computer hardware to perform the required native system services.	this feature is not provided for; however, the system does enable C++ (object oriented methods) wrappers on an existing OS (native Procedural Operating system). Also, Schmidt indicates that it is possible to use C Wrappers (page 54, right column, 1st paragraph). He further Indicates a disadvantage of using C++ Wrappers-lack of support for exception handling; therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize C Wrappers as suggested by Schmidt to enable access to existing operating system functions by object oriented programs to enable a programmer use the new features of C++ without paying a heavy penalty for having to learn a new language immediately, to enable portability between the languages, and to not have to sacrifice the exception handling features of C.

The applicant indicates that Schmidt's system does not respond to invocations of the object-oriented methods at run-time and makes references to Schmidt's use of the term "inline functions" as proof that Schmidt's system is limited to compile-time. The applicant also mentions that Schmidt "refers repeatedly to the benefit of his technique as allowing error checking at "compile-time" rather than run-time", which is considered irrelevant since neither the applicant's. system or Schmidt's system is designed for error checking. The applicant's claims also do not discuss error checking and neither does the examiner's response to the claims.

In reference to Schmidt's system not responding to invocations of the object-oriented methods at run-time, the examiner considers Schmidt's system to support the features. The features are indicated via the details of the Portability and Extensibility subheadings on page 4. Under the portability section notice that "Application programs may then be written using a single object-oriiented API, which is mapped **transparently** onto the appropriate system calls that access the particular underlying OS mechanisms." Transparent mapping is inherently a **run-time** function (ie. while running or executing).

Furthermore, under the extensibility subheading, notice the reference to **dynamic binding** (also a run-time feature, ie. not a feature specified by the developer) to "help improve the extensibility of the existing OS interfaces". "The goal is to allow applications to extend the original API's **without** modifying the design or implementation of the existing wrapper infrastructure."

However, assuming that the applicant is correct in indicating that Schmidt does not teach or suggest the feature of loading information during runtime, the feature is taught by Janis et al. (5,247,681) to reduce the amount of memory required at runtime to improve memory managemant, see col. 3 lines 6-21.

executable program memory associated with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code;	see col. 3 lines 24-37.
means for making determinations during runtime execution it object-oriented methods to be invoked are present in the executable program memory; and	see again the cited protions directly Above.
a runtime loader, responsive to the determinations, to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods.	See col. 3 lines 64-col. 4 line 15.

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize the feature taught by Janis in Schmidt's system for the same reasons specified by Janis to reduce the amount of runtime loading of modules to imporve memory management by only loading modules that have not previously been loaded. The feature would have been obvious to a person of ordinary skill in the art at the time of the invention because it would all one to improve access times to items that are already loaded.

Therefore, for the reasons cited in the previous action and the responses above, the rejection of claims 12, 16-20 and 22-30 is consider proper in view of the teachings of Schmidt.

Clm 12. The computer system of claim 16, wherein the procedural program logic code further comprises: procedural program logic code portions specific to each object oriented method to issue one or more procedural function calls compatible with the native interfaces to control the native system services performed by the hardware environment to correspond to the native system services required by the object oriented method.	See page 54, the remote procedure calls, which indicates that That object oriented function are mapped transparently onto the appropriate system call that accesses the underlying OS mechanisms (native system services).

In reference to claim 25, see claim 16, above.

As per claim 26, see claim 12.

The features of claims 27-30 are taught via claim 12.

As per claims 17-20 and 22-24 (see the rejections above and also note that:), Schmidt does not indicate that memory is checked to determine if functions are already loaded; however, It would have been obvious to a person of ordinary skill in the art at the time of the invention to check to determine if data is loaded to keep from loading multiple copies into a primary storage space. The feature would have been obvious to provide efficient utilization of resources; since, all programs must be loaded into executable memory before they are executed (hence the name "executable memory").

3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Examiner Chavis whose telephone number is (703) 305-9665. The examiner can normally be reached on Monday-Friday from 8:00am to 4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tariq Hafiz (New Art

Unit 2762), can be reached on (703) 305-9643. The fax phone number for this Group is (703) 305-0040.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-3900.

[illegible]

JQC
July 31, 2000

[illegible]

[illegible]
Supervisory Patent Examiner
Technology Center 2700

9

Notice of References Cited

Application No.	09/140,523
Applicant(s)	Orton et al.
Examiner	Chavis, J. G.
Group Art Unit	2762
Page	1 of 1

U.S. PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
	A	5,247,681	9-21-93	Janis et al.	709	305
	B	5,339,430	8-16-94	Lundin et al.	709	305
	C	5,901,313	5-4-99	Wolf et al.	345	335
	D	5,555,418	9-10-96	Nilsson et al.	709	305
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
	N						
	O						
	P						
	Q						
	R						
	S						
	T						

NON-PATENT DOCUMENTS

*		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U		
	V		
	W		
	X		

* A copy of this reference is not being furnished with this Office action.
(See Manual of Patent Examining Procedure, Section 707.05(a).)

2/22
~~2762~~
#18
mlf
1/19/01

OIPE JC135
DEC 29 2000
PATENT & TRADEMARK OFFICE

Docket No. 3048-7035US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVE
JAN 17 200
Technology Center

Applicant(s): Orton et al.

Serial No.: 09/140,523

Filed: 08/26/98

Group Art Unit: 2762

Examiner: J. Chavis

For: OBJECT-ORIENTED OPERATING SYSTEM

PETITION AND FEE FOR EXTENSION OF TIME (37 C.F.R. §1.136(a))

Commissioner for Patents
Washington, D.C. 20231

Sir:

1. This is a petition for an extension of time for responding to the official action mailed 08/02/00.

2. The communication in connection with the matter for which this extension is requested

☒ is filed herewith.

☐ has been filed on _____.

3. ☐ Applicant(s) is/are entitled to Small Entity Status. ☐ Small Entity Statement is attached. ☐ Statement has already been filed.

4.

		Total Months Requested	Fee for Other than Small Entity	Fee for Small Entity
a.	☐	one month	$110.00	$55.00
b.	☒	two months	$390.00	$195.00
c.	☐	three months	$890.00	$445.00
d.	☐	four months	$1,390.00	$695.00
e.	☐	five months	$1,890.00	$945.00

f. ☐ An extension for months has already been secured for filing the above-identified communication and the fee paid therefor of $ _____ is deducted from the total fee due for the total months of extension now requested. The fee for this extension ($ _____), minus the fee previously paid ($) equals $ (total fee due).

5. ☐ A check in the amount of $ _____ to cover the extension fee is attached.

6. ☒ Charge fee to Deposit Account No. 13-4503. Order No. 3048-7035US1. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

01/03/2001 EGRETCH1 00000006 134503 09140523
01 FC:116 2035 390.00 CH

983FH192

7. [✓] The Commissioner is hereby authorized to charge any additional fees which may be required by this paper, or credit any overpayment to Deposit Account No. 13-4503. Order No. 3048-7035US1. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

Respectfully submitted,
MORGAN & FINNEGAN, L.L.P.

Dated: 12/28/00

By: [illegible]
John E. Hoel
Registration No. 26,279
(202)857-7887 Telephone
(202)857-7929 Facsimile

CORRESPONDENCE ADDRESS:
MORGAN & FINNEGAN, L.L.P.
345 Park Avenue
New York, NY 10154-0053

19/20
[illegible]
1/19/01

OIPE JC133
DEC 29 2000
PATENT & TRADEMARK OFFICE

Docket No. 3048-7035US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Orton et al.	OTLC Docket:	P-046.63
Serial No.:	09/140,523	Group Art Unit:	2762
Filed:	August 26, 1998	Examiner:	J. Chavis
For:	OBJECT-ORIENTED OPERATING SYSTEM		

RECEIVED
JAN 17 2001
Technology Center 2100

AMENDMENT

Honorable Assistant Commissioner
of Patents and Trademarks
Washington, D.C. 20231
Sir:

01/24/2001 [illegible] 00000001 09140523
01 FC:103 18.00 CH

In response to the Office Action of August 2, 2000, please amend the application as follows:

IN THE CLAIMS:

Please amend claim 16 as follows:

1 ~~16~~. (TWICE AMENDED) A computer system, comprising:

computer hardware for performing native system services;

a procedural operating system, having a native interface, for controlling the computer hardware to perform the native system services;

object oriented methods requiring native system services;

X1 Cont

procedural program logic code, responsive to invocations of the object-oriented methods <u>during runtime</u>, for causing the procedural operating system to control the computer hardware to perform the required native system services;

executable program memory associated with the computer hardware for runtime execution of the procedural operating system, invocations of the object-oriented methods and related portions of the procedural program logic code;

01/04/2001 EGRETCH1 00000069 134503 09140523

01 FC:102 560.00 CH
02 FC:103 18.00 CH

20302_1

[illegible] 09140523 65

D

983FH194

A1 cont'd

means for making determinations during runtime execution if object-oriented methods to be invoked are present in the executable program memory; and

a runtime loader, responsive to the determinations, to selectively load required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods.

Please amend ~~claim~~ 22 as follows:

7 ~~22~~. (TWICE AMENDED) A method for operating a computer system, comprising the steps of:

A2

executing a procedural operating system on computer hardware, the procedural operating system including a native interface, responsive to procedural function calls, for providing native system services;

issuing calls during runtime, compatible with the native interface, to provide the native system services in response to invocations of object-oriented methods requiring such native system services;

determining during runtime if object-oriented methods to be invoked during runtime execution are present in executable program memory associated with the computer hardware; and

selectively loading the object-oriented methods into the executable program memory during runtime before invocation thereof, if not yet loaded.

Please amend ~~claim~~ 27 as follows:

12 ~~27~~. (TWICE AMENDED) A method for operating a computer system, comprising the steps of:

executing a procedural operating system, based on Windows or Unix operating systems, on a Unix or IBM compatible computer hardware environment;

A3 cont

providing an object-oriented interface, executing on the computer hardware environment, and responsive to object-oriented programming, for instantiating objects from object-oriented

66

classes, encapsulating data for exclusive use with each object, and invoking object-oriented methods in the objects for operating on the encapsulated data;

A3 Cont'd

providing procedural programming logic code, responsive during runtime to selected ones of said invoked object-oriented methods requiring native system services, for issuing procedural calls, compatible with a native interface of the procedural operating system, to cause the hardware environment to provide the native system services in response to the object-oriented methods;

and

loading the methods during runtime before invocation thereof;

whereby a choice of which system implementation to use can be deferred to run-time.

Please add new claim ~~31~~ as follows:

16 ~~31~~. (NEW) A method for operating a computer system including a memory, comprising the steps of:

storing in the memory a library of procedural program logic code;

said library including first procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform first type native system services;

A4 Cont

said library including second procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform second type native system services different from said first type;

executing a procedural operating system in the memory, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;

running an object-oriented program in a task address space of the memory, the program including an object-oriented method requiring the second type native system services;

determining during runtime whether said second type procedural program logic code is available in said task address space; and

loading said second type procedural program logic code into said task address space during runtime.

67

983FH196

[Please add new claim 32 as follows:]

17
~~32~~. (NEW) A method for operating a computer system including an executable program memory, comprising the steps of:

storing in the computer system a library of procedural program logic code;

said library including first procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform first type native system services;

said library including second procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform second type native system services different from said first type;

executing a procedural operating system in the executable program memory, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;

B4 Cont

running an object-oriented program in the executable program memory, the program including an object-oriented method requiring the second type native system services;

determining during runtime whether said second type procedural program logic code is available in the executable program memory; and

loading said second type procedural program logic code into the executable program memory during runtime.

[Please add new claim 33 as follows:]

18
~~33~~. (NEW) A method for operating a computer system including an executable program memory, comprising the steps of:

storing in the computer system a library of procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform native system services;

executing a procedural operating system in the executable program memory, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;

running an object-oriented program in the executable program memory, the program including an object-oriented method requiring native system services;

determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services; and

loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services.

Please add new claim 34 as follows:

19
~~34~~. (NEW) A method for operating a computer system including an executable program memory, comprising the steps of:

D4 Cont

storing in the computer system a library of procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform native system services;

executing a procedural operating system in the computer system, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;

running an object-oriented program in the executable program memory, the program including an object-oriented method requiring native system services;

determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services; and

loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services.

Please add new claim 35 as follows:

20
~~35~~. (NEW) A computer system including an executable program memory, comprising:

a library of procedural program logic code in the computer system which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform native system services;

a procedural operating system in the computer system, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;

an object-oriented program in the executable program memory, the program including an object-oriented method requiring native system services;

a processor in the computer system for determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services; and

said processor loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services.

[Please add new claim 36 as follows:]

21
~~36~~. (NEW) A method for operating a computer system including an executable program memory, comprising the steps of:

S4 Cont

storing in the computer system a library of procedural program logic code which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform native system services;

executing a procedural operating system in the computer system, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;

running an object-oriented program in the executable program memory, the program including an object-oriented method requiring native system services;

determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services;

loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services;

invoking said object-oriented method of said object-oriented program during runtime; and

responding with said loaded procedural program logic code to said invoking step to cause said procedural operating system to control the computer system to perform said required native system services.

[Please add new claim 37 as follows:]

22
~~37~~. (NEW) A computer system including an executable program memory, comprising:

a library of procedural program logic code in the computer system which is responsive to invocations of object-oriented methods, for causing a procedural operating system to control the computer system to perform native system services;

a procedural operating system in the computer system, the procedural operating system including a native interface responsive to procedural function calls, for providing native system services;

an object-oriented program in the executable program memory, the program including an object-oriented method requiring native system services;

a processor in the computer system for determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services;

X4
Concl

said processor loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services;

said processor invoking said object-oriented method of said object-oriented program during runtime; and

said loaded procedural program logic code responding to said invoking to cause said procedural operating system to control the computer system to perform said required native system services.

REMARKS

STATUS OF THE CLAIMS

Claims 12, 16-20, 22-30, as amended, remain in the case remain in the case and new claims 31 to 37 are added to the case.

REJECTION OF THE CLAIMS OVER THE PRIOR ART

Claims 12, 16-20, 22-30 were rejected under 35 U. S. C § 103 as being unpatentable over Schmidt, "Systems Programming With C++ Wrappers", C++ Report, October 1992.

Claims 12, 16-20, 22-30 were further rejected over a combination of the Schmidt reference and Janis et al. US patent 5,247,681. The Examiner's rejection reads as follows:

> Therefore, it would be obvious to a person of ordinary skill in the art at the time of the invention to utilize the feature taught by Janis (US patent 5,247,681) in Schmidt's system for the same reasons specified by Janis to reduce the amount of runtime loading of modules to improve memory management by only loading modules that have not previously been loaded. The feature would have been

71

obvious to a person of ordinary skill in the art at the time of the invention because it would it would all one to improve access times to items that are already loaded.

APPLICANT'S RESPONSE

The following remarks are in response to the Examiner's Office action of August 2, 2000.

[1] The Applicant's prior remarks in the Preliminary Amendment of June 28, 2000, concerning the Schmidt reference are incorporated herein by reference.

[2] The Office action of August 2, 2000 provides a table on pages 1 and 2, applying the Applicant's claim 16 to the Schmidt reference. In particular, the Office action quotes the Applicant's "procedural program logic code" claim limitation, which reads as follows:

> procedural program logic code, responsive to invocations of the object-oriented methods, for causing the procedural operating system to control the computer hardware to perform the required native system services;

The table in the Office action, concerning the Applicant's " procedural program logic code" claim limitation, reads as follows:

> this feature is not provided for; however, the system does enable C++ (object oriented methods) wrappers on an existing OS (native Procedural Operating system). Also, Schmidt indicates that it is possible to use C Wrappers (page 54, right column, 1st paragraph). He further Indicates a disadvantage of using C++ Wrappers-lack of support for exception handling; therefore, it would have been obvious to a person of ordinary skill in the art at the time of the invention to utilize C Wrappers as suggested by Schmidt to enable access to existing operating system functions by object oriented programs to enable a programmer use the new features of C++ without paying a heavy penalty for having to learn a new language immediately, to enable portability between the languages, and to not have to sacrifice the exception handling features of C.

In response, the Applicant points out that the "C Wrappers" which the Examiner is equating to the Applicant's claimed "native system services", is not in fact a native system service. A "C wrapper", as used in the Schmidt reference, and as is commonly acknowledged in the art, is an interface between programming constructs provided by the C programming language. A "C wrapper" is not a native system service of an operating system. Native system services are performed by computer hardware in response to calls to an operating system, such as the service of accessing data stored on a disk drive or the service of allocating storage areas in a memory.

[3] In the Office action of August 2, 2000, the second last paragraph on page 2 reads as follows:

> The applicant indicates that Schmidt's system does not respond to invocations of the object-oriented methods at run-time and makes references to Schmidt's use of the term "inline functions" as proof that Schmidt's system is limited to compile-time. The applicant also mentions that Schmidt "refers repeatedly to the benefit of his technique as allowing error checking at "compile-time" rather than run-time", which is considered irrelevant since neither the applicant's system or Schmidt's system is designed for error checking. The applicant's claims also do not discuss error checking and neither does the examiner's response to the claims.

In response, the Applicant must point out that the issue is whether the Schmidt reference discloses or suggests the Applicant's claimed steps or means for performing the steps of:

> determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services; and
> loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services in response to the determining step.

Still further, an additional issue is whether the Schmidt reference discloses or suggests the Applicant's additionally claimed steps or means for performing the steps of:

> invoking said object-oriented method of said object-oriented program during runtime following the loading step; and
> responding by said loaded procedural program logic code to said invoking to cause said procedural operating system to control the computer system to perform said required native system services.

The declarations by the Schmidt reference concerning error checking are pointed to by the Applicant as evidence that the Schmidt reference is limited to compile-time operations. The Schmidt reference fails to disclose or suggest the runtime steps of determining and loading claimed by the Applicant. Furthermore, the Schmidt reference fails to disclose or suggest the runtime steps of invoking and responding claimed by the Applicant. Further, the Schmidt reference fails to disclose or suggest the system operating during runtime to perform those steps claimed by the Applicant.

[4] In the Office action of August 2, 2000, the last paragraph on page 2 reads as follows (emphasis in original):

> In reference to Schmidt's system not responding to invocations of the object-oriented methods at run-time, the examiner considers Schmidt's system to support the features. The features are indicated via the details of the Portability and Extensibility subheadings on page 4. Under the portability section notice that "Application programs may then be written using a single object-oriented API, which is mapped **transparently** onto the appropriate system calls that access the particular underlying OS mechanisms." Transparent mapping is inherently a **run-time** function (i.e. while running or executing).

In response, the Applicant must point out that the word "transparently" quoted in the Office action is not an enabling disclosure of the Applicant's claimed invention. In order to be effective as prior art, the reference must provide a disclosure sufficient to enable a person skilled in the art to make and use the claimed invention. The characterization of a "transparent mapping" for an "object oriented API" is obscure. The Examiner is using the Applicant's own description of the claimed invention as a road map to interpret the ambiguous disclosure in the Schmidt reference. The Schmidt reference fails to disclose or suggest the Applicant's claimed steps or means for performing the steps of:

> determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services; and
> loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services in response to the determining step.

Still further, the Schmidt reference fails to disclose or suggest the Applicant's additionally claimed steps or means for performing the steps of:

> invoking said object-oriented method of said object-oriented program during runtime following the loading step; and
> responding by said loaded procedural program logic code to said invoking to cause said procedural operating system to control the computer system to perform said required native system services.

To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson, 424 F.2d 1382, 165 USPQ 494, 496 (CCPA 1970).

[5] In the Office action of August 2, 2000, the first full paragraph on page 3 reads as follows (emphasis in original):

> Furthermore, under the extensibility subheading, notice the reference to **dynamic binding** (also a run-time feature, i.e. not a feature specified by the developer) to "help improve the extensibility of the existing OS interfaces". "The goal is to allow applications to extend the original API's **without** modifying the design or implementation of the existing wrapper infrastructure."

In response, the Applicant must point out that the phrase "dynamic binding" quoted in the Office action is not an enabling disclosure of the Applicant's claimed invention. In order to be effective as prior art, the reference must provide a disclosure sufficient to enable a person skilled in the art to make and use the claimed invention. The characterization of "dynamic binding" for an "original API without modifying the design" is obscure. The Examiner is using the Applicant's own description of the claimed invention as a road map to interpret the ambiguous disclosure in the Schmidt reference. The Schmidt reference fails to disclose or suggest the Applicant's claimed steps or means for performing the steps of:

> determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services; and
> loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services in response to the determining step.

Still further, the Schmidt reference fails to disclose or suggest the Applicant's additionally claimed steps or means for performing the steps of:

> invoking said object-oriented method of said object-oriented program during runtime following the loading step; and
> responding by said loaded procedural program logic code to said invoking to cause said procedural operating system to control the computer system to perform said required native system services.

To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. In re Royka, op. cit.; "All words in a claim must be considered in judging the patentability of that claim against the prior art." In re Wilson, op. cit.

[6] In the Office action of August 2, 2000, on page 3, the Office action applies the Janis patent. The Examiner's comments read as follows:

> However, assuming that the applicant is correct in indicating that Schmidt does not teach or suggest the feature of loading information during runtime, the feature is taught by Janis et al. (5,247,681) to reduce the

amount of memory required at runtime to improve memory management, see cot. 3 lines 6-21.

The Applicant responds that the Janis reference does not disclose what the Examiner is suggesting. In applying the Janis patent, the table on page 3 of the Office action reads the "executable program memory" limitation of the Applicant's claim 16 onto the cited Summary of the Invention section at Column 3, lines 24-37 of the Janis patent, which reads as follows:

> The present invention overcomes the deficiencies of the previous techniques by providing a system and method for sharing previously loaded software modules which are part of a computer program without having to place them in a common area of main memory of a computer system. More specifically, the present invention keeps track of the location of any software modules which remain loaded in a private area of main memory, having been loaded by a previous execution of the computer program. In this way, a subsequent execution of that computer program requiring those software modules can immediately access them rather than having to re-load them into memory.

In response, the Applicant asserts that the cited section at Column 3, lines 24-37 of the Janis reference teaches away from the Applicant's claimed invention. Janis is describing sharing previously loaded software modules. Janis says it twice in the same quotation used by the Examiner. The Applicant is claiming a runtime loader that selectively loads the required object-oriented methods into the executable program memory during runtime before invocation of the object-oriented methods. The Janis reference teaches away from the Applicant's claimed invention. A prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention. W.L. Gore & Associates, Inc. v. Garlock, Inc., 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983), cert. denied, 469 U.S. 851 (1984).

Combining the Janis reference with the Schmidt reference would change the principle of operation alleged by the Examiner to be present in the Schmidt reference. Combining the Janis reference with the Schmidt reference would render the operation alleged by the Examiner to be present in the Schmidt reference to become unsatisfactory for its intended purpose. If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious. In re Ratti, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). If the proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. In re Gordon, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984)

As pointed out by Judge Rich in In re Vaeck, 947 F.2d 488, 20 U.S.P.Q.2d 1438, 1442 (Fed. Cir. 1991), the suggestion for the combination and the likelihood of success can not come from the applicant's disclosure. Specifically Judge Rich wrote:

> Where claimed subject matter has been rejected as obvious in view of a combination of prior art references, a proper analysis under §103 requires, inter alia, consideration of two factors: (1) whether the prior art would have suggested to those of ordinary skill in the art that they should make the claimed composition or device, or carry out the claimed process; and (2) whether the prior art would also have revealed that in so making or carrying out, those of ordinary skill would have a reasonable expectation of success. See In re Dow Chemical Co., 837 F.2d 469, 473, 5 USPQ2d 1529, 1531 (Fed. Cir. 1988). Both the suggestion and the reasonable expectation of success must be founded in the prior art, not in the applicant's disclosure.

The Federal Circuit has stated this principle in still other cases. For example, "In proceedings before the Patent and Trademark office, the Examiner bears the burden of establishing a prima facie case of obviousness based upon the prior art. The Examiner can satisfy this burden only by showing some objective teaching in the prior art or that knowledge generally available to one of ordinary skill in the art would lead that individual to combine the relevant teachings of the references...." In re John R. Fritch, 972 F.2d 1260, 1265 (Fed. Cir. 1992). However, the suggestion "must be other than the knowledge learned from the applicant's disclosure." In re the Dow chemical Co. 837 F.2d 469 (Fed. Cir. 1988) Furthermore, "[I]t is impermissible to use the claimed invention as an instruction manual or 'template' to piece together the teachings of the prior art so that the claimed invention is rendered obvious." In re John R. Fritch, 972 F.2d at 1266 (Fed. Cir. 1992); In re Gordon, 733 F.2d at 902, 221 USPQ at 1127.

There is no motivation to combine the Schmidt reference with the Janis reference, since the resulting combination is not the same as the Applicant's claimed invention and would not accomplish the same result as that of the Applicant's claimed invention. The Janis reference teaches away from the Applicant's claimed invention, so why would anyone want to combine it with Schmidt, as the Examiner suggests.

Furthermore, the combination of the Schmidt reference with the Janis reference fails to disclose or suggest the Applicant's claimed steps or means for performing the steps of:

> determining during runtime whether procedural program logic code is available in the executable program memory to provide said required native system services; and
> loading procedural program logic code from said library into the executable program memory during runtime to provide said required native system services in response to the determining step.

Still further, the combination of the Schmidt reference with the Janis reference fails to disclose or suggest the Applicant's additionally claimed steps or means for performing the steps of:

invoking said object-oriented method of said object-oriented program during runtime following the loading step; and
responding by said loaded procedural program logic code to said invoking to cause said procedural operating system to control the computer system to perform said required native system services.

The Applicant's claims are allowable over either the Schmidt reference or the Janis reference, or the combination of the Schmidt reference with the Janis reference.

By the above remarks, the Applicant believes all of the issues raise by the Examiner have been resolved. Accordingly, the Applicant respectfully requests the Examiner's reconsideration of the claims, allow the claims and pass the case to issue.

The Assistant Commissioner is hereby authorized to charge any additional fees which may be required for the timely consideration of this amendment under 37 C.F.R. §§ 1.16 , or credit any overpayment to Deposit Account No. 13-4503, Order No. 3048-7035.

Respectfully submitted,
MORGAN & FINNEGAN, L.L.P.

Dated: 12/28/00

By: [illegible]
John E. Hoel
Registration No. 26,279
202-857-7887 - Telephone
202-857-7929 - Facsimile

SENDER'S ADDRESS:
Morgan & Finnegan L.L.P.
1775 Eye Street, N.W. Suite 400
Washington, D.C. 20006

20302_1

OIPE JC139 DEC 2 9 2000 PATENT & TRADEMARK OFFICE

Docket No. 3048-7035US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Orton et al.

Group Art Unit: 2762

Serial No.: 09/140,523

Examiner: J. Chavis

Filed: August 26, 1998

For: OBJECT-ORIENTED OPERATING SYSTEM

RECEIVED
JAN 1 7 2001
Technology Center 2100

AMENDMENT FEE TRANSMITTAL

COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Sir:

Transmitted herewith is an Amendment for the above-identified application.

☐ No additional fee is required.

☒ The additional fee has been calculated as shown below:

CLAIMS AS AMENDED

	Claims Remaining After Amendment	Highest No. Covered by Previous Payments	Extra	Rate	Additional Fee
Total Claims*	21 -	20	1	$18.00/ $9.00	$ 18.00
Independent Claims	10 -	3	7	$80.00/ $40.00	$ 560.00
Multiple Dependent Claims	(If claims added by amendment include Multiple Dependent Claim(s) and there was no Multiple Dependent Claim(s) in application before amendment add $270.00 to additional fee ($135 for small entity).				$ 0.00
				TOTAL	$ 578.00

*Includes all independent and single dependent claims and all claims referred to in multiple dependent claims. See 37 C.F.R. §1.75(c).

☐ Statement of "Small Entity" Status Under 37 C.F.R. §1.27 filed ______, Reduced Fees Under 37 C.F.R. §1.9(f) paid herewith. $

20351_1

983FH208

WI-Apple0000651

☒ Charge fee to Deposit Account No. 13-4503, Order No. 3048-7035US1. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

☒ The Commissioner is hereby authorized to charge any additional fees which may be required for this amendment, including all fees pursuant to 37 C.F.R. §1.17 for its timely consideration, or credit any overpayment to Deposit Account No. 13-4503, Order No. 3048-7035US1. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

☐ _____ Pages Sequence Listing

☐ _____ Computer disk(s) containing substitute Sequence Listing

☐ Statement under 37 C.F.R. §1.825(b) that the computer and paper copies of the substitute Sequence Listing are the same.

☐ A check in the amount of $ _____ to cover the filing fee is attached.

Respectfully submitted,
MORGAN & FINNEGAN, L.L.P.

Dated: 12/28/00

By: [illegible]
John E. Hoel
Registration No. 26,279
(202) 857-7887 Telephone
(202) 857-7929 Facsimile

CORRESPONDENCE ADDRESS:

Morgan & Finnegan L.L.P.
345 Park Avenue
New York, New York 10154

983FH209