

# Exhibit 14

332486

Class 700  
Subclass  
ISSUE CLASSIFICATION



5481721

UTILITY SERIAL NUMBER 08/332486	DATE OF FILING JAN 02 1995	PATENT NUMBER
SERIAL NUMBER 08/332,486	FILING DATE 10/31/94	CLASS 395
	SUBCLASS 700	GROUP ART. CL. 2316
		EXAMINER KRIESS PAYNE

APPLICANTS

BERTRAND SERLET, PALO ALTO, CA; LEE BOYNTON, PALO ALTO, CA; AVADIS TEVANIAN, MOUNTAIN VIEW, CA.

\*\*CONTINUING DATA\*\*\*\*\*  
VERIFIED THIS APPLN IS A CON OF 07/731,636 07/17/91, now abandoned.  
MAP

\*\*FOREIGN/PCT APPLICATIONS\*\*\*\*\*  
VERIFIED  
MAP (none)

FOREIGN FILING LICENSE GRANTED 12/01/94

Foreign priority claimed 35 USC 119 conditions met	<input type="checkbox"/> yes <input checked="" type="checkbox"/> no	AS FILED	STATE OR COUNTRY CA	SHEETS DRWGS. 9	TOTAL CLAIMS 18	INDEP. CLAIMS 2	FILING FEE RECEIVED \$730.00	ATTORNEY'S DOCKET NO. 10010.929C
Verified and Acknowledged		Examiner's Initials MAP						

ADDRESS  
J.D. HARRIMAN II  
SUITE 1600  
2029 CENTURY PARK EAST  
LOS ANGELES, CA 90067

ISSUE FILE IN FILE

TITLE  
METHOD FOR PROVIDING AUTOMATIC AND DYNAMIC TRANSLATION OF OBJECT ORIENTED LANGUAGE-BASED MESSAGE PASSING INTO OPERATING SYSTEM MESSAGE PASSING USING PROGRAMMING  
PROXY OBJECTS  
U.S. DEPT. of COMM.-Pat. & TM Office-PTO-436L (rev. 10-78)

PARTS OF APPLICATION FILED SEPARATELY		6-27-95 April Y. Cherny Applications Examiner	
NOTICE OF ALLOWANCE MAILED 5-26-95		CLAIMS ALLOWED Total Claims: 24 Print Claim: 1	
ISSUE FEE Amount Due: \$1210.00 Date Paid: 8-25-95		DRAWING Sheets Drwg.: 10 Figs. Drwg.: 11 Print Fig.: 30	
Label Area		ISSUE BATCH NUMBER M-91	
PREPARED FOR ISSUE			
<p>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 &amp; Trademark Office is restricted to authorized employees and contractors only.</p>			

Form PTO-436A (Rev. 6/82)

(FACE)

721FH001

WI-Apple0001505

```

        syslog(LOG_ERR, "Remote Object 0x%x received free", self);
        return nil;
    }

    static unsigned hashProxy(const void *info, const void *data) {
        return ((RemoteObject *)data)->name;
    }

    static int isEqualProxy(const void *info, const void *data1, const void *data2) {
        return ((RemoteObject *)data1)->name == ((RemoteObject *)data2)->name;
    }

    static void freeProxy(const void *info, const void *data) {
        ((id)data reallyFree);
    }

    static NXHashTablePrototype proxyProto = {hashProxy, isEqualProxy, freeProxy, 0};

    - (unsigned)remoteObjectName { return name; }

    - remoteMethodInfo:(SEL)sel {
        id      res;
        id      args[4];
        if (sel == @selector(remoteMethodInfo)) return knownRemoteMethodInfo; /* to avoid infinite recursion */
        res = [knownSelectors valueForKey:(void *)sel];
        if (res) return res;
        ///? -> SN How to fill args cleanly
        bzero(args, sizeof(id)*4);
        Method method = class_getInstanceMethod((Class)[Object class], @selector(remoteMethodInfo));
        char      *type;
        int      offset2;
        int      offset0;
        SEL      *ref;
        method_getArgumentInfo(method, 0, &type, &offset0);
        method_getArgumentInfo(method, 2, &type, &offset2);
        ref = (SEL *) (((char *)args)+offset2-offset0);
        *ref = sel;
    }
    res = [(id) self forward:@selector(remoteMethodInfo) :args];
    if (! knownSelectors) knownSelectors = [HashTable newKeyDesc:@""];
    [knownSelectors insertKey:(void *)sel value:res];
    return res;
}

- encodeRemotelyFor:(Communication *)communication freeAfterEncoding:(BOOL *)flag {
    return self;
}

- writePortStream:(NXPortStream *) stream {
    [super writePortStream:stream];
    NXWritePortTypes (stream, "ii", &conn, &name);
    return self;
}

- readPortStream:(NXPortStream *) stream {
    [super readPortStream:stream];
    NXReadPortTypes (stream, "ii", &conn, &name);
    return self;
}

- afterPortReading:(Communication *)communication {

```



UNITED STATES DEPARTMENT OF COMMERCE  
Patent and Trademark Office

Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
---------------	-------------	----------------------	---------------------

07/731,636 07/17/91 SERLET

B 10010.929

EXAMINER

B3M1/0910

HECKER & HARRIMAN  
2049 CENTURY PARK EAST, STE. 1200  
LOS ANGELES, CA 90067

ART UNIT PAPER NUMBER

2316

2

DATE MAILED:

09/10/93

This is a communication from the examiner in charge of your application.  
COMMISSIONER OF PATENTS AND TRADEMARKS

This application has been examined  Responsive to communication filed on 7/17/91  This action is made final.

A shortened statutory period for response to this action is set to expire 3 month(s), 0 days from the date of this letter.  
Failure to respond within the period for response will cause the application to become abandoned. 35 U.S.C. 133

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

- |   |  |
|---|--|
| 1. <input checked="" type="checkbox"/> Notice of References Cited by Examiner, PTO-892. | 2. <input checked="" type="checkbox"/> Notice re Patent Drawing, PTO-948.        |
| 3. <input type="checkbox"/> Notice of Art Cited by Applicant, PTO-1449.                 | 4. <input type="checkbox"/> Notice of Informal Patent Application, Form PTO-152. |
| 5. <input type="checkbox"/> Information on How to Effect Drawing Changes, PTO-1474.     | 6. <input type="checkbox"/>  |

Part II SUMMARY OF ACTION

1.  Claims 1-18 are pending in the application.

Of the above, claims \_\_\_\_\_ are withdrawn from consideration.

2.  Claims \_\_\_\_\_ have been cancelled.

3.  Claims \_\_\_\_\_ are allowed.

4.  Claims 1-18 are rejected.

5.  Claims \_\_\_\_\_ are objected to.

6.  Claims \_\_\_\_\_ are subject to restriction or election requirement.

7.  This application has been filed with Informal drawings under 37 C.F.R. 1.85 which are acceptable for examination purposes.

8.  Formal drawings are required in response to this Office action.

9.  The corrected or substitute drawings have been received on \_\_\_\_\_. Under 37 C.F.R. 1.84 these drawings are  acceptable,  not acceptable (see explanation or Notice re Patent Drawing, PTO-948).

10.  The proposed additional or substitute sheet(s) of drawings, filed on \_\_\_\_\_ has (have) been  approved by the examiner.  disapproved by the examiner (see explanation).

11.  The proposed drawing correction, filed on \_\_\_\_\_, has been  approved,  disapproved (see explanation).

12.  Acknowledgment is made of the claim for priority under U.S.C. 119. The certified copy has  been received  not been received  been filed in parent application, serial no. \_\_\_\_\_; filed on \_\_\_\_\_.

13.  Since this application appears to be 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.

14.  Other

EXAMINER'S ACTION

PTOL-326 (Rev. 9-89)

721FH136

WI-Apple0001640

Serial No. 07/731636

-2-

Art Unit 2316

1. The disclosure is objected to because of Informal Drawings-  
OK see PTO-948.

2. The title of the invention is not descriptive and imprecise.  
A new title is required which should include more claimed  
features which differentiate the claimed invention from the Prior  
Art. The main idea or the improvement of the present invention  
should be reflected within twenty words. Specifically, the title  
OK should stress the use of 'proxy' objects in addition to the  
translation of a 'language-based' message to an 'operating  
system-based' message. In general, the title should cite a  
method of message passing.

3. The disclosure is objected to because of the following  
informalities:

7 → X (a) In figure 1, blocks 101, 102, 103 & 104 are not labelled.

OK (b) Figures 6, 7 & 8b are not labelled as prior art.

Appropriate correction is required.

4. The following is a quotation of the first paragraph of 35

U.S.C. § 112:

The specification shall contain a written description of the  
invention, and of the manner and process of making and using  
it, in such full, clear, concise, and exact terms as to  
enable any person skilled in the art to which it pertains,  
or with which it is most nearly connected, to make and use  
the same and shall set forth the best mode contemplated by  
the inventor of carrying out his invention.

721FH137

WI-Apple0001641

Serial No. 07/731636

-3-

Art Unit 2316

The specification is objected to under 35 U.S.C. § 112, first paragraph, as failing to adequately teach how to make and/or use the invention, i.e. failing to provide an enabling disclosure. It is unclear how a receiver object determines whether it has been given all the information it needs to execute a message, in order to determine if a query must be generated and sent back to the sender object. On p.22, paragraph 4, the specification cites that the present invention supports this feature, but does not fully disclose it. This rejection goes hand in hand with section (c) of the following 35 U.S.C. § 112, second paragraph rejection. This feature must be fully disclosed.

5. Claims 5-7, 11-18 are rejected under 35 U.S.C. § 112, first paragraph, for the reasons set forth in the objection to the specification.

*stands*

6. Claims 1-18 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The following claim language is unclear, vague and/or indefinite:

(a) as per claims 1,2,5,6 & 11, it is unclear who or what is executing these steps. If they are executed by a computer, this must be explicitly stated within the context of the claims,

721FH138

WI-Apple0001642

Serial No. 07/731636

-4-

Art Unit 2316

and the steps involving 'providing' must be clarified in relation to a computer actually implementing these steps. A computer 'providing' something is not a common art term. It appears to the examiner that these steps do not actually contribute to the claimed method of message sending.

(b) as per claims 2-10, they are rejected because they depend from previously rejected claim 1. In addition, claims 8 & 9 add irrelevant limitations to the claims from which they depend. Of what relevance is the use of C and Mach to the method of claim 1?

(c) as per claims 5 & 6, it is unclear how the steps of claim 5 fit into the steps of claim 2, and how the steps of claim 6 fit into the steps of claim 5. Are the steps of these claims executed directly after those of the claim from which they depend? Are they executed every time a message is sent? These seem to be steps which only occur under certain conditions, i.e. the second object needs further information from the first object. This claim must be clarified in conjunction with the above 35 U.S.C. § 112, first paragraph rejection.

(d) as per claims 12-18, they are rejected because they depend from previously rejected claim 11. In addition, claims 15 & 16 add irrelevant limitations to the claims from which they depend. Of what relevance is the use of C and Mach to the method of claim 11?

721FH139

WI-Apple0001643

Art Unit 2316

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

8. Claims 1-4, 8-10 are rejected under 35 U.S.C. § 102(a), (b) &

(e) as being anticipated by McCullough, "Transparent Forwarding: First Steps", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 331-341, 12/1987, and Bennet, "The Design and Implementation of Distributed Smalltalk", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 318-330, 12/1987. McCullough and Bennet both teach all that is claimed in these claims.

As stated in the 'BACKGROUND ART' section of the specification, p. 10, paragraph 3, and in the reference, McCullough clearly teaches the use of a 'ProxyObject' and



Serial No. 07/731636

-6-

Art Unit 2316

translation (encoding/decoding) of the message before and after transmission. This is taught in both directions, i.e. sending the message and the result. The translation from a language-based message into an operating system-based message is shown in the use of the `doesNotUnderstand:` primitive, the creation of an Ethernet packet, and the linearization of the arguments to the message. These features clearly indicate a system-dependent form of the message which is transmitted to the receiver object. This system-dependent form is then decoded, the message extracted, and the message executed by the receiver object. The result is then transmitted back to the sender in the same manner.

Bennet teaches a similar method of translation. His method includes the use of the `doesNotUnderstand:`, `perform:`, and `remoteSend:` primitives, a 'RemoteObjectTable' which uses a 'messageProcess' to construct a messageArray, and the encoding of an 'argument string'. These features also indicate a system-dependent form which is used during the transmission of the message.

Claims 3 & 4 do not contribute to making the claimed invention distinct from the prior art. The limitation, in claim 3, that the message comprises a method and an argument, is disclosed by Applicant in the 'BACKGROUND ART' section of the specification, p.1, lines 21-25, and is common in the art. It is also understood in the art that executing a message comprises executing the given method on the given argument, as described in

721FH141

WI-Apple0001645

Serial No. 07/731636

-7-

Art Unit 2316

claim 4.

Claims 8-10 do not make the claimed invention distinct from the prior art. Regarding claims 8 & 9, see the above 35 U.S.C. § 112, second paragraph rejection. Regarding claim 10, this limitation is understood in the general understanding of the 'proxy' concept.

9. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Quan, U.S. Pat. No. 5,230,051 discloses a distributed messaging system.

Simor, U.S. Pat. No. 5,060,150 discloses a message-based operating system.

Shapiro, "The Design of a Distributed Object-Oriented Operating System for Office Applications", ESPRIT '88. Putting the Technology to Use. Proceedings of the 5th Annual ESPRIT Conference, pp. 1020-7, vol. 2, Nov. 1988.

10. A shortened statutory period for response to this action is set to expire 3 (three) months, 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will cause the application to become ABANDONED according to 35 U.S.C. 133, M.P.E.P. 710.02, 710.02(b).

11. Any inquiry concerning this communication or earlier

721FH142

WI-Apple0001646


Serial No. 07/731636


-8-

Art Unit 2316

communications from the examiner should be directed to Matthew Payne whose telephone number is (703) 305-9593.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

  
KEVIN A. KRIESS  
PRIMARY EXAMINER  
GROUP 2300

  
September 7, 1993

721FH143

WI-Apple0001647



3/a  
William  
10010.9293-25-94

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of: )  
BERTRAND SERLET )  
Serial No.: 07/731,636 )  
Filed: JULY 17, 1991 )  
For: METHOD FOR PROVIDING )  
AUTOMATIC AND )  
DYNAMIC TRANSLATION )  
OF OBJECT ORIENTED )  
PROGRAMMING LANGUAGE )  
BASED MESSAGE PASSING )  
INTO OPERATING SYSTEM )  
MESSAGE PASSING USING )  
PROXY OBJECTS )

Group Art Unit 2306  
Examiner M. PAYNE  
**RECEIVED**  
MAR 24 1994  
**GROUP 2300**

*Handwritten:* 3/29/94

AMENDMENT AND RESPONSE

Hon. Commissioner of  
Patents and Trademarks  
Washington, D. C. 20231

Sir:

This is in response to the outstanding office action, paper number 2, in the present application. Please make the following changes in the present application:

IN THE TITLE

Please amend the title as follows:

Delete "METHOD FOR PROVIDING AUTOMATIC AND DYNAMIC TRANSLATION OF LANGUAGE-BASED MESSAGE PASSING INTO OPERATION SYSTEM MESSAGE PASSING" and insert in place thereof: --

METHOD FOR PROVIDING AUTOMATIC AND DYNAMIC TRANSLATION OF OBJECT ORIENTED PROGRAMMING LANGUAGE-BASED MESSAGE

*Handwritten:* aa

*Handwritten:* AA

aa)

PASSING INTO OPERATION SYSTEM MESSAGE PASSING USING PROXY OBJECTS.

IN THE DRAWINGS

Applicant submits herewith Figures 1, 6, 7, and 8B with corrections indicated in red. Applicant will submit formal drawings upon receipt of a notice of allowance.

IN THE CLAIMS

sub E1

1. (Once amended) A method for sending [a] an object oriented programming language based message having dynamic binding from a first object in a first process to a second object in a second process, said method comprising the steps of:

a1

providing, using a first processing means, said object oriented programming language based message to a first proxy in said first process;

using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

transmitting said operating system based message to said second process at run time;

decoding, using a second processing means, said operating system based message into a language based message;

providing, using said second processing means, said object oriented programming language based message to said second object.

2. (Once amended) The method of claim 1 further including the steps of:

said second object executing said object oriented programming language based message, using said second processing means, and generating [a] an object oriented programming language based result;

encoding, using said second processing means, said object oriented programming language based result into an operating system based result at run time;

transmitting, using said second processing means, said operating system based result to said first process at run time;

decoding said operating system based result into [a] an object oriented programming language based result at run time, using said first processing means;

providing, using said first processing means, said object oriented programming language based result to said first object.

3. (Once amended) The method of claim 1 wherein said object oriented programming language based message comprises a method and an argument.

5. (Once amended) The method of claim 2 further including the steps of:

said second object generating, using said second processing means, [a] an object oriented programming language based query;

encoding, using said second processing means, said object oriented programming language based query into an operating system based query at run time;

transmitting said operating system based query to said first process at run time, using said second processing means;

decoding, using said first processing means, said operating system based query into [a] an object oriented programming language based query at run time;

a<sup>2</sup> providing, using said first processing means, said object oriented programming language based query to said first object.

6. (Once amended) The method of claim 5 further including the steps of:

said first object generating, using said first processing means, [a] an object oriented programming language based reply;

encoding said object oriented programming language based reply into an operating system based reply at run time, using said first processing means;

transmitting, using said first processing means, said operating system based reply to said second process at run time;

decoding, using said second processing means, said operating system based reply into [a] an object oriented programming language based reply at run time;

providing, using said second processing means, said object oriented programming language based reply to said second object.

8. (Once amended) The method of claim 1 wherein said object oriented programming language based message comprises an objective C message.

a3

sub P<sup>3</sup>

11. (Once amended) A method for sending [a] an object oriented programming language based message having dynamic binding from a first object in a first process to a second object in a second process, said method comprising the steps of:

providing, using a first processing means, said object oriented programming language based message to a first proxy in said first process;

using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

transmitting, using said first processing means, said operating system based message to said second process at run time;

decoding, using said second processing means, said operating system based message into [a] an object oriented programming language based message at run time;

providing, using said second processing means, said object oriented programming language based message to said second object;

said second object generating [a] an object oriented programming language based query, using said second processing means;

creating, using said second processing means, a second proxy in said second process;

a4

10010.929

39

721FH150

WI-Apple0001654



providing, ~~using said second processing means,~~ said object oriented programming language based query to said second proxy;

using said second proxy ~~and said second processing means,~~ encoding said object oriented programming language based query into an operating system based query at run time;

transmitting, ~~using said second processing means,~~ said operating system based query to said first process at run time;

Q 4  
decoding, ~~using said first processing means,~~ said operating system based query into [a] an object oriented programming language based query at run time;

providing, ~~using said first processing means,~~ said object oriented programming language based query to said first object;

said first object generating [a] an object oriented programming language based reply, using said first processing means;

encoding, ~~using said first processing means,~~ said object oriented programming language based reply into an operating system based reply at run time;

transmitting, ~~using said first processing means,~~ said operating system based reply to said second process at run time;

decoding, ~~using a second processing means,~~ said operating system based reply into [a] an object oriented programming language based reply at run time;

a4  
providing, using said second processing means, said object oriented programming language based reply to said second object;

said second object executing said object oriented programming language based message, using said second processing means, and generating [a] an object oriented programming language based result;

encoding, using said second processing means, said object oriented programming language based result into an operating system based result at run time;

transmitting, using said second processing means, said operating system based result to said first process at run time;

decoding, using said first processing means, said operating system based result into [a] an object oriented programming language based result;

providing, using said first processing means, said object oriented programming language based result to said first object.

12. (Once amended) The method of claim 11 wherein said object oriented programming language based message comprises a method and an argument.

a5  
15. (Once amended) The method of claim 11 wherein said object oriented programming language based message comprises an objective C message.

REMARKS

Claims 1-18 have been rejected in the present application, and a new title of the invention is requested. Applicant has amended the title of the invention. Applicant has amended claims 1, 2, 3, 5, 6, 8, 11, 12, and 15. Reconsideration and reexamination of pending claims 1-18 is respectfully requested.

The title of the invention is not descriptive and imprecise. A new title is required which should include more claimed features which differentiate the claimed invention from the Prior Art. The main idea or the improvement of the present invention should be reflected within twenty words. Specifically, the title should stress the use of 'proxy' objects in addition to the translation of a 'language-based' message to an 'operating system-based' message. In general, the title should cite a method of message passing. Applicant has amended the Title accordingly.

The Examiner has objected to the disclosure because of the following informalities:

- (a) In figure 1, blocks 101, 102, 103 & 104 are not labeled.
- (b) Figures 6, 7 & 8b are not labeled as prior art. Appropriate correction is required.

Applicant submits herewith informal drawings with corrections indicated in red.

The Examiner has objected to the specification under 35 U.S.C. § 112, first paragraph, as failing to adequately teach how to make and/or use the invention, i.e. failing to provide an enabling disclosure. It is unclear how a receiver object determines whether it has been given all the information it needs to execute a message, in order to determine if a query must be generated and sent back to the sender object. On p. 22, paragraph 4, the specification cites

that the present invention supports this feature, but does not fully disclose it. This rejection goes hand in hand with section (c) of the following 35 U.S.C. § 112, second paragraph rejection. This feature must be fully disclosed.

Applicant contends that the feature is fully disclosed and is described on pages 22, 23, and 24 of the originally filed specification. Applicant also refers Examiner to Appendix A of the original filed specification, page 41, lines 25 - 40. Applicant contends that the code of Appendix A teaches and discloses the feature described.

5. The Examiner states that Claims 5-7, 11-18 are rejected under 35 U.S.C. § 112, first paragraph, for the reasons set forth in the objection to the specification. Applicant contends that the objection is mooted in light of the above.

The Examiner states that claims 1-18 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The Examiner states that the following claim language is unclear, vague and/or indefinite:

(a) as per claims 1, 2, 5, 6, & 11 it is unclear who or what is executing these steps. If they are executed by a computer, this must be explicitly stated within the context of the claims, and the steps involving 'providing' must be clarified in relation to a computer actually implementing these steps. A computer 'providing' something is not a common art term. It appears to the examiner that these steps do not actually contribute to the claimed method of message sending.

Applicant has amended these claims accordingly.

(b) as per claims 2-10, they are rejected because they depend from previously rejected claim 1. In addition, claims 8 & 9 add irrelevant limitations to the

claims from which they depend. Of what relevance is the use of C and Mach to the method of claim 1?

Applicant contends that claims 2 - 10 are now dependent on an allowable base claim and are therefore themselves allowable.

(c) as per claims 5 & 6, it is unclear how the steps of claim 5 fit into the steps of claim 2, and how the steps of claim 6 fit into the steps of claim 5. Are the steps of these claims executed directly after those of the claim from which they depend? Are they executed every time a message is sent? These seem to be steps which only occur under certain conditions, i.e. the second object needs further information from the first object. This claim must be clarified in conjunction with the above 35 U.S.C. § 112, first paragraph rejection.

Applicant contends that claims 5 and 6 are in proper form and that no further amendment is necessary in light of the mooted of the above section 112 rejections.

(d) as per claims 12-18, they are rejected because they depend from previously rejected claim 11. In addition, claims 15 & 16 add irrelevant limitations to the claims from which they depend. Of what relevance is the use of c and Mach to the method of claim 11?

Applicant contends that claims 12 - 18, being dependent on an allowable base claim, are themselves allowable.

The Examiner states that Claims 1-4, 8-10 are rejected under 35 U.S.C. § 102 (a), (b) & (e) as being anticipated by McCullough, "Transparent Forwarding: First Steps", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 331-341, 12/1987, and Bennet, "The Design and Implementation of Distributed Smalltalk", OOPSLA '87 Proceedings: Conference on Object Oriented

Programming, Systems, Languages, and Applications, pp. 318-330, 12/1987.  
McCullough and Bennet both teach all that is claimed in these claims.

As stated in the "BACKGROUND ART" section of the specification, p. 10, paragraph 3, and in the reference, McCullough clearly teaches the use of a 'ProxyObject' and translation (encoding/decoding) of the message before and after transmission. This is taught in both directions, i.e. sending the message and the result. The translation from a language-based message into an operating system-based message is shown in the use of the `doesNotUnderstand:` primitive, the creation of an Ethernet packet, and the linearization of the arguments to the message. These features clearly indicate a system-dependent form of the message which is transmitted to the receiver object. This system-dependent form is then decoded, the message extracted, and the message executed by the receiver object. The result is then transmitted back to the sender in the same manner.

Bennet teaches a similar method of translation. His method includes the use of the `doesNotUnderstand:`, `perform:`, and `remoteSend:` primitives, a 'RemoteObjectTable' which uses a 'messageProcess' to construct a messageArray, and the encoding of an 'argument string'. These features also indicate a system-dependent form which is used during the transmission of the message.

Applicant respectfully disagrees. Applicant contends that the claims as amended distinguish over the cited art. The present invention is directed to an object oriented programming language based system that includes dynamic binding. This provides features not available to Smalltalk based systems. For example, the Bennett reference does not allow remote classes. Instead, the system requires that classes and instances be co-resident on all processes and machines. This impacts object mobility adversely. Instances can only move to hosts with compatible classes and insuring class compatibility is difficult. In addition, the system of Bennett does not operate in an object-oriented programming system that utilizes class inheritance and reactiveness. (Reactiveness describes the ability of a system to present objects for inspection or modification). Likewise, McCullough does not operate in an

object oriented programming environment having class inheritance. McCullough requires the use of migration. The use of migration limits the performance and ease of use of these prior art schemes. Migration of objects from their home process adds to the complexity of the system. Another disadvantage is that each process and thread must be forked to anticipate each expected iteration. Both references require substantial run time support to implement communication between processes.

The Examiner states that claims 3 & 4 do not contribute to making the claimed invention distinct from the prior art. The limitation, in claim 3, that the message comprises a method and an argument, is disclosed by Applicant in the 'BACKGROUND ART' section of the specification, p. 1; lines 21-25, and is common in the art. It is also understood in the art that executing a message comprises executing the given method on the given argument, as described in claim 4.

Applicant contends that these dependent claims are dependent on an allowable base claim and are themselves allowable.

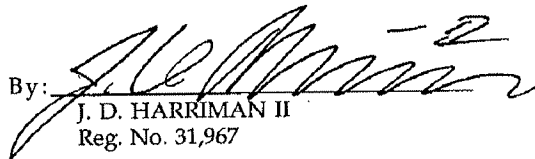
The Examiner states that claims 8 - 10 do not make the invention distinct from the prior art. Regarding claims 8 & 9, see the above 35 USC 112, second paragraph rejection. Regarding claim 10, this limitation is understood in the general understanding of the "proxy" concept.

Applicant contends that these dependent claims are dependent on an allowable base claim and are themselves allowable.

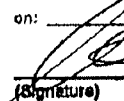
For the foregoing reasons, Applicant contends that none of the cited references, either alone or in combination, teach, suggest, or describe the present invention. Applicant respectfully submits that pending claims 1 - 18 are in condition for allowance.

Respectfully submitted,  
HECKER & HARRIMAN

Date: 3/10/94

By:   
J. D. HARRIMAN II  
Reg. No. 31,967

2049 Century Park East  
Suite 1200  
Los Angeles, California 90067  
(310) 286-0377

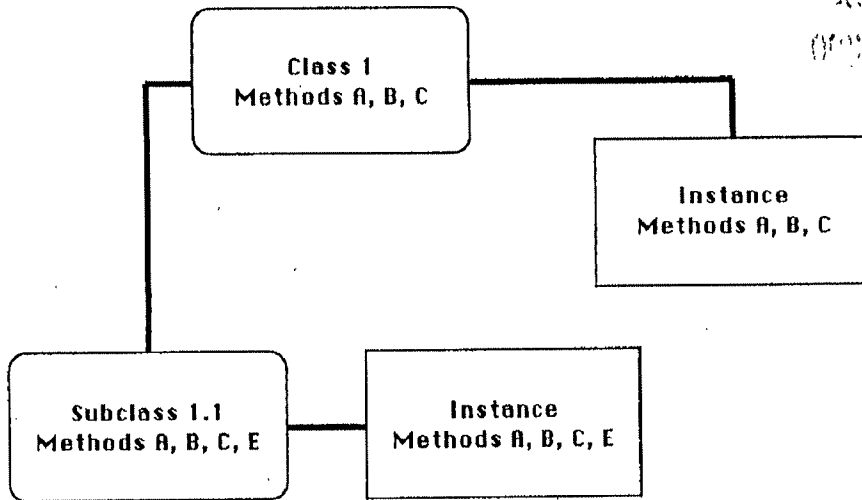
I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail (Label No. B565700080) in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231, on: 3/10/94  
(Date of Deposit)  
 3/10/94  
(Signature) Reg. No. (Date)



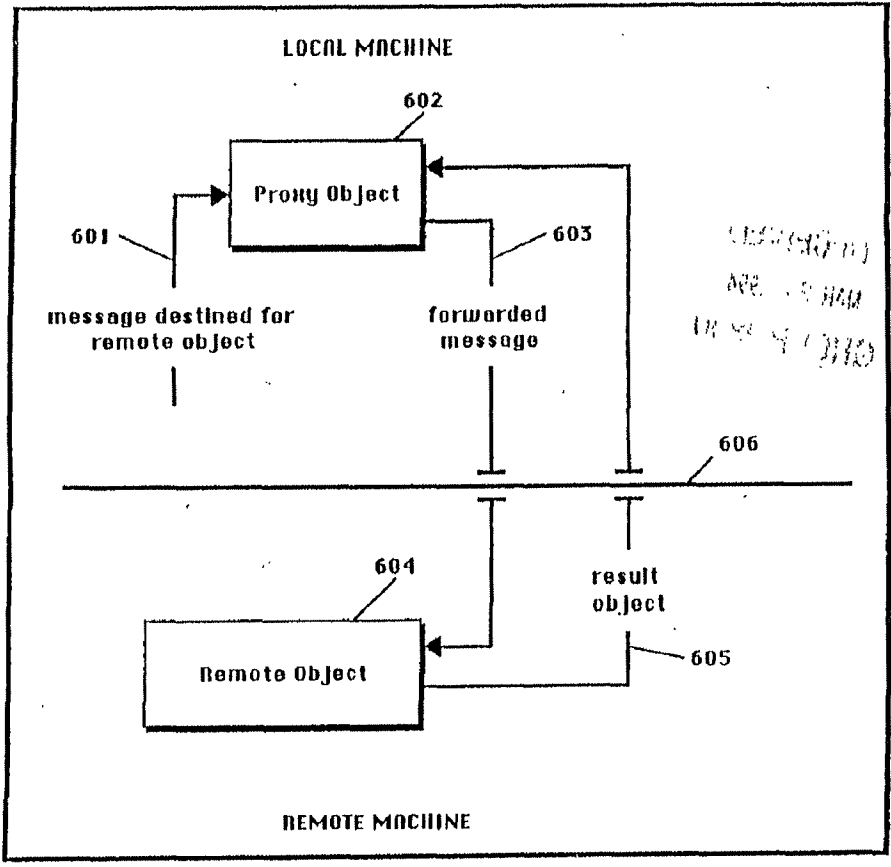
731636  
Part of # 3

Figure 1  
*Peer Act*

COMMISSION  
APR 11 1984  
DOWNEY, CALIF.



*Approved Map*



**FIG. 6**  
*Prior Art*

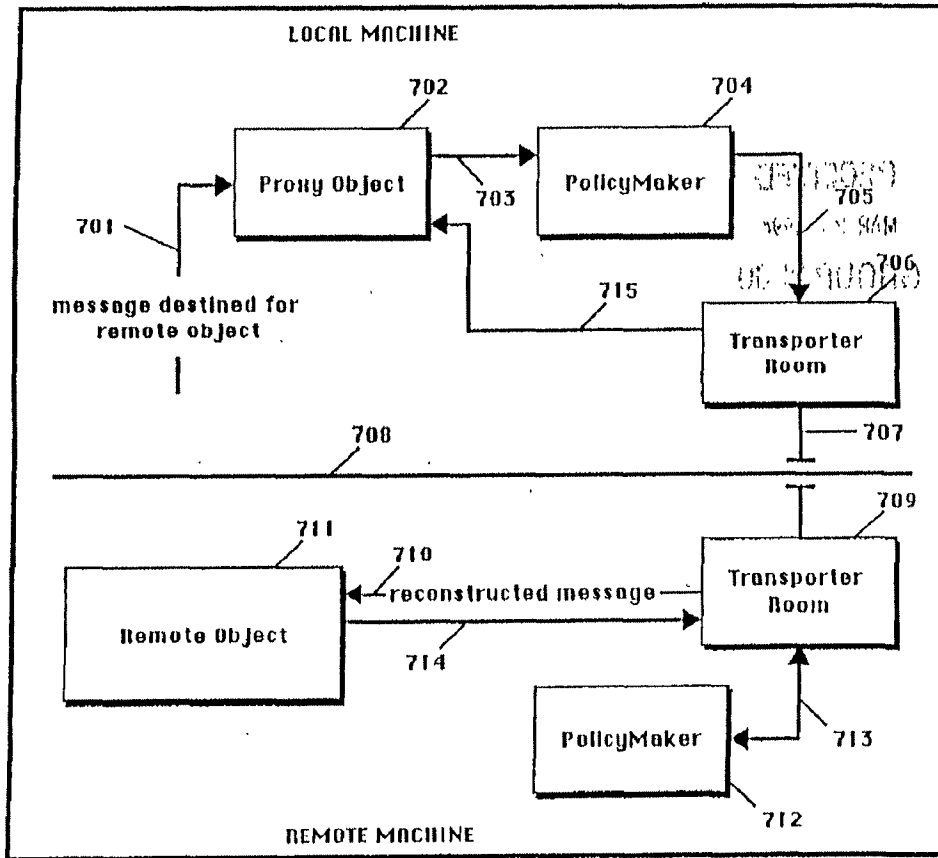


FIG. 7  
Prior Art

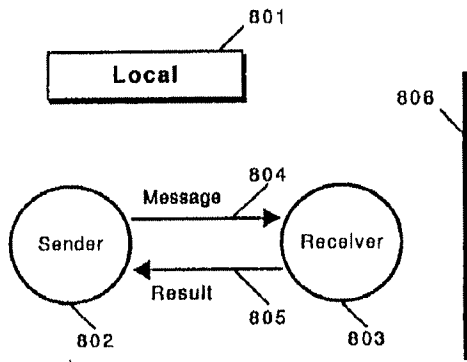


Figure 8A

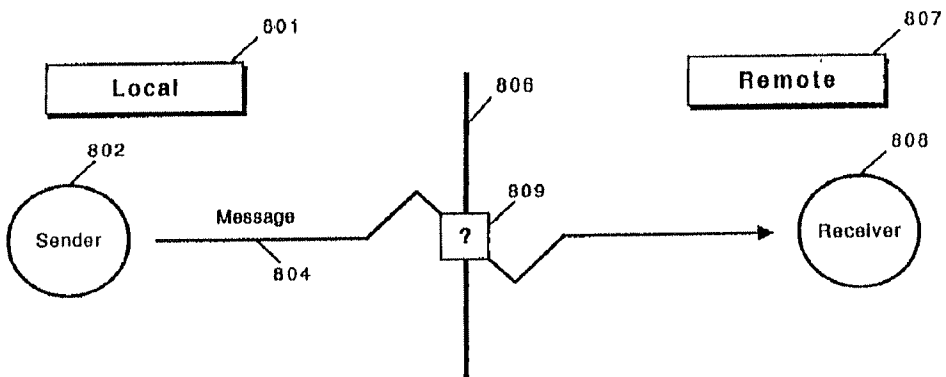


Figure 8B

*Prior Art*



UNITED STATES DEPARTMENT OF COMMERCE  
Patent and Trademark Office

Address : COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
---------------	-------------	----------------------	---------------------

07/731,636 07/17/91 SERLET

B 10010.929

EXAMINER  
PAYNE, M

B3M1/0602

HECKER & HARRIMAN  
2049 CENTURY PARK EAST, STE. 1200  
LOS ANGELES, CA 90067

ART UNIT PAPER NUMBER

2316

5

DATE MAILED: 06/02/94

This is a communication from the examiner in charge of your application.  
COMMISSIONER OF PATENTS AND TRADEMARKS

This application has been examined  Responsive to communication filed on 3/10/94  This action is made final.

A shortened statutory period for response to this action is set to expire 3 month(s), 0 days from the date of this letter.  
Failure to respond within the period for response will cause the application to become abandoned. 35 U.S.C. 133

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

- Notice of References Cited by Examiner, PTO-892.
- Notice re Patent Drawing, PTO-948.
- Notice of Art Cited by Applicant, PTO-1449.
- Notice of Informal Patent Application, Form PTO-152.
- Information on How to Effect Drawing Changes, PTO-1474.
- 

Part II SUMMARY OF ACTION

- Claims 1-18 are pending in the application.  
Of the above, claims \_\_\_\_\_ are withdrawn from consideration.
- Claims \_\_\_\_\_ have been cancelled.
- Claims \_\_\_\_\_ are allowed.
- Claims 1-18 are rejected.
- Claims \_\_\_\_\_ are objected to.
- Claims \_\_\_\_\_ are subject to restriction or election requirement.
- This application has been filed with informal drawings under 37 C.F.R. 1.85 which are acceptable for examination purposes.
- Formal drawings are required in response to this Office action.
- The corrected or substitute drawings have been received on \_\_\_\_\_. Under 37 C.F.R. 1.84 these drawings are  acceptable,  not acceptable (see explanation or Notice re Patent Drawing, PTO-948).
- The proposed additional or substitute sheet(s) of drawings, filed on \_\_\_\_\_, has (have) been  approved by the examiner,  disapproved by the examiner (see explanation).
- The proposed drawing correction, filed on 3/10/94, has been  approved,  disapproved (see explanation).
- Acknowledgment is made of the claim for priority under U.S.C. 119. The certified copy has  been received  not been received  been filed in parent application, serial no. \_\_\_\_\_; filed on \_\_\_\_\_.
- Since this application appears to be 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.
- Other

EXAMINER'S ACTION

PTOL-320 (Rev. 9-88)

721FH164

WI-Apple0001668

Serial No. 07/731636

-2-

Art Unit 2316

1. This office action is filed in response to application, serial number 07/731,636 (Amendment A), filed on March 10, 1994.

2. The objection to the title, as set forth in the previous office action, has been overcome by applicant's amendments.

3. The disclosure is objected to because of the following informalities:

(a) In figure 1, blocks 101,102,103 & 104 are not labelled.

Appropriate correction is required.

4. The following is a quotation of the first paragraph of 35 U.S.C. § 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

The specification is objected to under 35 U.S.C. § 112, first paragraph, as failing to adequately teach how to make and/or use the invention, i.e. failing to provide an enabling disclosure. It is unclear how a receiver object determines whether it has been given all the information it needs to execute a message, in order to determine if a query must be generated and sent back to the sender object. On p.22, paragraph 4, the specification cites that the present invention supports this

721FH165

WI-Apple0001669

Serial No. 07/731636

-3-

Art Unit 2316

feature, but does not fully disclose it. This rejection goes hand in hand with section (c) of the following 35 U.S.C. § 112, second paragraph rejection. This feature must be fully disclosed.

5. Claims 5-7, 11-18 are rejected under 35 U.S.C. § 112, first paragraph, for the reasons set forth in the objection to the specification.

6. Claims 1-18 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The following claim language is unclear, vague and/or indefinite:

(a) as per claims 1,2,5,6 & 11, the steps involving 'providing' are unclear and must be clarified. A computer 'providing' something is not a common art term. Another term indicating a computer-implemented action would be clearer, e.g. sending, transmitting, etc.

(b) as per claims 2-10, they are rejected because they depend from previously rejected claim 1. In addition, claims 8 & 9 add irrelevant limitations to the claims from which they depend. Of what relevance is the use of C and Mach to the method of claim 1?

(c) as per claims 5 & 6, it is unclear how the steps of

721FH166

WI-Apple0001670

Serial No. 07/731636

-4-

Art Unit 2316

claim 5 fit into the steps of claim 2, and how the steps of claim 6 fit into the steps of claim 5. Are the steps of these claims executed directly after those of the claim from which they depend? Are they executed every time a message is sent? These seem to be steps which only occur under certain conditions, i.e. the second object needs further information from the first object. This claim must be clarified in conjunction with the above 35 U.S.C. § 112, first paragraph rejection.

(d) as per claims 12-18, they are rejected because they depend from previously rejected claim 11. In addition, claims 15 & 16 add irrelevant limitations to the claims from which they depend. Of what relevance is the use of C and Mach to the method of claim 11?

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the

721FH167

WI-Apple0001671



Serial No. 07/731636

-5-

Art Unit 2316

applicant for patent.

8. Claims 1-4, 8-10 are rejected under 35 U.S.C. § 102(a), (b) & (e) as being anticipated by McCullough, "Transparent Forwarding: First Steps", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp.331-341, 12/1987, and Bennet, "The Design and Implementation of Distributed Smalltalk", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 318-330, 12/1987. McCullough and Bennet both teach all that is claimed in these claims.

As stated in the 'BACKGROUND ART' section of the specification, p. 10, paragraph 3, and in the reference, McCullough clearly teaches the use of a 'ProxyObject' and translation (encoding/decoding) of the message before and after transmission. This is taught in both directions, i.e. sending the message and the result. The translation from a language-based message into an operating system-based message is shown in the use of the `doesNotUnderstand:` primitive, the creation of an Ethernet packet, and the linearization of the arguments to the message. These features clearly indicate a system-dependent form of the message which is transmitted to the receiver object. This system-dependent form is then decoded, the message extracted, and the message executed by the receiver object. The result is then transmitted back to the sender in the same manner.

721FH168

WI-Apple0001672

Serial No. 07/731636

-6-

Art Unit 2316

Bennet teaches a similar method of translation. His method includes the use of the `doesNotUnderstand:`, `perform:`, and `remoteSend:` primitives, a `RemoteObjectTable` which uses a `messageProcess` to construct a `messageArray`, and the encoding of an `argument string`. These features also indicate a system-dependent form which is used during the transmission of the message.

Claims 3 & 4 do not contribute to making the claimed invention distinct from the prior art. The limitation, in claim 3, that the message comprises a method and an argument, is disclosed by Applicant in the `'BACKGROUND ART'` section of the specification, p.1, lines 21-25, and is common in the art. It is also understood in the art that executing a message comprises executing the given method on the given argument, as described in claim 4.

Claims 8-10 do not make the claimed invention distinct from the prior art. Regarding claims 8 & 9, see the above 35 U.S.C. § 112, second paragraph rejection. Regarding claim 10, this limitation is understood in the general understanding of the `'proxy'` concept.

9. Applicant's arguments filed March 10, 1993 have been fully considered but they are not deemed to be persuasive.

721FH169

WI-Apple0001673

Serial No. 07/731636

-7-

Art Unit 2316

10. The objection to the drawings has been revised in response to Applicant's amendments. The objection to Figure 1 stands, and it appears that this objection may have been misunderstood or overlooked. The BLOCKS 101, 102, 103, & 104, as referred to in the specification, pages 3-4, are not labelled in the figure. Whether this is a prior art figure or not is another question, not being referred to by this objection (the drawing was 'corrected' by adding a prior art label in the last response).

11. The objection to the specification, and the corresponding rejection of claims 5-7, 11-18, under 35 U.S.C. § 112, first paragraph, stands in response to Applicant's amendments. The cited pages of the specification, as well as the portion of the appendix, do not fully disclose nor teach this feature of the invention. There must be an explanation of how this 'query' feature is accomplished, i.e. when a query is done, under what circumstances, based on what factors, how is it implemented, in order to be enabling.

12. The 35 U.S.C. § 112, second paragraph, rejection of claims 1-18 has been revised in response to Applicant's amendments.

13. The rejection of claims 1-4, 8-10 under 35 U.S.C. § 102(a), (b) & (e) stands in response to Applicant's amendments. Applicant's arguments appear to be directed primarily to

721FH170

WI-Apple0001674

Serial No. 07/731636

-8-

Art Unit 2316

perceived differences between the invention and the cited references, rather than to delineating specifically claimed features of the invention which are not taught by the cited references. This rejection is based on the features in the claims, not those in the specification. Applicant's arguments are describing features not in the claims, and as such, are not pertinent.

14. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 C.F.R. § 1.136(a).

A SHORTENED STATUTORY PERIOD FOR RESPONSE TO THIS FINAL ACTION IS SET TO EXPIRE THREE MONTHS FROM THE DATE OF THIS ACTION. IN THE EVENT A FIRST RESPONSE IS FILED WITHIN TWO MONTHS OF THE MAILING DATE OF THIS FINAL ACTION AND THE ADVISORY ACTION IS NOT MAILED UNTIL AFTER THE END OF THE THREE-MONTH SHORTENED STATUTORY PERIOD, THEN THE SHORTENED STATUTORY PERIOD WILL EXPIRE ON THE DATE THE ADVISORY ACTION IS MAILED, AND ANY EXTENSION FEE PURSUANT TO 37 C.F.R. § 1.136(a) WILL BE CALCULATED FROM THE MAILING DATE OF THE ADVISORY ACTION. IN NO EVENT WILL THE STATUTORY PERIOD FOR RESPONSE EXPIRE LATER THAN SIX MONTHS FROM THE DATE OF THIS FINAL ACTION.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Matthew Payne whose telephone number is (703) 305-9593.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

*MMP*  
mmp  
May 24, 1994

KEVIN A. KRIESS  
PRIMARY EXAMINER  
GROUP 2300

721FH171

WI-Apple0001675



6/B  
N.E.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of )  
 )  
 BERTRAND SERLET )  
 )  
 Serial No: 07/731,636 )  
 )  
 Filed: July 17, 1991 )  
 )  
 For: METHOD FOR PROVIDING )  
 )  
 AUTOMATIC AND DYNAMIC )  
 )  
 TRANSLATION OF OBJECT )  
 )  
 ORIENTED PROGRAMMING )  
 )  
 LANGUAGE-BASED MESSAGE )  
 )  
 PASSING INTO OPERATING )  
 )  
 SYSTEM MESSAGE PASSING )  
 )  
 USING PROXY OBJECTS )

Examiner: PAYNE, M  
Group Art Unit: 2316

Kok  
8-30-94

RECEIVED  
AUG 31 1994  
GROUP 2300

AMENDMENT AND RESPONSE

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

Sirs:

This is in response to the outstanding Office Action, paper number 5,  
in the present application. Please amend the present application as follows:

IN THE CLAIMS

1. (Amended) A method for sending an object oriented  
programming language based message having dynamic binding from a first  
object in a first process to a second object in a second process, said method  
comprising the steps of:

[providing] transmitting, using a first processing means, said object  
oriented programming language based message to a first proxy in said first  
process;

using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

transmitting said operating system based message to said second process at run time;

decoding, using a second processing means, said operating system based message into a language based message;

[providing] ~~transmitting~~, using said second processing means, said object oriented programming language based message to said second object.

2. (Amended) The method of claim 1 further including the steps of:

said second object executing said object oriented programming language based message, using said second processing means, and generating an object oriented programming language based result;

encoding, using said second processing means, said object oriented programming language based result into an operating system based result at run time;

transmitting, using said second processing means, said operating system based result to said first process at run time;

decoding said operating system based result into an object oriented programming language based result at run time, using said first processing means;

[providing] ~~transmitting~~, using said first processing means, said object oriented programming language based result to said first object.

5. (Amended) The method of claim 2 wherein the step of executing said object oriented programming language based message further [including] includes the steps of:

said second object determining, using said second processing means, whether additional information is needed to execute said object oriented programming language based message;

said second object generating, using said second processing means, an object oriented programming language based query if it is determined that additional information is needed;

encoding, using said second processing means, said object oriented programming language based query into an operating system based query at run time if it is determined that additional information is needed;

transmitting said operating system based query to said first process at run time, using said second processing means if it is determined that additional information is needed;

decoding, using said first processing means, said operating system based query into an object oriented programming language based query at run time if it is determined that additional information is needed;

[providing] transmitting, using said first processing means, said object oriented programming language based query to said first object if it is determined that additional information is needed.

6. (Amended) The method of claim 5 further including the steps of:

said first object generating, using said first processing means, an object oriented programming language based reply to said object oriented programming language based query;

encoding said object oriented programming language based reply into an operating system based reply at run time, using said first processing means;

transmitting, using said first processing means, said operating system based reply to said second process at run time;

decoding, using said second processing means, said operating system based reply into an object oriented programming language based reply at run time;

[providing] ~~transmitting~~, using said second processing means, said object oriented programming language based reply to said second object.

11. (Amended) A method for sending an object oriented programming language based message having dynamic binding from a first object in a first process to a second object in a second process, said method comprising the steps of:

[providing] ~~transmitting~~, using a first processing means, said object oriented programming language based message to a first proxy in said first process;

using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

~~transmitting~~, using said first processing means, said operating system based message to said second process at run time;

decoding, using said second processing means, said operating system based message into an object oriented programming language based message at run time;

[providing] ~~transmitting~~, using said second processing means, said object oriented programming language based message to said second object;

said second object generating an object oriented programming language based query, using said second processing means;

creating, using said second processing means, a second proxy in said second process;

[providing] ~~transmitting~~, using said second processing means, said object oriented programming language based query to said second proxy;

using said second proxy and said second processing means, encoding said object oriented programming language based query into an operating system based query at run time;

~~transmitting~~, using said second processing means, said operating system based query to said first process at run time;



decoding, using said first processing means, said operating system based query into an object oriented programming language based query at run time;

[providing] transmitting, using said first processing means, said object oriented programming language based query to said first object;

said first object generating an object oriented programming language based reply, using said first processing means;

encoding, using said first processing means, said object oriented programming language based reply into an operating system based reply at run time;

transmitting, using said first processing means, said operating system based reply to said second process at run time;

decoding, using a second processing means, said operating system based reply into an object oriented programming language based reply at run time;

[providing] transmitting, using said second processing means, said object oriented programming language based [message] reply, using said second processing means, and generating an object oriented programming language based result;

encoding, using said second processing means, said object oriented programming language based result into an operating system based result at run time;

transmitting, using said second processing means, said operating system based result to said first process at run time;

decoding, using said first processing means, said operating system based result into an object oriented programming language based result;

[providing] transmitting, using said first processing means, said object oriented programming language based result to said first object.

### REMARKS

The Examiner has rejected Claims 1-18. Applicant has amended Claims 1, 2, 5, 6, and 11. Reexamination and reconsideration of pending Claims 1-18 is respectfully requested.

The Examiner objects to Figure 1 of the disclosure because the blocks contained in the figure are not labeled with their associated reference numerals, 101, 102, 103, and 104. Applicant submits herewith informal drawings with corrections indicated in red.

The Examiner has rejected the specification under 35 U.S.C. §112, first paragraph, as failing to adequately teach how to make and/or use the invention. The Examiner states that it is unclear how a receiver object determines whether it has been given all the information it needs to execute a message to determine whether a query must be generated and sent back to the sender object. The Examiner states that there must be "an explanation of how this 'query' feature is accomplished, i.e., when a query is done, under what circumstances, based on what factors, how is it implemented." (emphasis in original)

Applicant contends that the specification is sufficient to enable one skilled in the art to make and use the invention, and, in particular, the 'query' feature noted by the Examiner.

As stated in the specification beginning at page 2, line 15, an object consists of data and one or more operations, procedures, or methods that can be performed by the data. The data needed and the operations that can be performed on that data are therefore inherent in an object's definition. Because the data and the operations that can be performed on the data are inherent in an object definition, an object can determine what operation(s) can be performed and what data is needed to perform the operation(s).

As stated in the specification beginning at page 22, line 7 for example, local process 901 includes a sender object 905 that sends a message to receiver object 909. Alternatively, the message is sent, for example, via the receiver proxy 904 that executes a *forward::* method to encode the message and transmit the message to the remote object 902. Thus, the receiver object 909 receives the message. The receiver object 909 of the remote process 902 decodes the message encoded as a Mach message, for example, for execution and handling in the remote process 902. The message received by the remote object includes the objective C message.

The remote object uses the information available to the remote object in its definition to determine whether it has been given all the information it needs to execute a message and/or to determine whether a query must be generated and sent back to the sender object for additional information, for example. Where additional information is needed, the present invention supports nested, recursive, remote messages. The specification beginning at page 23, line 6 and Figures 3B-3C provide an illustration of messaging used to obtain additional information.

Applicant therefore contends that the Specification is sufficient to enable one skilled in the art to make and use the invention, and, in particular, the 'query' feature.

The Examiner rejects Claims 5-7 and 11-18 under 35 U.S.C. § 112, first paragraph, for the reasons set forth in the objection to the Specification. Applicant respectfully refers the Examiner to the Applicant's response to the 35 U.S.C. § 112, first paragraph rejection provided above. Given the response provided above, Applicant contends that Claims 5-7 and 11-18 are allowable.

The Examiner rejects Claims 1-18 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and

distinctly claim the subject matter which Applicant regards as the subject matter of the invention.

Regarding Claims 1, 2, 5, 6, and 11, the Examiner states that the "providing" steps are unclear and must be clarified. The Examiner states that another term such as "sending", "transmitting", etc. is clearer. Applicant has amended Claims 1, 2, 5, 6, and 11 accordingly.

Regarding Claims 2-10, the Examiner states that they are rejected because they depend from previously rejected Claim 1. Applicant contends that independent Claim 1 is in condition for allowance. Therefore, Claims 2-10 being dependent on allowable claims are themselves allowable.

Regarding Claims 8 and 9, the Examiner states that they add irrelevant limitations to the claims from which they depend. Applicant respectfully disagrees. Applicant contends that Claims 8 and 9, being dependent on allowable claims, are allowable.

Regarding Claims 5 and 6, the Examiner states that it is unclear how the steps of Claim 5 fit into the steps of Claim 2, and how the steps of Claim 6 fit into the steps of Claim 5. The Examiner states that it is unclear whether the steps of Claims 5 and 6 execute directly after those of the claim upon which they depend. Further, the Examiner states that it is unclear whether these steps are executed every time a message is sent. The Examiner states that the steps seem to occur under certain circumstances such as when the second object needs further information from the first object. Applicant has amended Claims 5 and 6 accordingly. The Examiner states that these claims must be clarified in conjunction with the above 35 U.S.C. § 112, first paragraph rejection. Applicant respectfully refers the Examiner to the previous portion of this response related to the 35 U.S.C. § 112, first paragraph

rejection. Given this, Applicant contends that amended Claims 5 and 6 are now in condition for allowance.

Regarding Claims 12-18, the Examiner states that they are rejected because they depend from previously rejected Claim 11. Further, the Examiner states that Claims 15 and 16 add irrelevant limitations to the claims from which they depend. Applicant respectfully disagrees. Applicant contends that Claims 12-18, being dependent on an allowable independent claim, Claim 11, are therefore allowable.

The Examiner rejects Claims 1-4 and 8-10 under 35 U.S.C. §102(a), (b), and (e) as being unpatentable over McCullough, "Transparent Forwarding: First Steps", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 331-41, 12/1987 (hereinafter referred to as McCullough) and Bennett, "The Design and Implementation of Distributed Smalltalk", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 318-30, 12/1987 (hereinafter referred to as Bennett).

The Examiner states that both Bennett and McCullough indicate a system-dependent form of the message. Regarding McCullough, the Examiner states that the use of the `doesNotUnderstand:` primitive, the creation of an Ethernet packet, and the linearization of the arguments to the message clearly indicate a system-dependent form of the message which is transmitted to the receiver object.

The Examiner states that Bennett teaches a method that uses the `doesNotUnderstand:`, `perform:`, and `remoteSend:` primitives, a "RemoteObjectTable" which uses a "messageProcess" to construct a "messageArray" and the encoding of an "argument string". The Examiner

states that these features indicate a system-dependent form which is used during transmission of the message.

The Examiner states that Applicant's arguments filed March 10, 1993 were not persuasive because they were directed to perceived differences between the invention and the cited references rather than being directed to delineating specifically claimed features of the invention not taught by the cited references.

Applicant is unclear as to the applicability of 35 U.S.C. § 102(e) to the rejection of Claims 1-4 and 8-10 under 35 U.S.C. § 102 as it does not appear that the Examiner is relying on a "patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent" in rejecting Claims 1-4 and 8-10. Applicant requests that the Examiner explicitly state any such reference, if the Examiner continues to hold that this basis for rejection exists relative to Claims 1-4 and 8-10.

Regarding the McCullough reference, Applicant contends that it does not teach a system-dependent form of the message. In fact, McCullough teaches away from a system-dependent form of the message. In McCullough, a textual representation of the selector of the message and a linearization of each of the arguments to the message are encoded in an Ethernet packet. An Ethernet package is not system-dependent. Ethernet is a standardized communication protocol. It is therefore not system-dependent. Rather, it is designed to be system independent. There is nothing in McCullough that describes the encoding of an object oriented programming language based message into an operating system based message as is provided in Claim 1 of the present Application. McCullough merely teaches the encapsulation of a selector and the linearized arguments into an Ethernet packet, a system-independent communication protocol frame format.

Further, in McCullough, there is a single instance of class `TransporterRoom` at a host for each remote machine. The `TransporterRoom` takes care of communications protocols between machines as well as the linearization of messages and objects. The `TransporterRoom` that performs these duties for each remote machine is centralized at a host. Thus, where processes at the same host send a message to the same remote machine, the duties of taking care of communications protocols and linearizing messages and objects is performed external to the processes. These duties are performed by the central `TransporterRoom` instance that is associated with the remote machine at the host.

In contrast, Claim 1 of the present Application provides for the use of a proxy that is resident in the process. One of the functions of the process-resident proxy is to encode an object oriented programming language based message into an operating system based message for transmittal. A host can therefore have multiple proxies that are not central to the host but are resident within each process executing at the host that encode a message for transmittal. When two or more processes at a single host send a message to the same remote object resident on the same remote machine, there are multiple proxies at the single host that are encoding messages for the same remote machine. This is in direct contrast to the scheme used in McCullough where there is only a single `transporterRoom` instance for each remote machine.

In Bennett, there is one `ProxyObject` per host per remote object. There is one `RemoteObjectTable` per host. The single `ProxyObject` on a host represents a remote receiver to the local sender. The one `RemoteObjectTable` on a host represents a remote sender to the local receiver. The central `ProxyObject` is used for sending a message to a remote receiver, and the central `RemoteObjectTable` is used to receive and reply to messages sent by a central `ProxyObject`. Thus, every process executing within a host computer

that wishes to send a message to a remote receiver would use the one, central ProxyObject on the host. The ProxyObject is therefore not part of any one process, but is central to all of the processes executing on the host.

In contrast, Claim 1 of the present Application provides for the use of a proxy that is resident in the process that is sending a message. Because a proxy is resident in a process, every process resident at a host that sends a message can have a proxy. Thus, a host can have multiple proxies that are not central to the host but are resident within each process executing at the host. Further, when two or more processes at a single host send a message to the same remote object, there are multiple proxies at the single host for the one remote object referenced by the host. This is in direct contrast to the scheme used in Bennett where there is only one proxyObject per host per remote object referenced by the host.

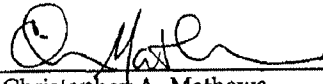
For at least the foregoing reasons, applicant contends that Claims 1-4 and 8-10 are not anticipated by Bennett or McCullough either alone or in combination. Therefore, Applicant contends that Claims 1-4 and 8-10 are allowable.



For the foregoing reasons, applicant contends that none of the references cited, either alone or in combination, teach, describe, or suggest the present invention. Applicant contends that pending Claims 1-18 are allowable.

Respectfully submitted,  
Hecker & Harriman

Date: August 23, 1994

By:   
Christopher A. Mathews  
Reg. No. 35,944

2049 Century Park East  
Suite 1200  
Los Angeles, CA 90067  
(310) 286-0377

File No.: 10010.929

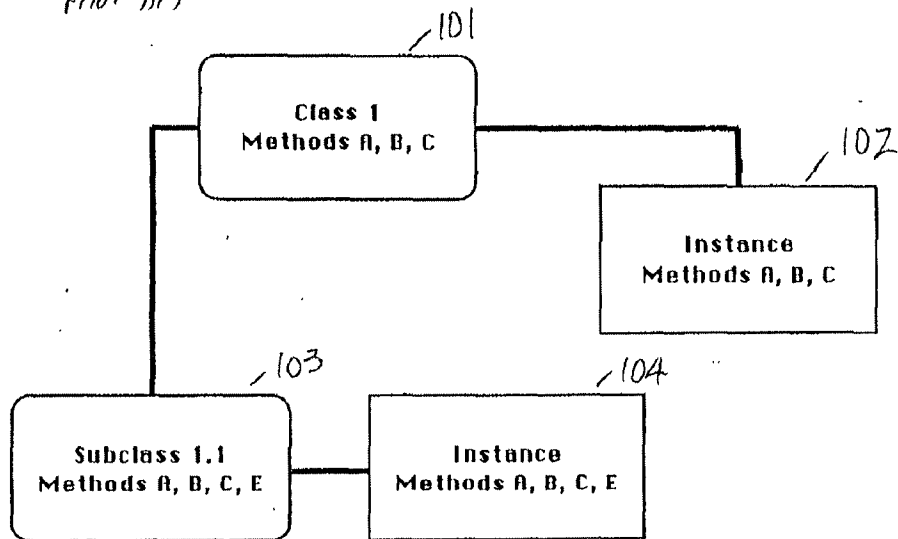
I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:  
Commissioner of Patents and Trademarks, Washington, D.C. 20231, on Aug 23, 1994  
Quonda Goybal 812894  
Signature Reg. No. Date

~~I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail (Label No. 1516371952A) in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on: 08-23-94  
(Date of Deposit)~~

~~Quonda Goybal 812894  
(Signature) Reg. No. (Date)~~

731 636

Figure 1  
Prior Art



Approved MHP 9/6/94



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#11  
J.E.  
12/14/94

In re the Application of )  
BERTRAND SERLET )  
Serial No: 07/731,636 )  
Filed: July 17, 1991 )  
For: METHOD FOR PROVIDING )  
AUTOMATIC AND DYNAMIC )  
TRANSLATION OF OBJECT )  
ORIENTED PROGRAMMING )  
LANGUAGE-BASED MESSAGE )  
PASSING INTO OPERATING )  
SYSTEM MESSAGE PASSING )  
USING PROXY OBJECTS )

Examiner: PAYNE, M  
Group Art Unit: 2316

3-30-94

RECEIVED  
AUG 31 1994  
GROUP 2300

AMENDMENT AND RESPONSE

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

Sirs:

This is in response to the outstanding Office Action, paper number 5,  
in the present application. Please amend the present application as follows:

IN THE CLAIMS

1. (Amended) A method for sending an object oriented  
programming language based message having dynamic binding from a first  
object in a first process to a second object in a second process, said method  
comprising the steps of:

[providing] transmitting, using a first processing means, said object  
oriented programming language based message to a first proxy in said first  
process;


using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

transmitting said operating system based message to said second process at run time;

decoding, using a second processing means, said operating system based message into a language based message;

[providing] ~~transmitting~~, using said second processing means, said object oriented programming language based message to said second object.

2. (Amended) The method of claim 1 further including the steps of:

 said second object executing said object oriented programming language based message, using said second processing means, and generating an object oriented programming language based result;


encoding, using said second processing means, said object oriented programming language based result into an operating system based result at run time;

transmitting, using said second processing means, said operating system based result to said first process at run time;

decoding said operating system based result into an object oriented programming language based result at run time, using said first processing means;

[providing] ~~transmitting~~, using said first processing means, said object oriented programming language based result to said first object.

---

 2  
5. (Amended) The method of claim 2 wherein the step of executing said object oriented programming language based message further [including] includes the steps of:

Handwritten mark resembling a stylized 'D' or '2' with a vertical line extending downwards.

said second object determining, using said second processing means, whether additional information is needed to execute said object oriented programming language based message;

said second object generating, using said second processing means, an object oriented programming language based query if it is determined that additional information is needed;

encoding, using said second processing means, said object oriented programming language based query into an operating system based query at run time if it is determined that additional information is needed;

transmitting said operating system based query to said first process at run time, using said second processing means if it is determined that additional information is needed;

decoding, using said first processing means, said operating system based query into an object oriented programming language based query at run time if it is determined that additional information is needed;

[providing] transmitting, using said first processing means, said object oriented programming language based query to said first object if it is determined that additional information is needed.

6. (Amended) The method of claim 5 further including the steps of:

said first object generating, using said first processing means, an object oriented programming language based reply to said object oriented programming language based query;

encoding said object oriented programming language based reply into an operating system based reply at run time, using said first processing means;

transmitting, using said first processing means, said operating system based reply to said second process at run time;

decoding, using said second processing means, said operating system based reply into an object oriented programming language based reply at run time;

Handwritten number '42'.

D<sup>2</sup>

[providing] ~~transmitting~~, using said second processing means, said object oriented programming language based reply to said second object.

---

11. (Amended) A method for sending an object oriented programming language based message having dynamic binding from a first object in a first process to a second object in a second process, said method comprising the steps of:

[providing] ~~transmitting~~, using a first processing means, said object oriented programming language based message to a first proxy in said first process;

using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

D<sup>3</sup>

transmitting, using said first processing means, said operating system based message to said second process at run time;

decoding, using said second processing means, said operating system based message into an object oriented programming language based message at run time;

[providing] ~~transmitting~~, using said second processing means, said object oriented programming language based message to said second object;

said second object generating an object oriented programming language based query, using said second processing means;

creating, using said second processing means, a second proxy in said second process;

[providing] ~~transmitting~~, using said second processing means, said object oriented programming language based query to said second proxy;

using said second proxy and said second processing means, encoding said object oriented programming language based query into an operating system based query at run time;

transmitting, using said second processing means, said operating system based query to said first process at run time;

43

decoding, using said first processing means, said operating system based query into an object oriented programming language based query at run time;

[providing] transmitting, using said first processing means, said object oriented programming language based query to said first object;

said first object generating an object oriented programming language based reply, using said first processing means;

encoding, using said first processing means, said object oriented programming language based reply into an operating system based reply at run time;

transmitting, using said first processing means, said operating system based reply to said second process at run time;

3  
D  
decoding, using a second processing means, said operating system based reply into an object oriented programming language based reply at run time;

[providing] transmitting, using said second processing means, said object oriented programming language based [message] reply, using said second processing means, and generating an object oriented programming language based result;

encoding, using said second processing means, said object oriented programming language based result into an operating system based result at run time;

transmitting, using said second processing means, said operating system based result to said first process at run time;

decoding, using said first processing means, said operating system based result into an object oriented programming language based result;

[providing] transmitting, using said first processing means, said object oriented programming language based result to said first object.

---

REMARKS

The Examiner has rejected Claims 1-18. Applicant has amended Claims 1, 2, 5, 6, and 11. Reexamination and reconsideration of pending Claims 1-18 is respectfully requested.

The Examiner objects to Figure 1 of the disclosure because the blocks contained in the figure are not labeled with their associated reference numerals, 101, 102, 103, and 104. Applicant submits herewith informal drawings with corrections indicated in red.

The Examiner has rejected the specification under 35 U.S.C. §112, first paragraph, as failing to adequately teach how to make and/or use the invention. The Examiner states that it is unclear how a receiver object determines whether it has been given all the information it needs to execute a message to determine whether a query must be generated and sent back to the sender object. The Examiner states that there must be "an explanation of how this 'query' feature is accomplished, i.e., when a query is done, under what circumstances, based on what factors, how is it implemented." (emphasis in original)

Applicant contends that the specification is sufficient to enable one skilled in the art to make and use the invention, and, in particular, the 'query' feature noted by the Examiner.

As stated in the specification beginning at page 2, line 15, an object consists of data and one or more operations, procedures, or methods that can be performed by the data. The data needed and the operations that can be performed on that data are therefore inherent in an object's definition. Because the data and the operations that can be performed on the data are inherent in an object definition, an object can determine what operation(s) can be performed and what data is needed to perform the operation(s).



As stated in the specification beginning at page 22, line 7 for example, local process 901 includes a sender object 905 that sends a message to receiver object 909. Alternatively, the message is sent, for example, via the receiver proxy 904 that executes a *forward::* method to encode the message and transmit the message to the remote object 902. Thus, the receiver object 909 receives the message. The receiver object 909 of the remote process 902 decodes the message encoded as a Mach message, for example, for execution and handling in the remote process 902. The message received by the remote object includes the objective C message.

The remote object uses the information available to the remote object in its definition to determine whether it has been given all the information it needs to execute a message and/or to determine whether a query must be generated and sent back to the sender object for additional information, for example. Where additional information is needed, the present invention supports nested, recursive, remote messages. The specification beginning at page 23, line 6 and Figures 3B-3C provide an illustration of messaging used to obtain additional information.

Applicant therefore contends that the Specification is sufficient to enable one skilled in the art to make and use the invention, and, in particular, the 'query' feature.

The Examiner rejects Claims 5-7 and 11-18 under 35 U.S.C. § 112, first paragraph, for the reasons set forth in the objection to the Specification. Applicant respectfully refers the Examiner to the Applicant's response to the 35 U.S.C. § 112, first paragraph rejection provided above. Given the response provided above, Applicant contends that Claims 5-7 and 11-18 are allowable.

The Examiner rejects Claims 1-18 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and

distinctly claim the subject matter which Applicant regards as the subject matter of the invention.

Regarding Claims 1, 2, 5, 6, and 11, the Examiner states that the "providing" steps are unclear and must be clarified. The Examiner states that another term such as "sending", "transmitting", etc. is clearer. Applicant has amended Claims 1, 2, 5, 6, and 11 accordingly.

Regarding Claims 2-10, the Examiner states that they are rejected because they depend from previously rejected Claim 1. Applicant contends that independent Claim 1 is in condition for allowance. Therefore, Claims 2-10 being dependent on allowable claims are themselves allowable.

Regarding Claims 8 and 9, the Examiner states that they add irrelevant limitations to the claims from which they depend. Applicant respectfully disagrees. Applicant contends that Claims 8 and 9, being dependent on allowable claims, are allowable.

Regarding Claims 5 and 6, the Examiner states that it is unclear how the steps of Claim 5 fit into the steps of Claim 2, and how the steps of Claim 6 fit into the steps of Claim 5. The Examiner states that it is unclear whether the steps of Claims 5 and 6 execute directly after those of the claim upon which they depend. Further, the Examiner states that it is unclear whether these steps are executed every time a message is sent. The Examiner states that the steps seem to occur under certain circumstances such as when the second object needs further information from the first object. Applicant has amended Claims 5 and 6 accordingly. The Examiner states that these claims must be clarified in conjunction with the above 35 U.S.C. § 112, first paragraph rejection. Applicant respectfully refers the Examiner to the previous portion of this response related to the 35 U.S.C. § 112, first paragraph

rejection. Given this, Applicant contends that amended Claims 5 and 6 are now in condition for allowance.

Regarding Claims 12-18, the Examiner states that they are rejected because they depend from previously rejected Claim 11. Further, the Examiner states that Claims 15 and 16 add irrelevant limitations to the claims from which they depend. Applicant respectfully disagrees. Applicant contends that Claims 12-18, being dependent on an allowable independent claim, Claim 11, are therefore allowable.

The Examiner rejects Claims 1-4 and 8-10 under 35 U.S.C. §102(a), (b), and (e) as being unpatentable over McCullough, "Transparent Forwarding: First Steps", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 331-41, 12/1987 (hereinafter referred to as McCullough) and Bennett, "The Design and Implementation of Distributed Smalltalk", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 318-30, 12/1987 (hereinafter referred to as Bennett).

The Examiner states that both Bennett and McCullough indicate a system-dependent form of the message. Regarding McCullough, the Examiner states that the use of the `doesNotUnderstand:` primitive, the creation of an Ethernet packet, and the linearization of the arguments to the message clearly indicate a system-dependent form of the message which is transmitted to the receiver object.

The Examiner states that Bennett teaches a method that uses the `doesNotUnderstand:`, `perform:`, and `remoteSend:` primitives, a "RemoteObjectTable" which uses a "messageProcess" to construct a "messageArray" and the encoding of an "argument string". The Examiner

states that these features indicate a system-dependent form which is used during transmission of the message.

The Examiner states that Applicant's arguments filed March 10, 1993 were not persuasive because they were directed to perceived differences between the invention and the cited references rather than being directed to delineating specifically claimed features of the invention not taught by the cited references.

Applicant is unclear as to the applicability of 35 U.S.C. § 102(e) to the rejection of Claims 1-4 and 8-10 under 35 U.S.C. § 102 as it does not appear that the Examiner is relying on a "patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent" in rejecting Claims 1-4 and 8-10. Applicant requests that the Examiner explicitly state any such reference, if the Examiner continues to hold that this basis for rejection exists relative to Claims 1-4 and 8-10.

Regarding the McCullough reference, Applicant contends that it does not teach a system-dependent form of the message. In fact, McCullough teaches away from a system-dependent form of the message. In McCullough, a textual representation of the selector of the message and a linearization of each of the arguments to the message are encoded in an Ethernet packet. An Ethernet package is not system-dependent. Ethernet is a standardized communication protocol. It is therefore not system-dependent. Rather, it is designed to be system independent. There is nothing in McCullough that describes the encoding of an object oriented programming language based message into an operating system based message as is provided in Claim 1 of the present Application. McCullough merely teaches the encapsulation of a selector and the linearized arguments into an Ethernet packet, a system-independent communication protocol frame format.

Further, in McCullough, there is a single instance of class TransporterRoom at a host for each remote machine. The TransporterRoom takes care of communications protocols between machines as well as the linearization of messages and objects. The TransporterRoom that performs these duties for each remote machine is centralized at a host. Thus, where processes at the same host send a message to the same remote machine, the duties of taking care of communications protocols and linearizing messages and objects is performed external to the processes. These duties are performed by the central TransporterRoom instance that is associated with the remote machine at the host.

In contrast, Claim 1 of the present Application provides for the use of a proxy that is resident in the process. One of the functions of the process-resident proxy is to encode an object oriented programming language based message into an operating system based message for transmittal. A host can therefore have multiple proxies that are not central to the host but are resident within each process executing at the host that encode a message for transmittal. When two or more processes at a single host send a message to the same remote object resident on the same remote machine, there are multiple proxies at the single host that are encoding messages for the same remote machine. This is in direct contrast to the scheme used in McCullough where there is only a single transporterRoom instance for each remote machine.

In Bennett, there is one ProxyObject per host per remote object. There is one RemoteObjectTable per host. The single ProxyObject on a host represents a remote receiver to the local sender. The one RemoteObjectTable on a host represents a remote sender to the local receiver. The central ProxyObject is used for sending a message to a remote receiver, and the central RemoteObjectTable is used to receive and reply to messages sent by a central ProxyObject. Thus, every process executing within a host computer

that wishes to send a message to a remote receiver would use the one, central ProxyObject on the host. The ProxyObject is therefore not part of any one process, but is central to all of the processes executing on the host.

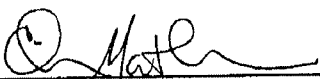
In contrast, Claim 1 of the present Application provides for the use of a proxy that is resident in the process that is sending a message. Because a proxy is resident in a process, every process resident at a host that sends a message can have a proxy. Thus, a host can have multiple proxies that are not central to the host but are resident within each process executing at the host. Further, when two or more processes at a single host send a message to the same remote object, there are multiple proxies at the single host for the one remote object referenced by the host. This is in direct contrast to the scheme used in Bennett where there is only one proxyObject per host per remote object referenced by the host.

For at least the foregoing reasons, applicant contends that Claims 1-4 and 8-10 are not anticipated by Bennett or McCullough either alone or in combination. Therefore, Applicant contends that Claims 1-4 and 8-10 are allowable.

For the foregoing reasons, applicant contends that none of the references cited, either alone or in combination, teach, describe, or suggest the present invention. Applicant contends that pending Claims 1-18 are allowable.

Respectfully submitted,  
Hecker & Harriman

Date: August 23, 1994

By:   
Christopher A. Mathews  
Reg. No. 35,944

2049 Century Park East  
Suite 1200  
Los Angeles, CA 90067  
(310) 286-0377

File No.: 10010.929

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:  
Commissioner of Patents and Trademarks, Washington, D.C. 20231, on Aug 23, 1994  
Quonda Raybal 812394  
Signature Reg. No. Date

~~I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail (Label No. 15137652A) in an envelope addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231 on: 08-23-94  
Quonda A. Raybal 812394  
(Signature) (Date of Deposit) (Reg. No.) (Date)~~



UNITED STATES DEPARTMENT OF COMMERCE  
Patent and Trademark Office

Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
---------------	-------------	----------------------	---------------------

08/332,486 10/31/94 SERLET

B 10010.929C

PAYNE, M EXAMINER

B3M1/1222

ART UNIT	PAPER NUMBER
----------	--------------

J. D. HARRIMAN II  
SUITE 1600  
2029 CENTURY PARK EAST  
LOS ANGELES, CA 90067

12

2316

DATE MAILED: 12/22/94

This is a communication from the examiner in charge of your application.  
COMMISSIONER OF PATENTS AND TRADEMARKS

This application has been examined  Responsive to communication filed on 10/31/94  This action is made final.

A shortened statutory period for response to this action is set to expire 3 month(s), 0 days from the date of this letter.  
Failure to respond within the period for response will cause the application to become abandoned. 35 U.S.C. 133

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

- Notice of References Cited by Examiner, PTO-892.
- Notice of Draftsman's Patent Drawing Review, PTO-948.
- Notice of Art Cited by Applicant, PTO-1449.
- Notice of Informal Patent Application, PTO-152.
- Information on How to Effect Drawing Changes, PTO-1474.
- \_\_\_\_\_

Part II SUMMARY OF ACTION

1.  Claims 1-18 are pending in the application.

Of the above, claims \_\_\_\_\_ are withdrawn from consideration.

2.  Claims \_\_\_\_\_ have been cancelled.

3.  Claims \_\_\_\_\_ are allowed.

4.  Claims 1-18 are rejected.

5.  Claims \_\_\_\_\_ are objected to.

6.  Claims \_\_\_\_\_ are subject to restriction or election requirement.

7.  This application has been filed with informal drawings under 37 C.F.R. 1.85 which are acceptable for examination purposes.

8.  Formal drawings are required in response to this Office action.

9.  The corrected or substitute drawings have been received on \_\_\_\_\_. Under 37 C.F.R. 1.84 these drawings are  acceptable;  not acceptable (see explanation or Notice of Draftsman's Patent Drawing Review, PTO-948).

10.  The proposed additional or substitute sheet(s) of drawings, filed on \_\_\_\_\_, has (have) been  approved by the examiner;  disapproved by the examiner (see explanation).

11.  The proposed drawing correction, filed \_\_\_\_\_, has been  approved;  disapproved (see explanation).

12.  Acknowledgement is made of the claim for priority under 35 U.S.C. 119. The certified copy has  been received  not been received  been filed in parent application, serial no. \_\_\_\_\_; filed on \_\_\_\_\_.

13.  Since this application appears to be 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.

14.  Other

EXAMINER'S ACTION

PTOL-325 (Rev. 2/93)

721FH210

WI-Apple0001714



Serial No. 08/332,486

-2-

Art Unit 2316

1. This office action is filed in response to application, serial number 08/332,486 (Preliminary Amendment D), filed on October 31, 1994.

2. The following is a quotation of the first paragraph of 35 U.S.C. § 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

The specification is objected to under 35 U.S.C. § 112, first paragraph, as failing to adequately teach how to make and/or use the invention, i.e. failing to provide an enabling disclosure. It is unclear how a receiver object determines whether it has been given all the information it needs to execute a message, in order to determine if a query must be generated and sent back to the sender object. On p.22, paragraph 4, the specification cites that the present invention supports this feature, but does not fully disclose it. This rejection goes hand in hand with section (c) of the following 35 U.S.C. § 112, second paragraph rejection. This feature must be fully disclosed.

3. Claims 5-7, 11-18 are rejected under 35 U.S.C. § 112, first paragraph, for the reasons set forth in the objection to the

721FH211

WI-Apple0001715

Serial No. 08/332,486

-3-

Art Unit 2316

specification.

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-4, 8-10 are rejected under 35 U.S.C. § 102(a)&(b) as being anticipated by Bennet, "The Design and Implementation of Distributed Smalltalk", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 318-330, 12/1987. Bennet teaches all that is claimed.

Bennet clearly teaches the use of a 'ProxyObject' and translation (encoding/decoding) of the message before and after transmission. This is taught in both directions, i.e. sending the message and the result. The translation from a language-based message into an operating system-based message is shown in the use of the `doesNotUnderstand:`, `perform:`, and `remoteSend:` primitives, a 'RemoteObjectTable' which uses a 'messageProcess' to construct a messageArray, and the encoding of an 'argument

721FH212

WI-Apple0001716

Serial No. 08/332,486

-4-

Art Unit 2316

string'. These features clearly indicate a system-dependent form of the message which is transmitted to the receiver object. This system-dependent form is then decoded, the message extracted, and the message executed by the receiver object. The result is then transmitted back to the sender in the same manner.

Claims 3 & 4 do not contribute to making the claimed invention distinct from the prior art. The limitation, in claim 3, that the message comprises a method and an argument, is disclosed by Applicant in the 'BACKGROUND ART' section of the specification, p.1, lines 21-25, and is common in the art. It is also understood in the art that executing a message comprises executing the given method on the given argument, as described in claim 4.

Claims 8-10 do not make the claimed invention distinct from the prior art. Regarding claims 8 & 9, see the above 35 U.S.C. § 112, second paragraph rejection. Regarding claim 10, this limitation is understood in the general understanding of the 'proxy' concept.

6. Applicant's arguments filed October 31, 1994 have been fully considered but they are not deemed to be persuasive.

7. The objection to the drawings has been overcome in response to Applicant's amendments.

721FH213

WI-Apple0001717

Serial No. 08/332,486

-5-

Art Unit. 2316

8. The objection to the specification, and the corresponding rejection of claims 5-7, 11-18, under 35 U.S.C. § 112, first paragraph, stands in response to Applicant's amendments. The cited pages of the specification, as well as the portion of the appendix, do not fully disclose nor teach this feature of the invention. Applicants remarks describing the inherent ability of objects to determine if all necessary information is present for execution of a method is not persuasive. It is common to expect that if an object receives an incomplete method call, that it would produce an error message. However, this is far from the described capability of an object to determine whether additional information is needed for execution, to generate a 'query' if more information is needed, and for the sender to have the functionality to understand this 'query' and respond with a 'reply'.

9. The 35 U.S.C. § 112, second paragraph, rejection of claims 1-18 has been overcome in response to Applicant's amendments.

10. The rejection of claims 1-4, 8-10 under 35 U.S.C. § 102(a)&(b) has been revised in response to Applicant's amendments. The rejection over McCullough has been overcome by Applicants amendments. The rejection under section (e) of 35 U.S.C. § 102 was inadvertent, and has been removed. Applicant's arguments appear to be directed primarily to perceived

721FH214

WI-Apple0001718

Serial No. 08/332,486

-6-

Art Unit 2316

differences between the invention and the cited references, rather than to delineating specifically claimed features of the invention which are not taught by the cited references. This rejection is based on the features in the claims, not those in the specification. Applicant's arguments are describing features not in the claims, and as such, are not pertinent. Applicants arguments regarding the number and placement of the ProxyObject are not persuasive. Claim 1 describes the transmission of a message to a first proxy in said first process. Bennet clearly describes ProxyObjects present in each host, this host inherently being a process address space.

11. A shortened statutory period for response to this action is set to expire 3 (three) months, 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will cause the application to become ABANDONED according to 35 U.S.C. 133, M.P.E.P. 710.02, 710.02(b).

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Matthew Payne whose telephone number is (703) 305-9593.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

MMP  
mmp  
December 21, 1994

  
KEVIN A. KRIESS  
PRIMARY EXAMINER  
GROUP 2300

721FH215

WI-Apple0001719

88-103 GP 2316

#13/  
E  
3-22-95



10010.929C

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Handwritten notes in a vertical column on the left side of the page, including "322-95" and other illegible markings.

In re the Application of  
BERTRAND SERLET et al.  
Serial No: 08/332,486  
Filed: October 31, 1994  
For: METHOD FOR PROVIDING  
AUTOMATIC AND DYNAMIC  
TRANSLATION OF OBJECT-ORIENTED  
PROGRAMMING LANGUAGE BASED  
MESSAGE PASSING INTO OPERATING  
SYSTEM MESSAGE PASSING USING  
PROXY OBJECTS

Examiner: M. PAYNE  
Group Art Unit: 2316

RECEIVED  
MAR 16 1995  
GROUP 2300

AMENDMENT AND RESPONSE

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

Sirs:

This is in response to the outstanding Office Action (paper number 12)  
in the present application. Please amend the present application as follows:

IN THE CLAIMS

1. (Amended) A method for sending an object oriented  
programming language based message having dynamic binding from a first  
object in a first process to a second object in a second process, said method  
comprising the steps of:

transmitting, using a first processing means, said object oriented  
programming language based message to a first proxy in said first process;

090 BA 03/15/95 08332486 1 103 88.00 CK

Handwritten mark "E" on the left side.

Handwritten mark "2/5" at the bottom center.

using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

transmitting said operating system based message to said second process in said second processing means at run time;

decoding, using a second [processing means] process, said operating system based message into a language based message;

transmitting, using said second processing means, said object oriented programming language based message to said second object in said second process];

executing said object oriented programming language based message by said second object in said second process.

2. (Amended) The method of claim 1 further including the steps of:

said second object in said second process [executing said object oriented programming language based message, using said second processing means,] and] generating an object oriented programming language based result;

encoding, using said second processing means, said object oriented programming language based result into an operating system based result at run time;

transmitting, using said second processing means], said operating system based result to said first process at run time;

decoding said operating system based result into an object oriented programming language based result at run time, using said first processing means;

transmitting, using said first processing means, said object oriented programming language based result to said first object.

4. (Amended) The method of claim [3] 1 wherein said second object executes said method on said argument when executing said message.

10010.929C

4/2

721FH217

WI-Apple0001721

5. (Amended) The method of claim [2] 1 wherein the step of executing said object oriented programming language based message further includes the steps of:

said second object determining, using said second processing means, whether additional information is needed to execute said object oriented programming language based message;

said second object generating, using said second processing means, an object oriented programming language based query if it is determined that additional information is needed;

E2 encoding, using said second processing means, said object oriented programming language based query into an operating system based query at run time if it is determined that additional information is needed;

transmitting said operating system based query to said first process at run time, using said second processing means if it is determined that additional information is needed;

decoding, using said first processing means, said operating system based query into an object oriented programming language based query at run time if it is determined that additional information is needed;

transmitting, using said first processing means, said object oriented programming language based query to said first object if it is determined that additional information is needed.

E3 7. (Amended) The method of claim 6 wherein said first [process] processing means and said second [process] processing means are [located on first and second computers respectively] the same processing means.

E4 Please add the following new claims:

19. A method for sending, in a C environment with minimal run time support, an object oriented programming language based message having dynamic binding from a first object in a first process to a second object in a second process, said method comprising the steps of:

10010.929C

4/3

721FH218

WI-Apple0001722



transmitting, using a first processing means implementing said C environment, said object oriented programming language based message to a first proxy in said first process;

using said first proxy and said first processing means, encoding said object oriented programming language based message into an operating system based message at run time;

transmitting said operating system based message to said second process at run time;

decoding, using a second processing means implementing said C environment, said operating system based message into a language based message;

transmitting, using said second processing means, said object oriented programming language based message to said second object.\*

\*20. The method of claim 19 further including the steps of:

E4  
said second object executing said object oriented programming language based message, using said second processing means, and generating an object oriented programming language based result;

encoding, using said second processing means, said object oriented programming language based result into an operating system based result at run time;

transmitting, using said second processing means, said operating system based result to said first process at run time;

decoding said operating system based result into an object oriented programming language based result at run time, using said first processing means;

transmitting, using said first processing means, said object oriented programming language based result to said first object.\*\*

\*21. The method of claim 20 wherein the step of executing said object oriented programming language based message further includes the steps of:

said second object determining, using said second processing means, whether additional information is needed to execute said object oriented programming language based message;

said second object generating, using said second processing means, an object oriented programming language based query if it is determined that additional information is needed;

encoding, using said second processing means, said object oriented programming language based query into an operating system based query at run time if it is determined that additional information is needed;

transmitting said operating system based query to said first process at run time, using said second processing means if it is determined that additional information is needed;

decoding, using said first processing means, said operating system based query into an object oriented programming language based query at run time if it is determined that additional information is needed;

transmitting, using said first processing means, said object oriented programming language based query to said first object if it is determined that additional information is needed.

E4

22. The method of claim 21 further including the steps of:

said first object generating, using said first processing means, an object oriented programming language based reply to said object oriented programming language based query;

encoding said object oriented programming language based reply into an operating system based reply at run time, using said first processing means;

transmitting, using said first processing means, said operating system based reply to said second process at run time;

decoding, using said second processing means, said operating system based reply into an object oriented programming language based reply at run time;

10010.929C

495

721FH220

WI-Apple0001724

transmitting, using said processing means, said object oriented programming language based reply to said second object.¶

E4

¶23. The method of claim 21 wherein said operating system based message comprises a Mach message.¶¶

¶24. The method of claim 21 wherein said first processing means and said second processing means are the same processing means.¶¶

---

REMARKS

The Examiner has rejected claims 1-18. Applicant has amended claims 1, 2, 4, 5, and 7. Applicant has added new claims 19-24. Reexamination and reconsideration of pending claims 1-24 is respectfully requested.

OBJECTION TO THE SPECIFICATION BASED ON 35 U.S.C. § 112, FIRST PARAGRAPH

The Examiner has rejected the specification under 35 U.S.C. §112, first paragraph, as failing to adequately teach how to make and/or use the invention. The Examiner states that it is unclear how a receiver object determines whether it has been given all the information it needs to execute a message to determine whether a query must be generated and sent back to the sender object. The Examiner states that there must be "an explanation of how this 'query' feature is accomplished, i.e., when a query is done, under what circumstances, based on what factors, how is it implemented." (emphasis in original) The Examiner states:

"[i]t is common to expect that if an object receives an incomplete method call, that it would produce an error message. However, this is far from the described capability of an object to determine whether additional information is needed for execution, to generate a 'query' if more information is needed, and for the sender to have the functionality to understand this 'query' and respond with a 'reply'."

Applicant contends that the specification is sufficient to enable one skilled in the art to make and use the invention, and, in particular, the 'query' feature noted by the Examiner.

Applicant respectfully refers the Examiner to the specification starting at page 23, line 6, for example. As indicated in the specification, when a receiver object requires additional information from a sender object, it generates a request message. The sender object receives the request and generates a response which is sent to the receiver object.

10010.929C



721FH222

WI-Apple0001726

Further, the specification, starting at page 23, line 25, provides an example of the recursive nature of the present invention where a remote object does not recognize a method. When a remote object does not recognize a method (e.g., "foo") sent to it by a local object, for example, the remote object asks the sending object "what is foo?" All objects recognize the method "what is". The sender object can then respond with instructions concerning the nature of the method being investigated. As illustrated in the specification (starting at page 23, line 30), the sender object might reply that "foo" is a method that requires an integer. Once the receiver object receives the instruction (e.g., a message that "foo" is a method that requires an integer), the receiver object can determine whether or not an integer has been provided. If no integer has been provided, the remote object can then generate a second request to get the integer. The process for obtaining the integer would be similar.

The above example discloses the capability of an object to determine whether additional information is needed, to generate a 'query' to get the additional information, to understand this 'query' and respond to this 'query'. That is, the remote object determines that it does not understand "foo". As a result, the remote object generates a "what is foo?" message. Since all objects understand the "what is foo?" message, the sender is able to understand, and respond to, the query initiated by the remote object. The sender responds by providing a further explanation of "foo". In the example, this explanation includes the fact that "foo" requires an integer. Based on the reply from the sender, the remote object knows that "foo" needs an integer. The remote object can then determine whether it received an integer argument as part of the original message. If it did not receive an integer as part of the original message, the remote object (as indicated in the example) can generate another message to request the integer argument.

Therefore, as indicated in the specification, the remote object determines that it needs additional information for execution (e.g., additional

information regarding "foo") and it generates a 'query' (e.g., "what is foo?") to obtain additional information. Since "what is" is global, the sender has the functionality to understand the 'query'. Further, the sender is able to respond to the 'query'. The sender replies by providing additional information about "foo".

Applicant therefore contends that the Specification is sufficient to enable one skilled in the art to make and use the invention, and, in particular, the 'query' feature.

REJECTION OF CLAIMS 5-7 AND 11-18 BASED ON 35 U.S.C. § 112, FIRST PARAGRAPH

The Examiner rejects Claims 5-7 and 11-18 under 35 U.S.C. § 112, first paragraph, for the reasons set forth in the objection to the Specification. Applicant respectfully refers the Examiner to the Applicant's response to the 35 U.S.C. § 112, first paragraph rejection provided above. Given the response provided above, Applicant contends that Claims 5-7 and 11-18 are allowable.

REJECTION OF CLAIMS 1-4 AND 8-10 BASED ON 35 U.S.C §§ 102(a) & (b)

The Examiner rejects Claims 1-4 and 8-10 under 35 U.S.C. §§ 102(a) and (b) as being unpatentable over Bennett, "The Design and Implementation of Distributed Smalltalk", OOPSLA '87 Proceedings: Conference on Object Oriented Programming, Systems, Languages, and Applications, pp. 318-30, 12/1987 (hereinafter referred to as Bennett).

The Examiner states that Bennett indicate a system-dependent form of the message. The Examiner states that Bennett teaches a method that uses the `doesNotUnderstand:`, `perform:`, and `remoteSend:` primitives, a "RemoteObjectTable" which uses a "messageProcess" to construct a "messageArray" and the encoding of an "argument string". The Examiner

states that these features indicate a system-dependent form which is used during transmission of the message.

The Examiner states that Applicants' arguments filed August 23, 1994 and relating to Bennett were directed to perceived differences between the invention and the cited reference rather than being directed to delineating specifically claimed features of the invention not taught by the cited references. The Examiner states that Claim 1 describes the transmission of a message to a first proxy in said first process. The Examiner states that Bennett clearly describes ProxyObjects present in each host, this host inherently being a process address space.

Applicants respectfully disagree. Applicants contend that Claim 1 differs from Bennett for at least the following reasons:

1. In Bennett, a host is a computer or machine. A process address space is not the same as a host.

2. In Bennett, ProxyObject and RemoteObjectTable are centralized processes that are not included in either the sending or receiving processes.

1. In Bennett, a host is a computer or machine. A process address space is not the same as a host.

The Examiner states that:

"Bennett clearly describes ProxyObjects present in each host, this host inherently being a process address space.

Applicants respectfully disagree. Bennett uses the terms "host" and "machine" interchangeable or in combination to mean the same thing (i.e., a

computer having a processor and memory). On page 318, left-hand column, Bennett states:

"[a]lthough Smalltalk host machines can be interconnected with high bandwidth networks, only rudimentary support exists within Smalltalk for cooperation among users, and no support exists within Smalltalk for object sharing between users, communication between objects that reside on different machines, or cooperation between processes on several machines." (emphasis added.)

Thus, Bennett combines the terms "host" and "machine" in the previous passage. In addition, Bennett uses the terms "host" and "machine" interchangeably. For example, in discussing class replication in the Distributed Smalltalk design in Bennett, option (1) on page 320, left-hand column is chosen. Option (1) states:

"[d]isallow remote classes (i.e., require that classes and instances be co-resident.) This approach impacts object mobility. Instances can only move to hosts with compatible classes. Ensuring class compatibility is hard." (emphasis added.) (page 320, left-hand column)

In further discussion of Option (1), Bennett states (at page 320, left-hand column):

"[c]lass replication in a reactive system can also create problems. If users on different machines are allowed to create incompatible versions of a system class, considerable confusion can ensue if instances of these classes move between machines." (emphasis added.) (page 320, left-hand column)

The following quotes from Bennett further illustrate that the terms "host" and "machine" are used interchangeably in Bennett:

1. ProxyObjects' response to the *doesNotUnderstand:* message is to forward the original message to the RemoteObjectTable on the appropriate machine. (emphasis added.) (page 323, left-hand column.)
2. There is one RemoteObjectTable per host. (emphasis added.) (page 323, right-hand column.)



Therefore, in Bennett, the terms "host" and "machine" are used interchangeably to mean the same thing. In Bennett, both of the terms refer to a computer that has a processor, memory, etc.

The Examiner states that, in Bennett, a host is inherently a "process address space." Applicants respectfully disagree. As indicated above, in Bennett, a host or machine is a computer having a processor and memory. A process address space is the portion of memory that is addressable by a process executing in a computer system. In Bennett, a host or machine is different than a process. In Bennett, a process executes in a host. In Bennett, a RemoteObjectTable executes on a host. As stated in Bennett:

"The RemoteObjectTable has associated with it three processes: the messageProcess, the userProcess, and the kernelProcess. The messageProcess waits for any Distributed Smalltalk message to the host. Upon receipt of such a message, the messageProcess constructs a messageArray containing the remoteSend parameters." (page 324, left-hand column)

Thus, in Bennett, the RemoteObjectTable for a host has three processes that are executing on the host. Each process has a range of memory (or address space) that the process can access. Therefore, a host has at least three process address spaces for each of these processes. Further, as stated in Bennett (page 319, left-hand column), each user has an address space. Thus, in Bennett a host is not a process address space. Rather, a host has multiple address spaces to accommodate, for example, each user and the processes associated with the RemoteObjectTable.

2. In Bennett, ProxyObject and RemoteObjectTable are centralized processes that are not included in either the sending or receiving processes.

In Bennett, a ProxyObject is centralized on a host or machine. In Bennett, a sending process must access the centralized ProxyObject process to forward a message to a remote object on a remote host.

In Bennett, the remote host (remote machine) has a centralized RemoteObjectTable that is responsible for receiving and replying to messages forwarded by the ProxyObject. The receiver process is scheduled by the RemoteObjectTable when a message is received from a proxyObject. The receiver process executes the required action and returns a value to the centralized RemoteObjectTable for transmittal to the ProxyObject. As stated in Bennett:

"The RemoteObjectTable is responsible for receiving and replying to messages forwarded by proxyObjects. There is one RemoteObjectTable per host. It is the sole instance of class RemoteObjectTable .... When the RemoteObjectTable receives a message from some proxyObject, it schedules a process that will contain the execution context of the actual message receiver by sending the message perform to the receiver with the forwarded selector and argument (if any) as arguments to the perform message." (page 323, right-hand column)

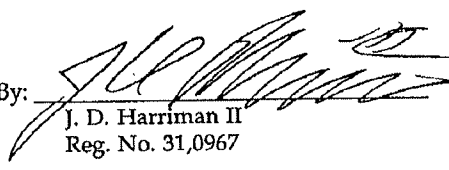
In Bennett, therefore, the ProxyObject is not in the sending process. Further, the process that executes the message is not the same process as the RemoteObjectTable (i.e., the receiving process). Therefore, in Bennett, the sending process, receiving process, ProxyObject process, RemoteObjectTable process are different processes. In contrast, in claim 1 of the present application the first proxy object is contained in the sending process. Further, in claim 1 of the present application, the process that executes the message is the same as the receiving process.

Therefore, Applicants contend that claim 1 is allowable. Further, Applicants contend that claims 2-4 and 8-10 being dependent on an allowable dependent claim are, therefore, allowable.

For the foregoing reasons, applicant contends that none of the references cited, either alone or in combination, teach, describe, or suggest the present invention. Applicant contends that pending claims 1-24 are allowable.

Respectfully submitted,  
Hecker & Harriman

Date: 2/22/95

By:   
J. D. Harriman II  
Reg. No. 31,0967

2029 Century Park East  
Suite 1600  
Los Angeles, CA 90067  
(310) 286-0377

File No.: 10010.929C

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to:  
Commissioner of Patents and Trademarks, Washington, D.C. 20231, on February 22, 1995  
2/22/95 Rhonda Royal  
Signature                      Reg. No.                      Date