Exhibit 18

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Apple Inc.

Ex Parte Reexamination 90/010,965

U.S. Patent No. 5,455,854 to Dilts *et al.*

December 14, 2010

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Agenda

I. Introductions

II. Status of Litigation

III. Overview of Patentability

IV. Story of the Invention

V. NeXTSTEP I and II

VI. Discussion of the Office Action

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Introductions

- Technical Expert:
  - David A. Wilson, Ph.D.
- Inventor:
  - Michael R. Dilts, M.A.
- Apple:
  - R. "Chip" Lutton (Chief Patent Counsel), Reg. No. 39,756
- SKGF:
  - Robert G. Sterne, Reg. No. 28,912
  - Glenn J. Perry, Reg. No. 28,458
  - Richard D. Coller III, Reg. No. 60,390
  - Salvador M. Bezos, Reg. No. 60,889

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

David A. Wilson, Ph.D. (expert)

- Programming
  - 1966: IBM Federal Systems Division - 7094 mainframe in assembly language ...
  - 1983: self-employed
  - 1984: teach procedural Mac programming

- Object-Oriented Programming
  - 1987: teach MacApp for Apple (OOP, frameworks)
    - lead author on two Addison-Wesley books about MacApp
  - 1988: teach Smalltalk programming for Xerox ParcPlace
  - 1989: teach intro to NextStep programming in Objective-C
  - 1990: teach Pink/Taligent programming in C++
  - 1998: teach advanced Java programming for Sun Microsystems
  - 2008: developed iPhone/iPad apps using Objective-C

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Michael R. Dilts (inventor)

- Education:
  - University of California, Berkeley: A.B. Linguistics
  - Harvard University, M.A. Linguistics
  - Phi Beta Kappa
  - National Science Foundation Graduate Fellow
- Employment:
  - Texas Instruments - Speak n' Spell Voice Encoding Chip
  - Wang Laboratories - Digital Voice Messaging System
  - Apple Computer - Speech and Telephone Integration
  - Apple/Taligent - Object Oriented Telephony (Pink)
  - Apple Computer - International Software
  - Currently Independent Consultant
- Publications:
  - 15 Papers and Articles
  - *Electronic Speech Synthesis.* Granada: 1984 (Contributing Author)
  - *Editing Synthetic Speech.* T.I.Technical Manual: 1982
  - 7 Patents including 854

Status of the Litigation

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Nokia Corp. v. Apple Inc. (D. Del.)

- The '854 patent is in litigation (*Nokia v. Apple*, 1:09-cv-00791 (D. Del.)) before C.J. Sleet

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Apple's Business Model

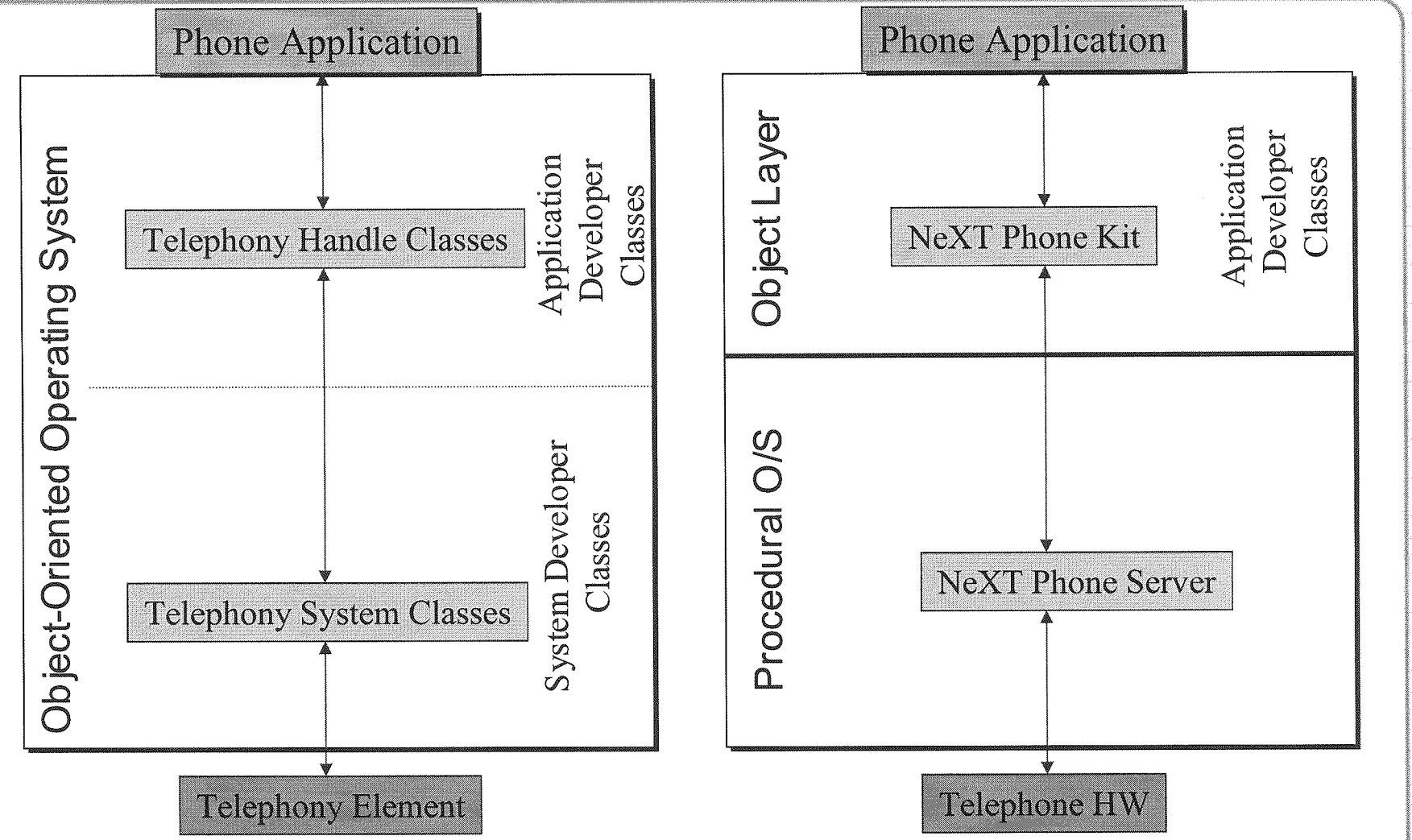
Apple has designed a business strategy based on the convergence of personal computers, mobile communications, and digital consumer electronics, and produced cutting-edge, technologically superior, and user-friendly devices.

See, *Nokia v. Apple*, 1:09-cv-00791, Apple Inc.'s First Amended Answer, p. 3 (Filed Feb. 19, 2010)

Overview of Patentability

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

'854 vs. NeXTSTEP I and II



Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Claims 1 and 13

Claim 1. A telephony apparatus, comprising:

(a) a processor;

(b) a storage attached to and controlled by the processor;

(c) **an object oriented operating system**, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage and controlling operations of the processor;

(d) **a display attached to the processor under the control of the object oriented operating system**;

(e) a telephony element attached to the processor;

(f) a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element **under the control of the object-oriented operating system**, stored in the storage and displayed on the display; and

(g) means for **controlling the telephony element by the object oriented operating system utilizing the logic in the telephony object to interface the telephony element to the processor** by initiating a call connection, monitoring call progress, activating call featurs and **storing status information in the data of the telephony object**.

Claim 13. A method for enabling telephony elements on a computer system, including a processor with an attached storage, display and telephony element, comprising the steps of:

(a) controlling operations of the processor with **an object oriented operating system**, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage;

(b) creating a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element **under the control of the object-oriented operating system**, stored in the storage and displayed on the display; and

(c) **controlling the telephony element by the object-oriented operating system utilizing logic in the telephony object to interface the telephony element to the processor** by initiating a call connection, monitoring call progress, activating call features and **storing status information in the data of the telephony object**.

Story of the Invention

Object-Oriented Programming Benefits

- Managing complexity
- Reusing code
- Simpler program development

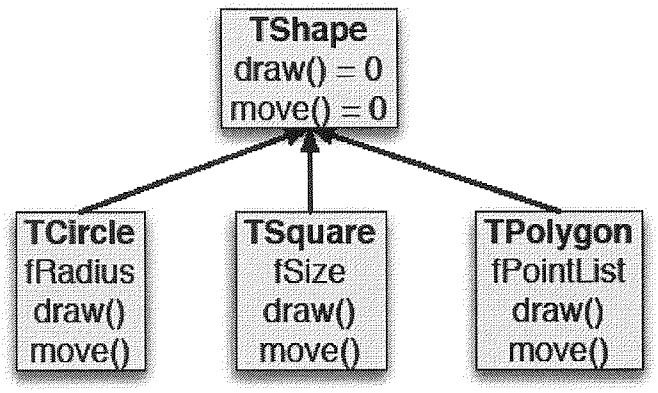
Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

The O-O Approach

- Object-oriented development is better than procedural development
- The '854 patent describes an object-oriented system for the application developer and the system software developer
- Layered approach (NeXTSTEP) does ***not*** provide object-oriented system software

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

OOP Example




```
TCircle* aCircle;

// make instance of class
// i.e., create a circle object
aCircle = new TCircle();

// set value of instance variable
aCircle.fRadius = 30.0;
aCircle.setRadius(45.3);

// call method
aCircle.draw()
```

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

The "Pink" Operating System

- During Apple's Macintosh operating system development discussions, a set of advanced ideas were written on index cards (blue and pink)
- "Blue" became System 7, which remained a procedural operating system
- "Pink" was designed and built to be an object-oriented operating system implemented in C++
- Pink development team was spun-off as Taligent

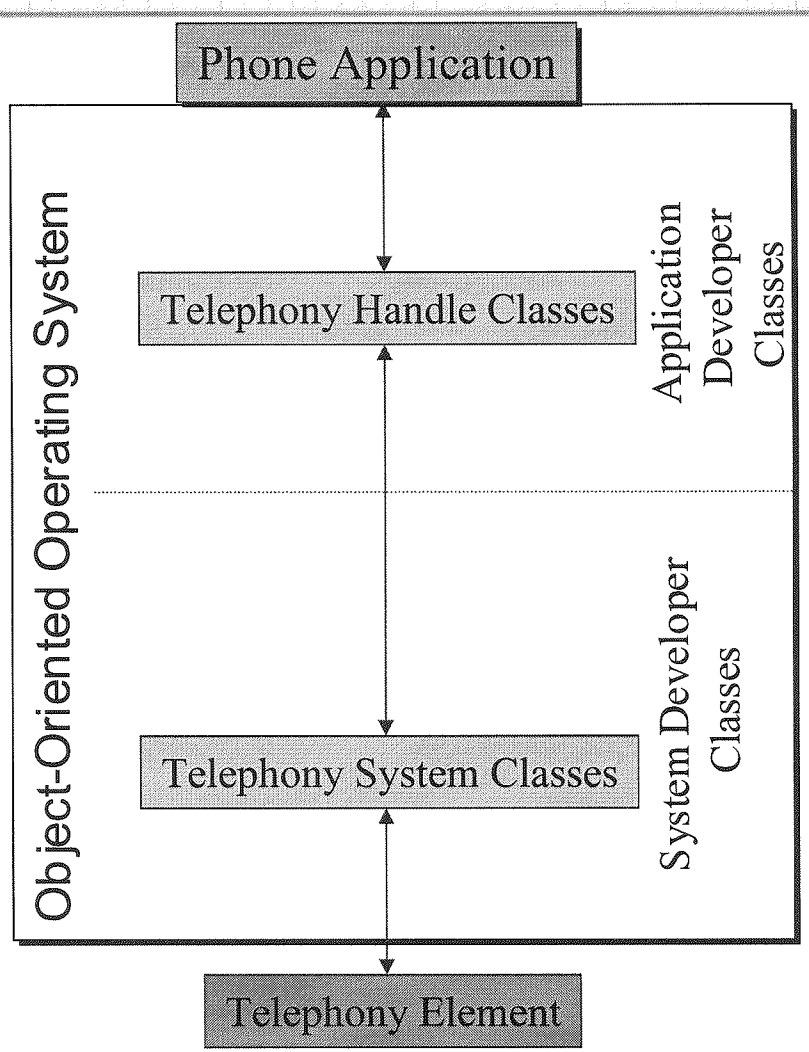
Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Invention Timeline

- 1988
  - Launch of Apple "Pink" OS Project
- 1989
  - "Cathedral" Project - Digital Signal Processing Project on Blue
  - Simultaneous Speech, Telephony, Graphics Rendering
- 1990
  - First Internal Implementation of Object-Oriented Telephony (Nov.)
- 1992
  - Launch of Taligent with Support from IBM (March)
- 1993
  - Final Design Review for Telephony Classes (June)
  - Patent Application Filed (October)
- 1995
  - CommonPoint 1.0 Reference Release (June)
  - USPTO Interview with Prosecuting Attorney (June)
  - Patent Issued (October)

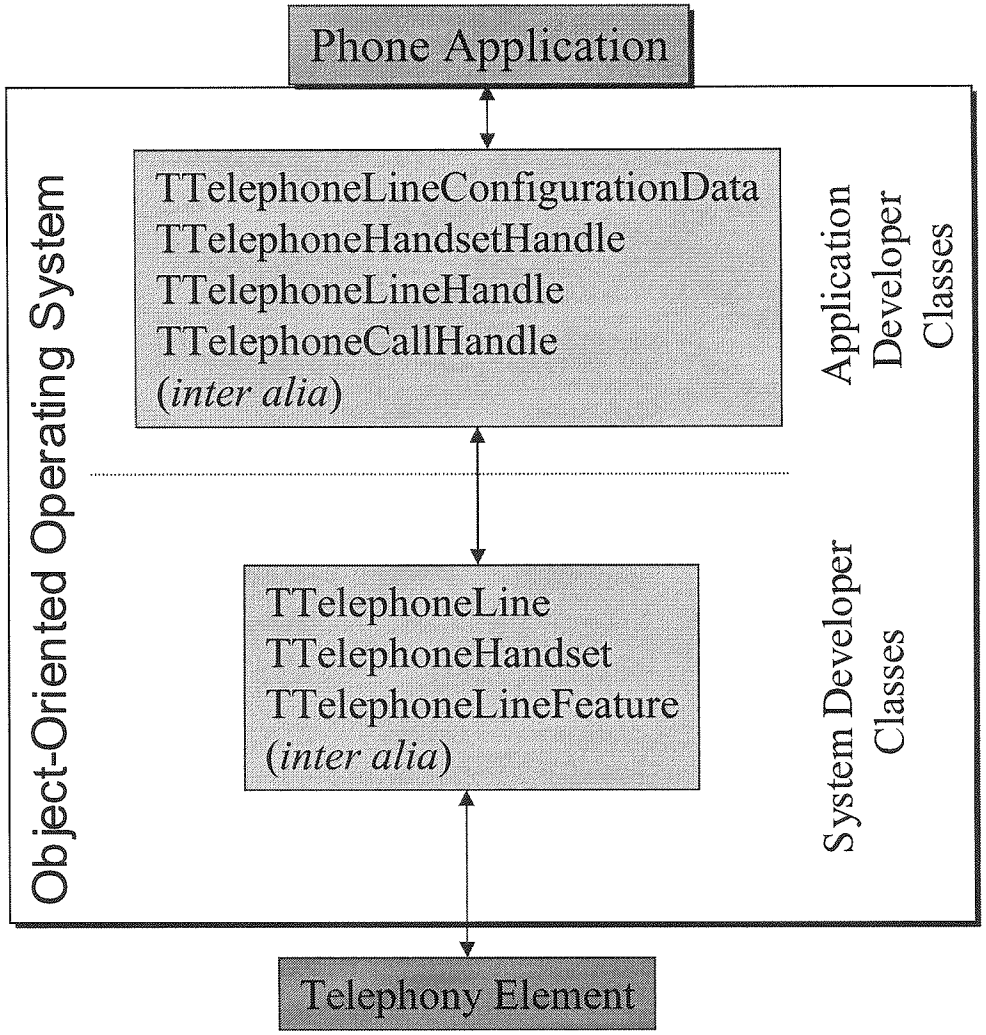
Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

'854 Patent Technical Overview



Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Example Classes in '854 Patent



Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Specification Support for Claim Term "Object-Oriented Operating System"

"A prior art approach is to layer objects and class libraries in a procedural environment. Many application frameworks on the market take this design approach. In this design, there are one or more object layers on top of a monolithic operating system.

...

"[D]ifficulties arise from the fact that while it is easy for a developer to reuse their own objects, it is difficult to use objects from other system and the developer still needs to reach into the lower non-object layers with procedural Operating System (OS) calls."

(Col. 5, l. 62 – Col. 6, l. 7)

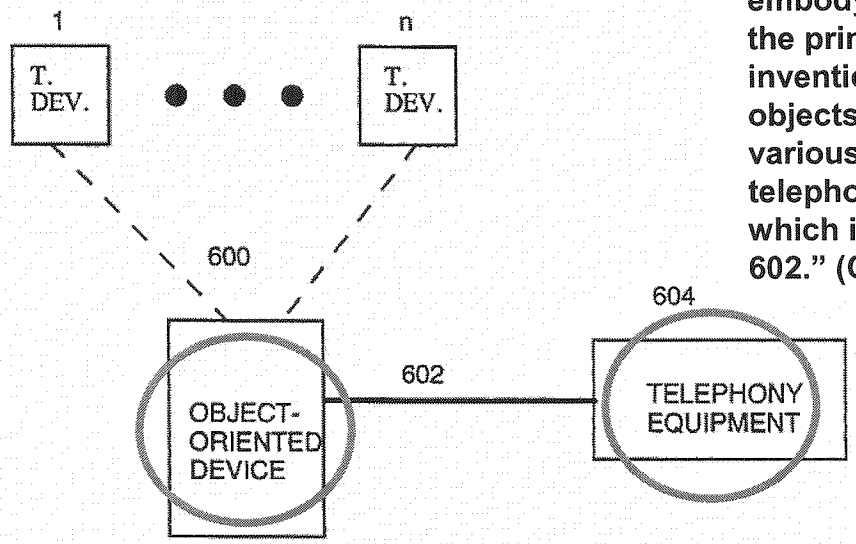


Figure 6

"FIG. 6 demonstrates a device embodying at least some of the principles of the present invention. Device 600 utilizes objects for interfacing with various elements of the telephony system 604 to which it is connected via line 602." (Col. 12, ll. 5-9)

"An object-oriented operating system is an essential element of computer-based telephony applications in a preferred embodiment." (Col. 3, ll. 7-9).

NeXTSTEP I and II Do Not Teach Or Suggest An Object Oriented Operating System

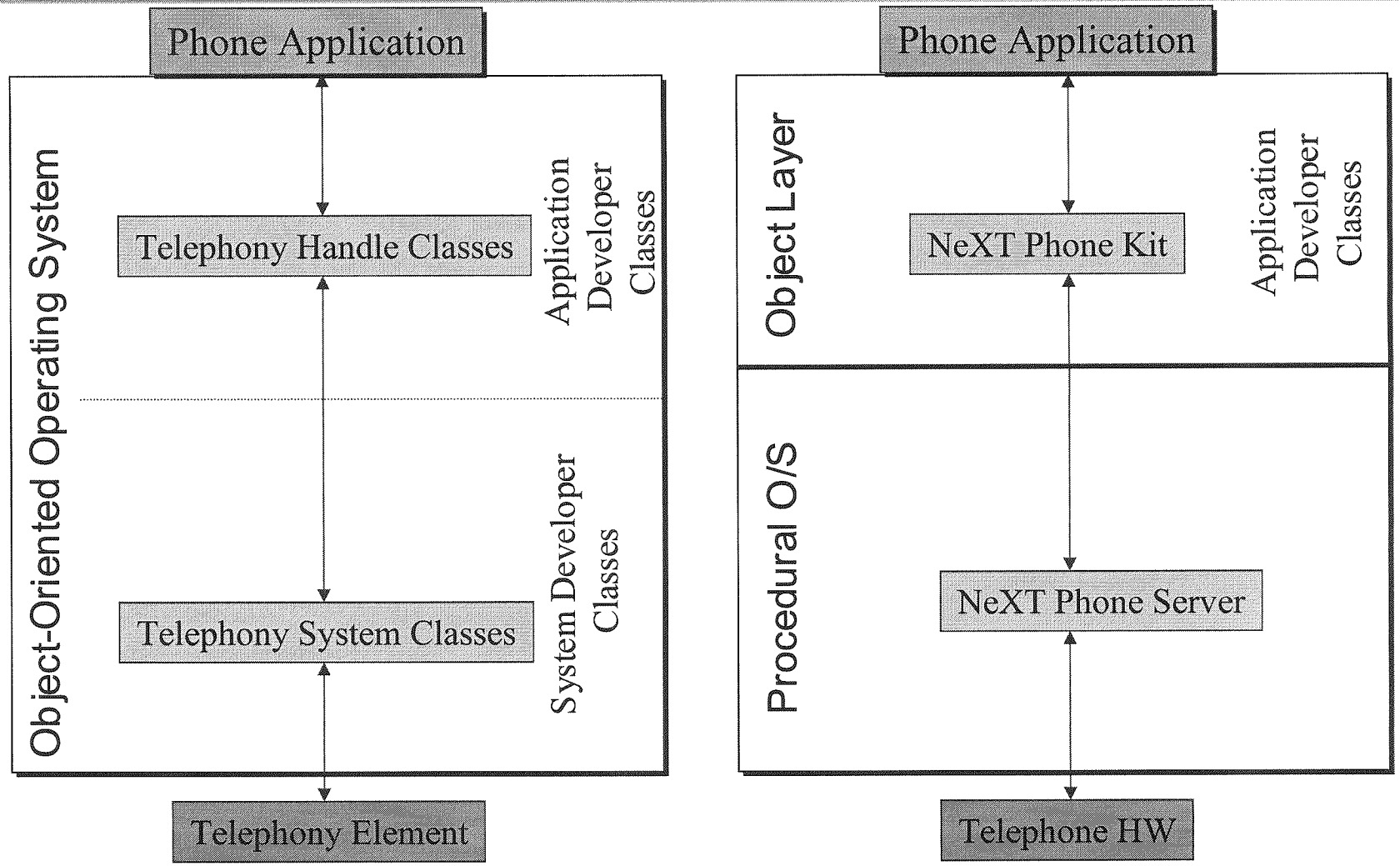
Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Deficiencies in NeXTSTEP I and II

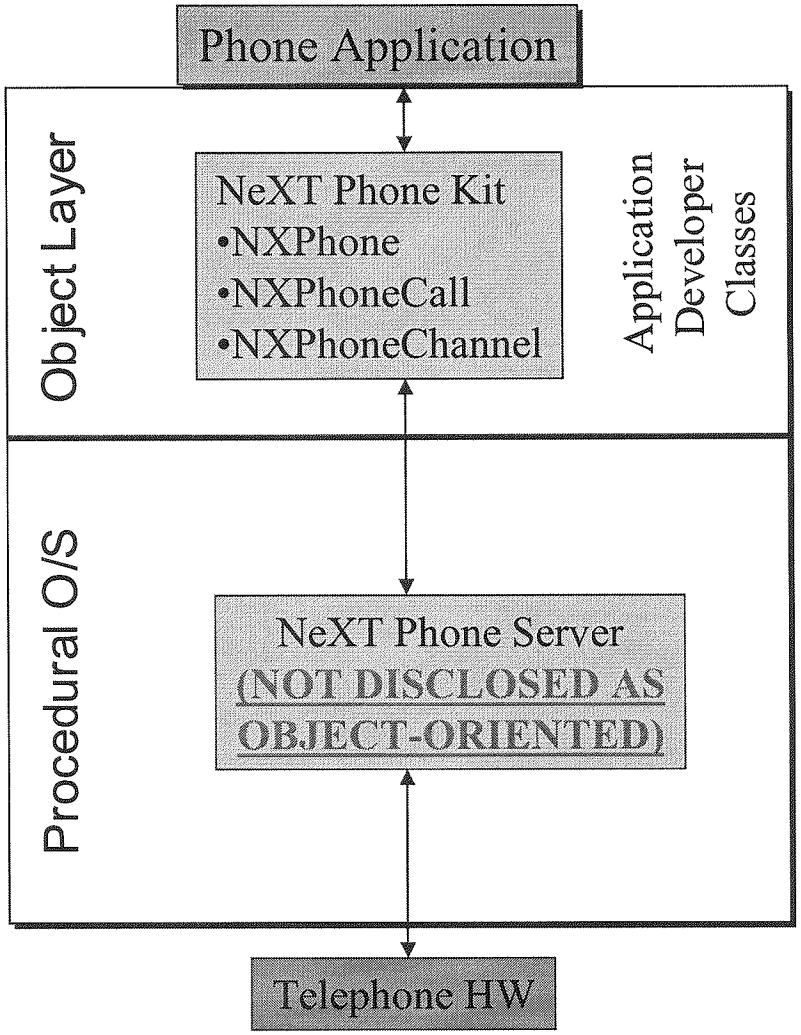
- NeXTSTEP I and II fail to provide:
  - an "object-oriented operating system"
  - control of telephony elements by an object-oriented operating system
  - enabling disclosure to implement a Phone Server
  - storing status information in the data of the telephony object

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

'854 vs. NeXTSTEP I and II



NeXTSTEP Telephone Management Software



The NeXTSTEP Phone Kit

- The Phone Kit provides Objective C classes for use by an application developer. These classes are used to instantiate objects **within the application**.

Phone Kit Classes

The Phone Kit consists of just three Objective C classes, all direct subclasses of the Object class:

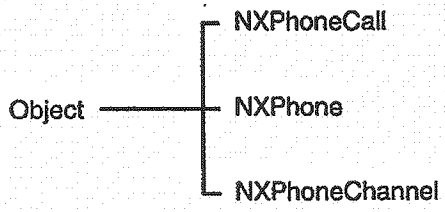


Figure 13-4. Phone Kit Inheritance Hierarchy

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

The NeXTSTEP Operating System

- NeXTSTEP provides object-oriented capability by **layering** objects and class libraries in a **procedural environment**
- This would not be understood to be an "object-oriented **operating system**" by one skilled in the art (see, Col. 5, ll. 62-64)
- The '854 patent distinguishes the "layering" approach of simply "layer[ing] objects and class libraries in a procedural environment" (Col. 5, ll. 62-64).

"[D]ifficulties arise from the fact that while it is easy for a developer to reuse their own objects, it is difficult to use objects from other systems and the developer still needs to reach into the lower non-object layers with procedural Operating System (OS) calls." (Col. 6, ll. 3-7)

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

The NeXTSTEP Phone Server

- "The Phone Kit is ***just one half*** of NeXTSTEP telephone-management software." (NeXTSTEP II, p. 13-6)
- The only operating system component described in NeXTSTEP is the Phone Server.
- Issues:
  - The NeXTSTEP documents suggest that the Phone Server is not object-oriented.
  - Little description of the Phone Server is provided at all

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

NeXTSTEP Documents Suggest the Phone Server is NOT Object-Oriented

- "The Phone Kit is an object-oriented **interface to** the Phone Server, in much the same way that the Application Kit is an object-oriented **interface to** the Window Server." (NeXTSTEP II, p. 13-6)
- The Window Server is ***procedural***

Little Description Regarding the Phone Server

The Phone Server and Phone Kit

The Phone Kit is just one half of NeXTSTEP telephone-management software. The other half is the Phone Server, which monitors the phone line and understands the POTS and ISDN signaling protocols. The Server is an independent process that conveys information to and from the phone line at the behest of client applications

The Phone Server controls input and output on a telephone line much as the Window Server controls input from the keyboard and mouse and output to the screen. Just as the Window Server is able to translate user actions into events and turn PostScript code into visible images, the Phone Server is able to translate information received over the phone line into a form that's useful for applications, and it can take information produced by applications and send it over the line. Applications "talk to" the telephone network through the Phone Server.

The Phone Kit is an object-oriented interface to the Phone Server, in much the same way that the Application Kit is an object-oriented interface to the Window Server. The Phone Kit provides a framework for delivering instructions to the Phone Server and for receiving notifications from the Server of activity on the phone line. Communication between the Kit and the Server is through the mechanism of distributed objects and remote Objective C messages.

See NeXTSTEP II at 13-6.

Little Description Regarding the Phone Server

Monitoring the Connection

Phone calls involve the give and take of information. To get information from the Phone Server (and over the phone line), an application must be listening for remote input from that source. Applications that display a user interface and respond to events listen for phone input just as they listen for other remote input—between events. Sending the NXPhone object a **runFromAppKit** message adds a dedicated port for the Phone Server to the list of sources that are checked whenever the application is finished with its

Phone Server Messages

All communication between an application and the Phone Server is asynchronous. When a Kit method calls upon the Server to do something—for example to dial a number or transmit some data over the phone line—it doesn't wait for the Server to finish; it returns immediately. Similarly, when the Server sends a message to the application, it doesn't wait for the application to respond.

An asynchronous message can have no valid return. As soon as the message is dispatched to the receiving application, it returns in the sending application. The sender can't get the results of any processing the receiver does, and nothing the receiver returns will ever get back to the sender.

This fact is indicated in the method descriptions by a **void** return type. For example:

– (void)**remotePickup**

A **void** return is a fairly accurate indication of a method that communicates with the Phone Server. In some cases, **void** marks methods that you must implement to respond to remote messages from the Server. In other cases, **void** methods are ones you invoke to send a remote message to the Server.

See NeXTSTEP II at 13-8 and 13-9.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Limited Disclosure of the Phone Server

- The limited description of the Phone Server provides no explanation for:
  - How to implement a Phone Server for generic hardware (e.g., POTS (plain old telephone service) and ISDN)
  - How to customize a Phone Server for custom hardware (e.g., PBX)
- NeXTSTEP I and II do not inform a person of ordinary skill in the art how to make and use a Phone Server

NeXTSTEP I and II Do Not Teach or Suggest Storing Status Information in a Telephony Object

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Storing Status Information ('854 Patent)

- The '854 Patent provides for status information in a telephony object (see, e.g., TTelephoneStatusNotification, TTelephoneRingNotification, and TTelephoneDigitsNotification) which provide "[i]nformation regarding the state of lines, calls, and features." (Col. 14, ll. 30-44)
- For example, TNotification can provide an application with a TTelephoneStatusNotification, which allows the application to be notified of a telephony event (i.e., status) associated with a line
- Can ask TTelephoneCallHandle for the status ***independent*** of the notifications (stores data from notifications for later querying)

Storing Status Information (NeXTSTEP)

- NeXTSTEP does not teach or suggest any telephony object capable of "storing status information".
- In particular, the three derived classes outlined in the NeXTSTEP references (NXPhone, NXPhoneChannel, NXPhoneCall) have no instance variables that could be used to store status information in the data of the telephony object.
- Any data stored in the Phone Server would not be in a telephony ***object***.

NXPhone

Instance Variables

None declared in this class.

NXPhoneChannel

Instance Variables

None declared in this class.

NXPhoneCall

Instance Variables

None declared in this class.

See NeXTSTEP II at 13-16, 13-22, and 13-32.

Discussion of the Office Action October 28, 2010

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Status Overview

- Claims 1-24 under *Ex Parte* Reexamination
- '854 patent contains two Independent Claims (1 and 13)
- Office Action rejected claims 1-24 under 35 U.S.C. §103(a) as being unpatentable over NeXTSTEP I in view of NeXTSTEP II

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Claims 1 and 13

Claim 1. A telephony apparatus, comprising:

(a) a processor;

(b) a storage attached to and controlled by the processor;

(c) **an object oriented operating system**, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage and controlling operations of the processor;

(d) **a display attached to the processor under the control of the object oriented operating system**;

(e) a telephony element attached to the processor;

(f) a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element **under the control of the object-oriented operating system**, stored in the storage and displayed on the display; and

(g) means for **controlling the telephony element by the object oriented operating system utilizing the logic in the telephony object to interface the telephony element to the processor** by initiating a call connection, monitoring call progress, activating call featurs and **storing status information in the data of the telephony object.**

Claim 13. A method for enabling telephony elements on a computer system, including a processor with an attached storage, display and telephony element, comprising the steps of:

(a) controlling operations of the processor with **an object oriented operating system**, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage;

(b) creating a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element **under the control of the object-oriented operating system**, stored in the storage and displayed on the display; and

(c) **controlling the telephony element by the object-oriented operating system utilizing logic in the telephony object to interface the telephony element to the processor** by initiating a call connection, monitoring call progress, activating call features and **storing status information in the data of the telephony object.**

Reasons Why Claims Should be Confirmed

- NeXTSTEP I and II do not teach or suggest:
  - an "object-oriented operating system"
  - elements (c), (d), (f), and (g) of claim 1 and steps (a), (b), and (c) of claim 13
  - storing status information in the data of the telephony object

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

1. NeXTSTEP Has a Procedural Operating System

Claim 1. A telephony apparatus, comprising:

(a) a processor;

(b) a storage attached to and controlled by the processor;

(c) **an object oriented operating system**, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage and controlling operations of the processor;

(d) a display attached to the processor under the control of the object oriented operating system;

(e) a telephony element attached to the processor;

(f) a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element under the control of the object-oriented operating system, stored in the storage and displayed on the display; and

(g) means for controlling the telephony element by the object oriented operating system utilizing the logic in the telephony object to interface the telephony element to the processor by initiating a call connection, monitoring call progress, activating call featurs and storing status information in the data of the telephony object.

Claim 13. A method for enabling telephony elements on a computer system, including a processor with an attached storage, display and telephony element, comprising the steps of:

(a) controlling operations of the processor with **an object oriented operating system**, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage;

(b) creating a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element under the control of the object-oriented operating system, stored in the storage and displayed on the display; and

(c) controlling the telephony element by the object-oriented operating system utilizing logic in the telephony object to interface the telephony element to the processor by initiating a call connection, monitoring call progress, activating call features and storing status information in the data of the telephony object.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

1. NeXTSTEP Has a Procedural Operating System

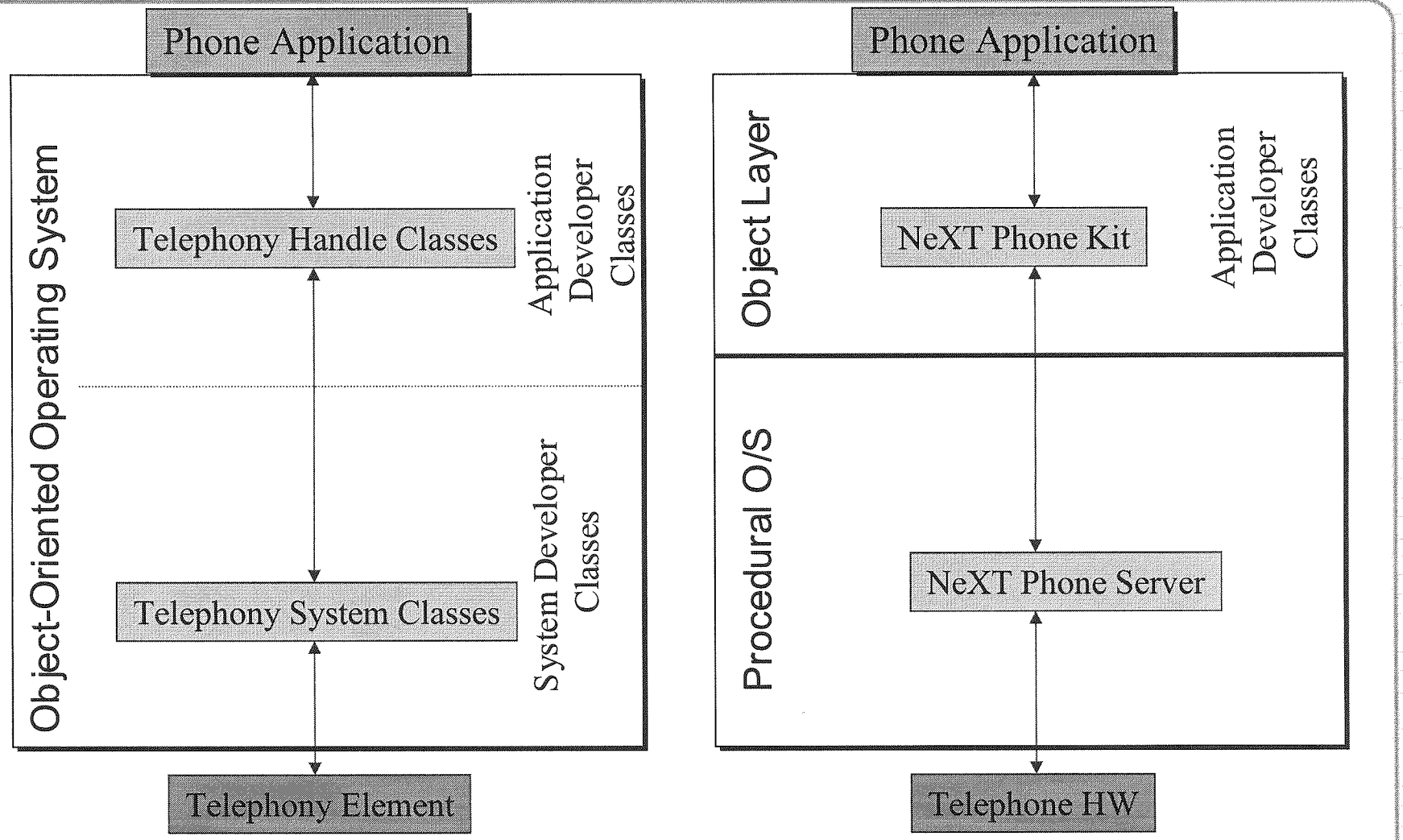
- The Office Action does not distinguish between an "Object Oriented Operating System" (as recited in the claims) and other forms of object-oriented software.
  - "Nextstep is object-oriented software. See Title Page." (Office Action, p. 3)

- NeXTSTEP is just object oriented ***programming***, not an object-oriented operating system. This distinction is necessary in order to give the broadest ***reasonable*** interpretation to the claim terms.

- "Mr. Stephens agreed to further amend claims 1 and 14 in order to further **distinguish** between the **claimed object oriented operating system** and **object oriented programming**." (Interview Summary of June 27, 1995)

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

1. NeXTSTEP Has a Procedural Operating System



Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

1. NeXTSTEP Has a Procedural Operating System

- '854 patent supports an "object oriented operating system" as ***something other than*** an approach to "layer objects and class libraries in a procedural environment" (Col. 5, ll. 62-64)
- The NeXTSTEP Phone Kit layers objects and class libraries on top of a procedural operating system (Col. 5, ll. 62-66)
- Therefore, the NeXTSTEP I and II documents do <u>not</u> teach or suggest the required "telephony apparatus" including an "object oriented operating system"

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

2. NeXTSTEP Does Not Teach or Suggest Control of Telephony Element by O-O Operating System Using a Telephony Object

Claim 1. A telephony apparatus, comprising:

(a) a processor;

(b) a storage attached to and controlled by the processor;

(c) an object oriented operating system, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage and controlling operations of the processor;

(d) **a display attached to the processor under the control of the object oriented operating system**;

(e) a telephony element attached to the processor;

(f) a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element **under the control of the object-oriented operating system**, stored in the storage and displayed on the display; and

(g) means for **controlling the telephony element by the object oriented operating system utilizing the logic in the telephony object to interface the telephony element to the processor** by initiating a call connection, monitoring call progress, activating call featurs and storing status information in the data of the telephony object.

Claim 13. A method for enabling telephony elements on a computer system, including a processor with an attached storage, display and telephony element, comprising the steps of:

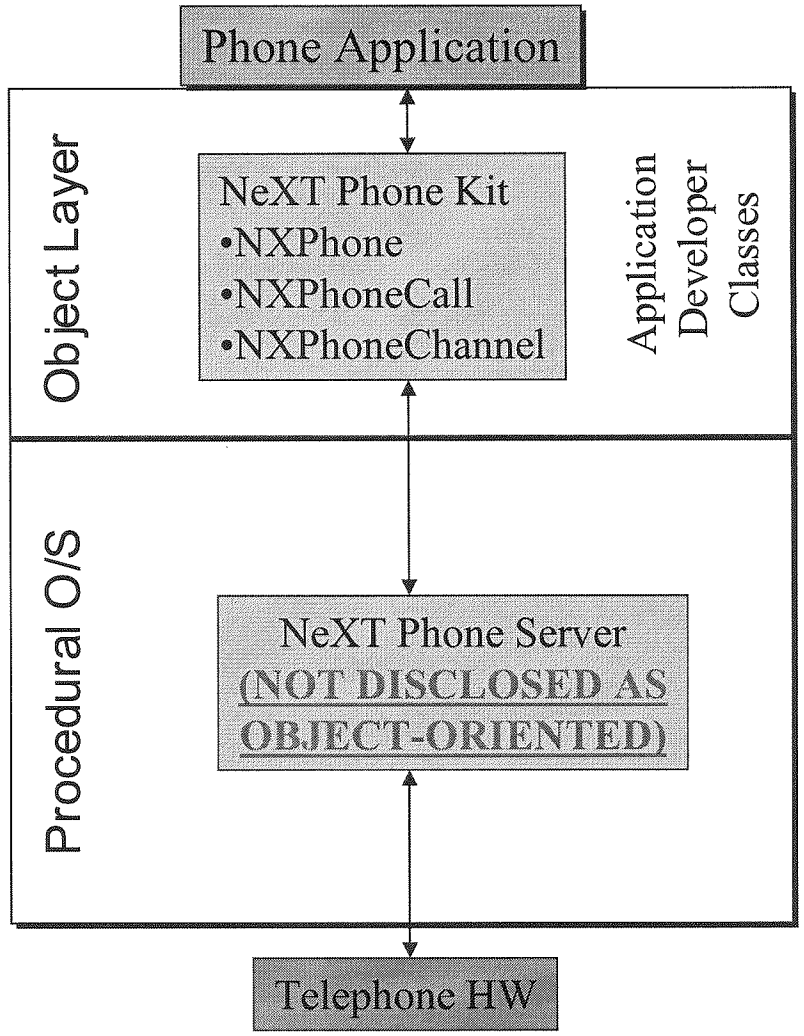
(a) controlling operations of the processor with an object oriented operating system, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage;

(b) creating a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element **under the control of the object-oriented operating system**, stored in the storage and displayed on the display; and

(c) **controlling the telephony element by the object-oriented operating system utilizing logic in the telephony object to interface the telephony element to the processor** by initiating a call connection, monitoring call progress, activating call features and storing status information in the data of the telephony object.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

2. NeXTSTEP Does Not Teach or Suggest Control of Telephony Element by O-O Operating System Using a Telephony Object



Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

2. NeXTSTEP Does Not Teach or Suggest Control of Telephony Element by O-O Operating System Using a Telephony Object

- NeXTSTEP fails to teach or suggest:
  - 1. (d) a display attached to the processor under the control of the object oriented operating system
- The Office Action cites the following support:
  - "Computers are inherently attached to a display and a processor. Further, a computer with an object oriented operating system, is inherently under the control of the object-oriented operating system." (Office Action, p. 4)
- The limited disclosure in the NeXTSTEP documents fails to teach a mechanism by which to control a display by an object-oriented operating system.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

2. NeXTSTEP Does Not Teach or Suggest Control of Telephony Element by O-O Operating System Using a Telephony Object

- NeXTSTEP fails to teach or suggest:
  - 1. (f) a telephony object … representative of the telephony element under the control of the object-oriented operating system …
- The Office Action cites the following support:
  - "An NXPhone object corresponds to a telephone line that's connected to the user's computer." (Office Action, p. 4 (citing NextStepII, pg. 13-8))
- The limited disclosure in the NeXTSTEP documents fails to teach a mechanism by which to control the telephony element by an object-oriented operating system.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

2. NeXTSTEP Does Not Teach or Suggest Control of Telephony Element by O-O Operating System Using a Telephony Object

- NeXTSTEP fails to teach or suggest:
  - 1. (g) means for controlling the telephony element by the object oriented operating system utilizing the logic in the telephony object to interface the telephony element to the processor …
- The Office Action cites the following support:
  - Re: "controlling the telephony element by the object oriented operating system …"
    - "the computer is the means for controlling" (Office Action, pp. 5 and 9)
  - Re: "… utilizing the logic in the telephony object to interface the telephony element to the processor"
    - "An NXPhone object corresponds to a telephone line that's connected to the user's computer." (Office Action, pp. 5 and 9 (citing NextStepII, pp. 13-16))
- The limited disclosure in the NeXTSTEP documents fails to teach a mechanism by which to control the telephony element by an object-oriented operating system.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

3. NeXTSTEP Does Not Teach or Suggest Storing Status Information in the Data of the Telephony Object

Claim 1. A telephony apparatus, comprising:

(a) a processor;

(b) a storage attached to and controlled by the processor;

(c) an object oriented operating system, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage and controlling operations of the processor;

(d) a display attached to the processor under the control of the object oriented operating system;

(e) a telephony element attached to the processor;

(f) a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element under the control of the object-oriented operating system, stored in the storage and displayed on the display; and

(g) means for controlling the telephony element by the object oriented operating system utilizing the logic in the telephony object to interface the telephony element to the processor by initiating a call connection, monitoring call progress, activating call featurs and **storing status information in the data of the telephony object.**

Claim 13. A method for enabling telephony elements on a computer system, including a processor with an attached storage, display and telephony element, comprising the steps of:

(a) controlling operations of the processor with an object oriented operating system, supporting encapsulation, polymorphism and inheritance, including objects, each of the objects containing logic and data resident in the storage;

(b) creating a telephony object, including logic for interfacing the telephony element to the processor and data for storing status information associated with the telephony element in the telephony object, and representative of the telephony element under the control of the object-oriented operating system, stored in the storage and displayed on the display; and

(c) controlling the telephony element by the object-oriented operating system utilizing logic in the telephony object to interface the telephony element to the processor by initiating a call connection, monitoring call progress, activating call features and **storing status information in the data of the telephony object.**

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

3. NeXTSTEP Does Not Teach or Suggest Storing Status Information in the Data of the Telephony Object

- NeXTSTEP fails to teach or suggest, "storing status information in the data of the telephony object"
- The Office Action cites to the following:
  - "To begin using the Phone Kit, an application first creates an NXPhone instance and an NXPhoneChannel for each channel that will be used. The NXPhone object must be informed about the channels." Further, see the instance variables of NextStepII, pg. 13-8.
- This functionality informs the NXPhone object of channels being utilized, but does not ***store*** status information.
- Additionally, as previously noted, none of the three classes in the NeXTSTEP Phone Kit have any instance variables declared in the class itself, and therefore any status information would not be stored ***in the data of the telephony object***.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Conclusion

- NeXTSTEP I and II do not teach or suggest:
  - an "object oriented operating system"
  - elements (c), (d), (f), and (g) of claim 1 and steps (a), (b), and (c) of claim 13
  - storing status information in the telephony object

Questions and Comments