Exhibit 17

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Apple Inc.

Ex Parte Reexamination 90/010,967

U.S. Patent No. 5,315,703 to Matheny *et al.*

February 8, 2011

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Agenda

I. Introductions

II. Status of the Litigation

III. Overview of Patentability

IV. Story of the Invention

V. Technical Overview of '703 Patent

VI. Overview of the Cohen Reference (Gypsy)

VII. Discussion of the Office Action

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Introductions

Technical Expert:

- David A. Wilson, Ph.D.

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

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Status of the Litigation

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

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

The '703 Patent is in concurrent litigation (*Nokia v. Apple*, 1:09-cv-00791 (D. Del.)) before Chief Judge Sleet

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Apple's Business Model

Apple has adopted 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

Issued Claim 8

8. A method for implementing an object-oriented notification framework system, comprising the steps of:

connecting a plurality of objects to a notification source;

storing connection information for the plurality of objects in a connection object of an object-oriented operating system;

registering connection information, including registration information indicative of a notification status, in the connection object of the object-oriented operating system;

selectively dispatching notification to at least one of the plurality of objects based on the connection registration information stored in the connection object of the object-oriented operating system; and

receiving the notification by the at least one of the plurality of objects and taking action based on the notification.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

All Claims Should Be Confirmed

Cohen does not anticipate or render obvious claims 1 and 8 because it does not teach, suggest, or disclose:

1. a **notification receiver object** (e.g. "receiving the notification by the at least one of the ***plurality of objects***"); and

2. a "**connection object**".

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Exemplary Use Case Comparison

'703 Patent Example

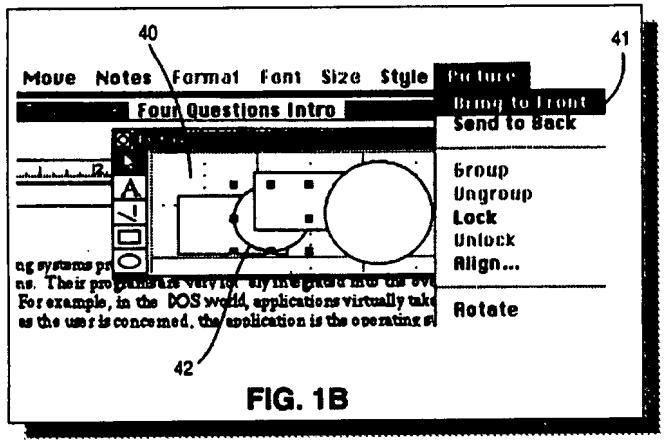
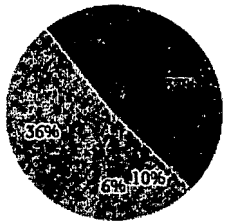


FIG. 1B

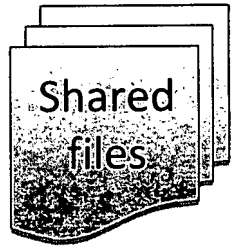
Category	Amount
Home	$ (872.40)
Food	$ (226.00)
Gas	$ (137.50)
Credit Card	$ (850.00)
Entertainment	$ (245.00)
Total	$ (2,330.90)



Object-Oriented Operating System

Cohen

"The Gypsy Programming Support Environment provides support for a team of developers to produce and maintain systems built from multiple components." (Cohen, p. 201)



Gypsy

UNIX Operating System

Story of the Invention

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 Apple's "System 7" operating system, which remained a procedural operating system
- "Pink" was designed and built to be an object-oriented operating system implemented in C++
- The "Pink" development team was spun-off into Taligent

Object-Oriented Programming Benefits

- Managing complexity
- Reusing code
- Simpler program development

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

David R. Anderson (Co-Inventor)

- Education:
  - Purdue University: B.S.E.E. (1980)
- Relevant Employment:
  - Apple/Taligent:
    - Senior Software Engineer (1989-1992)
    - Software Development Manager (1992-1995)
    - Developed applications to run on top of the "Pink" OS
    - Designed and built the "Pink" **notification system**
- Named inventor on over 30 U.S. patents, approximately half of which resulted from work done at Apple/Taligent

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Story of the Invention

- As part of the Pink project, Apple/Taligent sought to create an advanced event notification system
- Goal: to design and build a system with class hierarchies, **objects**, and other elements that could provide distinct advantages in an event notification system
- Leverage the aforementioned benefits of object-oriented programming

Problems with Existing Technology

Existing technology:

- Could not easily integrate notifications with the increasingly complex applications being developed
- Could not support the more pervasive use of notifications desired by application developers

Led to performance problems – notifications not sufficiently scaleable or reusable in different problem domains

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Benefits of the Invention

- Apple/Taligent invented a new event notification system based on interactions among three types of elements:
  - –notification sources
  - –notification receiver objects
  - –connection objects

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

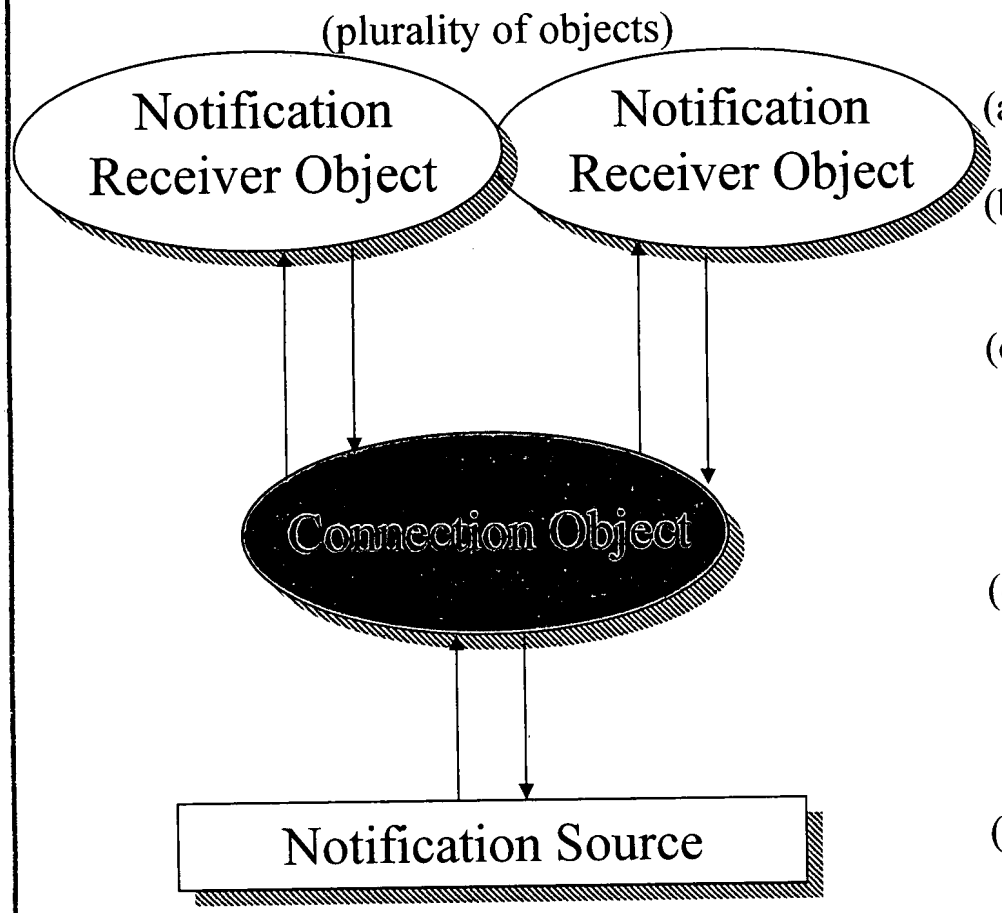
Benefits of the Invention

- Use of the three-element system allows for ***scaleable***, efficient, and ***flexible*** application development and notification distribution.
- Once designed and built, the notification system was ***pervasively integrated*** with the Pink user interface.
- Development of the new notification system was regarded internally as a significant technical accomplishment and one of the most important aspects of Pink.

Technical Overview of the '703 Patent

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

'703 Patent Technical Overview



8. A method for implementing an object-oriented notification framework system, comprising the steps of:

(a) connecting a plurality of objects to a notification source;

(b) storing connection information for the plurality of objects in a connection object of an object-oriented operating system;

(c) registering connection information, including registration information indicative of a notification status, in the connection object of the object-oriented operating system;

(d) selectively dispatching notification to at least one of the plurality of objects based on the connection registration information stored in the connection object of the object-oriented operating system; and

(e) receiving the notification by the at least one of the plurality of objects and taking action based on the notification.

Overview of the Cohen Reference (Gypsy)

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Cohen's Narrow Problem Domain

'703 Patent Example

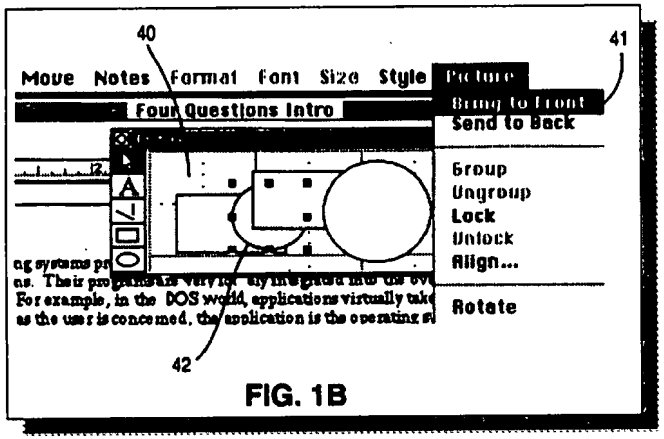
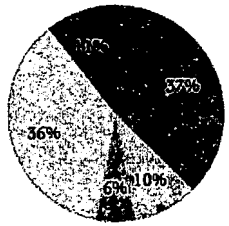


FIG. 1B

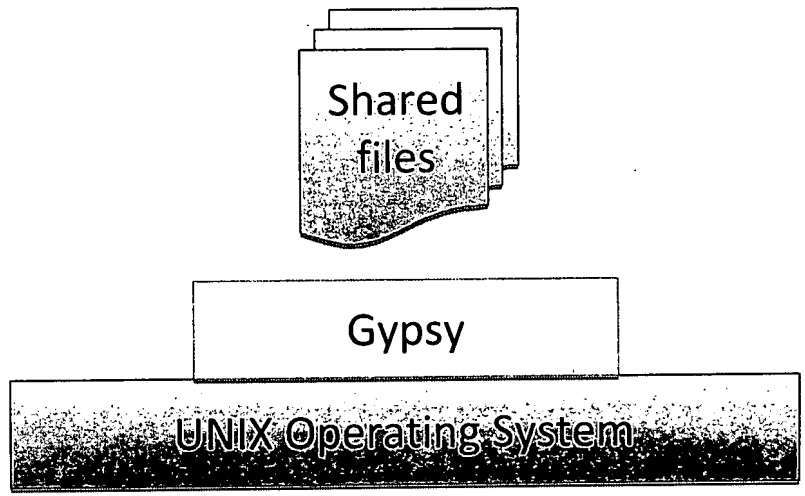
Category	Amount
Home	$ (872.40)
Food	$ (226.00)
Gas	$ (137.50)
Credit Card	$ (850.00)
Entertainment	$ (245.00)
Total	$ (2,330.90)



Object-Oriented Operating System

Cohen

"The Gypsy Programming Support Environment provides support for a team of developers to produce and maintain systems built from multiple components." (Cohen, p. 201)

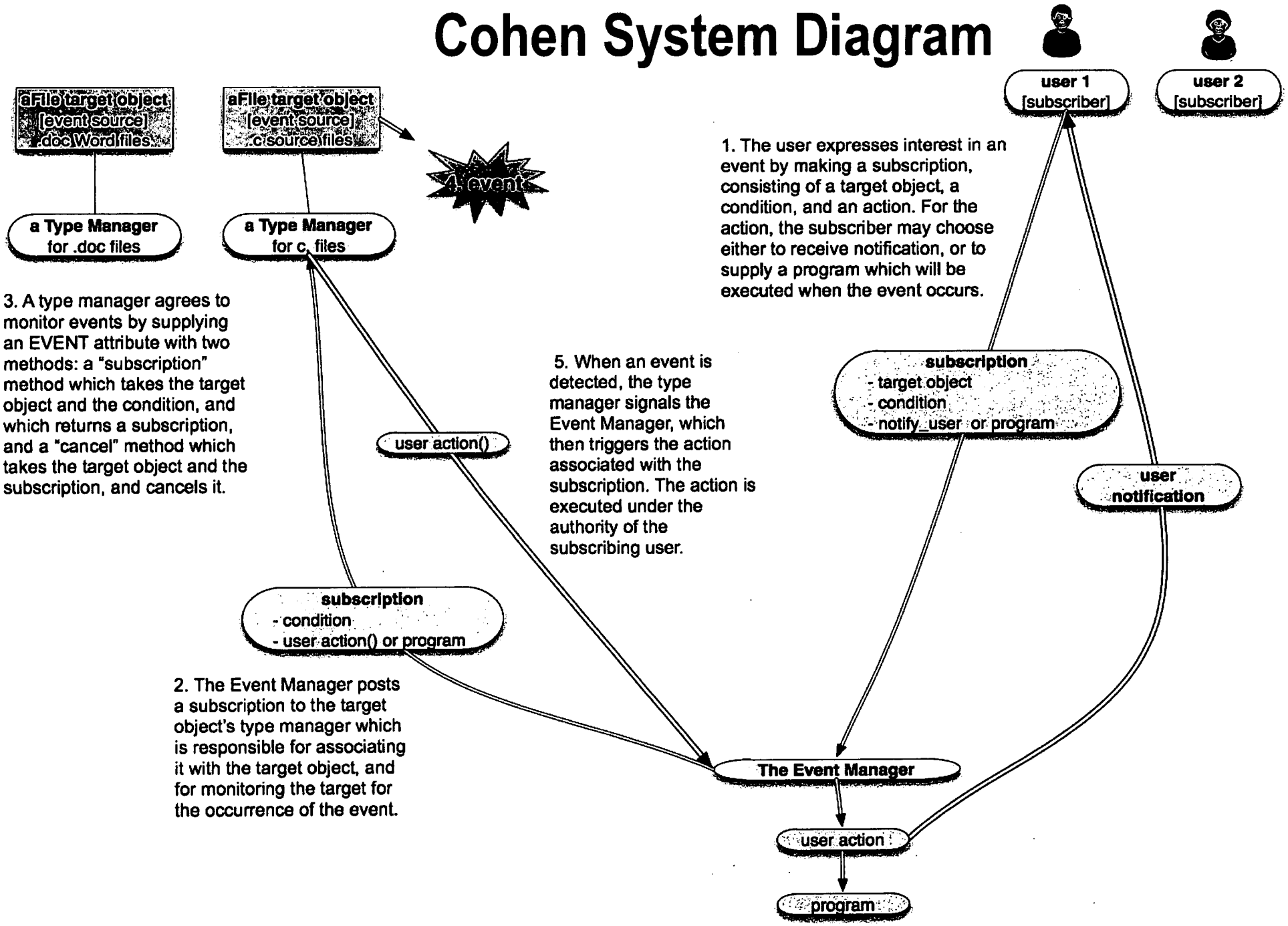


Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Cohen's Narrow Problem Domain

- Cohen was a narrowly-tailored solution to a very specific problem
  - "The Gypsy Programming Support Environment provides support for a team of developers to produce and maintain systems built from multiple components." (Cohen, p. 201)
  - requires a new type manager for each data type ("***each*** type manager determines the sort of events it will monitor and interprets the condition accordingly") (Cohen, p. 211).
  - requires modification for each problem domain

Cohen System Diagram



Discussion of the Office Action

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Rejections At Issue

- Independent claims 1 and 8 are rejected over Cohen
  - Claim 8 under § 102(b)
  - Claim 1 under § 103(a) (single reference)
- Several dependent claims rejected under § 103(a) over Cohen in view of Bernstein
- Rejections discussed in context of claim 8

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

All Claims Should Be Confirmed

Cohen does not anticipate or render obvious claims 1 and 8 because it does not teach, suggest, or disclose:

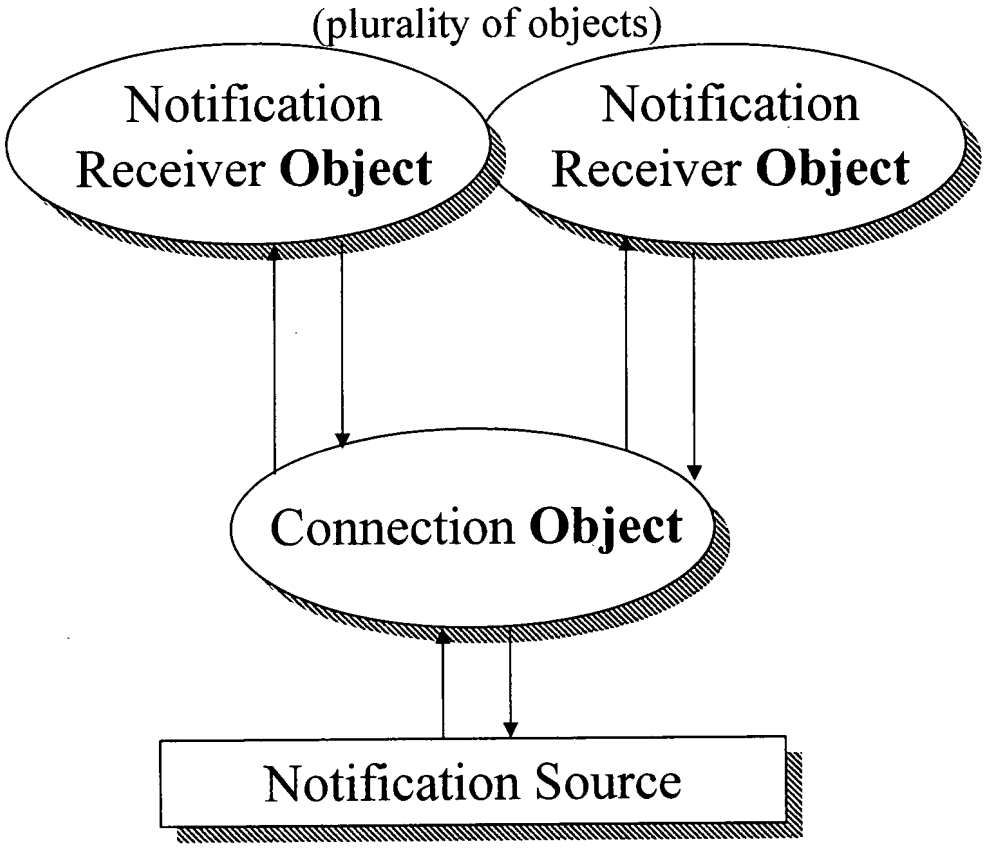
1. a **notification receiver object** (e.g. "receiving the notification by the at least one of the ***plurality of objects***"); and

2. a "**connection object**".

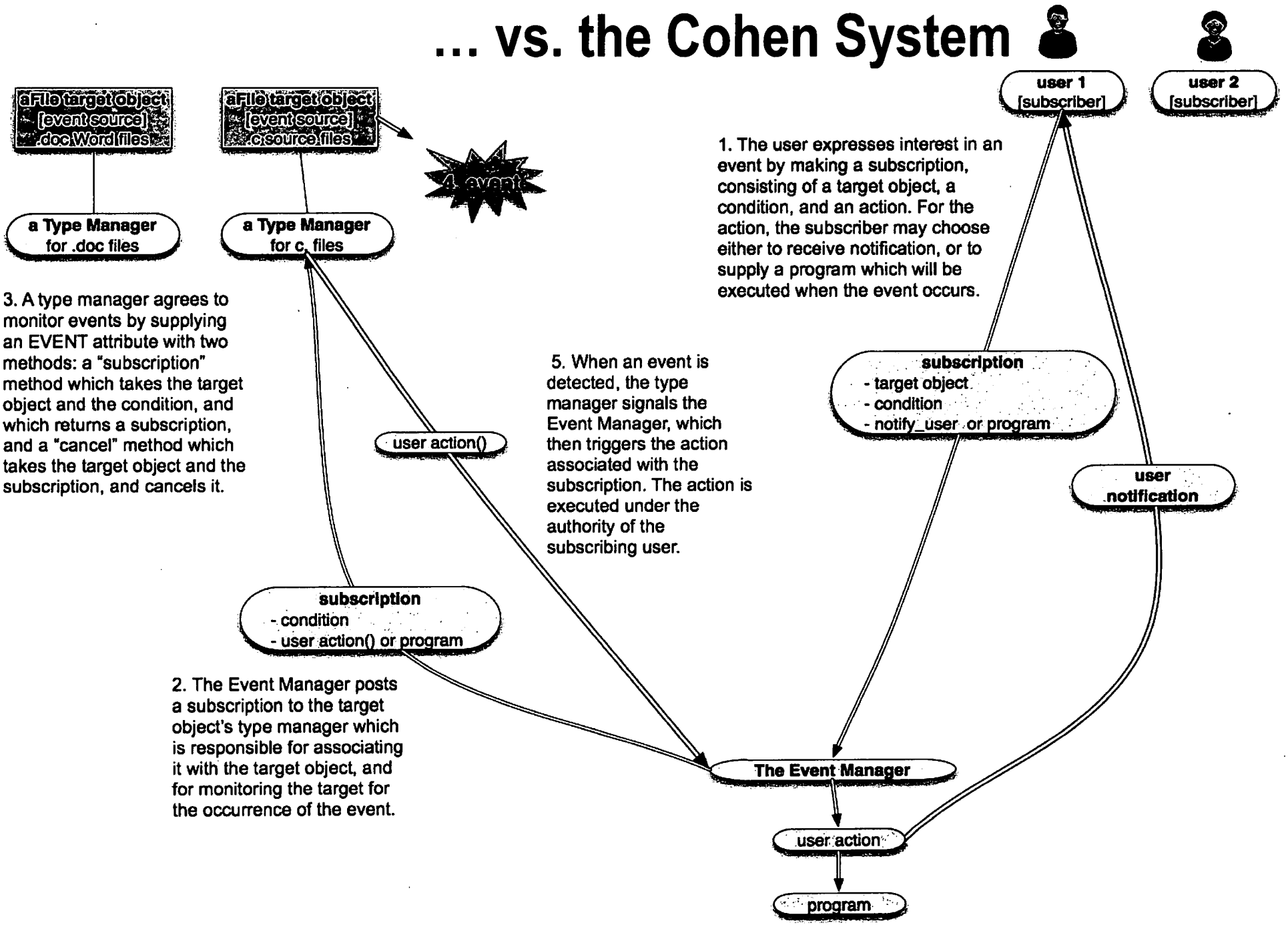
Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

The '703 Patent

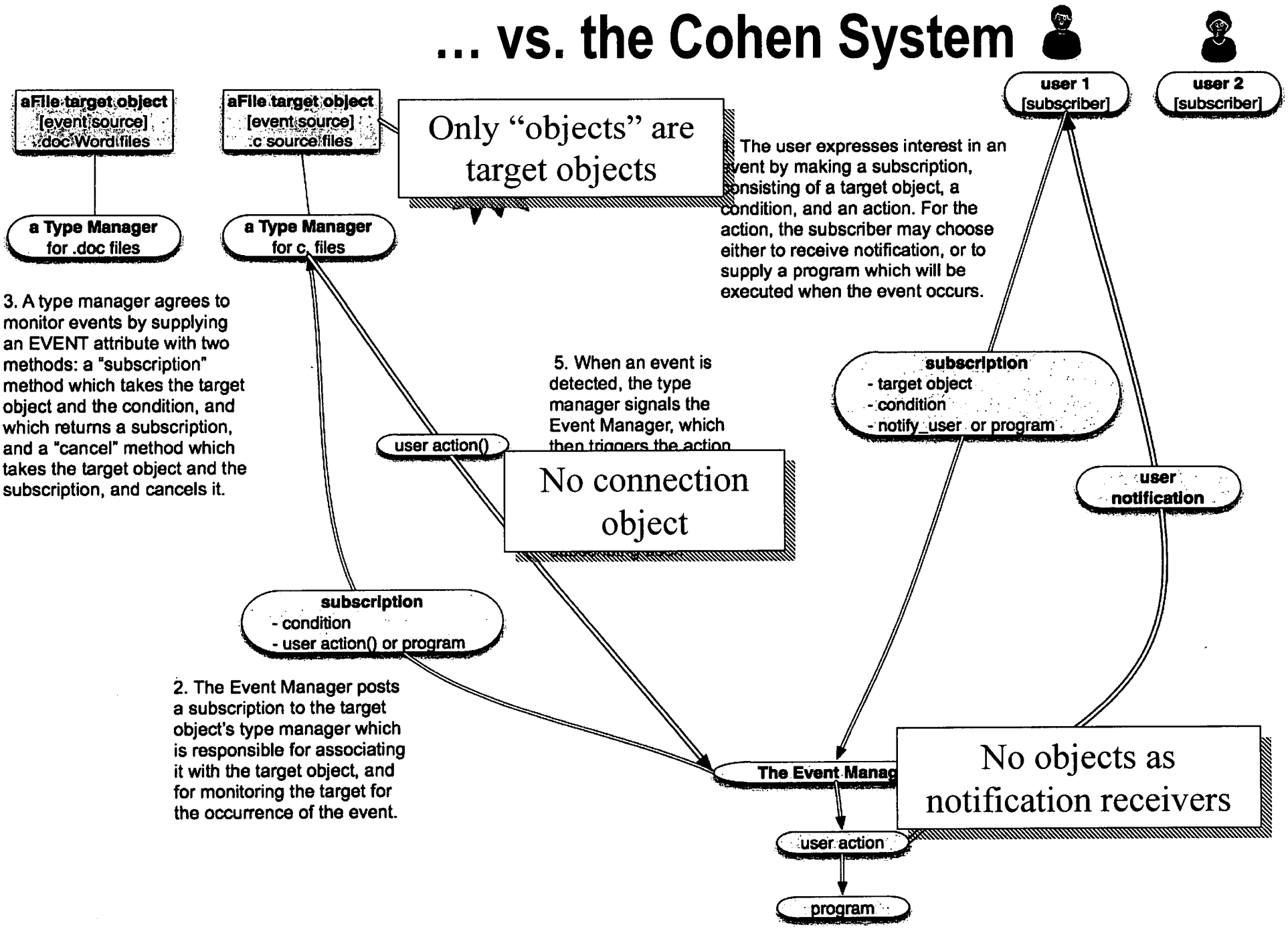
Claim 8



... vs. the Cohen System



... vs. the Cohen System



Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

1. Objects as Notification Receivers

- Claim 8 recites, *inter alia*, "connecting a plurality of objects to a notification source" and "receiving the notification by the at least one of the plurality of objects".

- The Office Action acknowledges that in Cohen, "the subscriber may choose either to receive notification, or to supply a program which will be executed when the event occurs". (Office Action, p. 5)

- Neither of these possible notification receivers in Cohen is an **object** as would be understood by one of ordinary skill in the art.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

1. Objects as Notification Receivers

'703 Patent Claim 8

(plurality of objects)

Notification Receiver **Object**

Notification Receiver **Object**

Claim 8:
(e) receiving the notification by the at least one of the plurality of objects and taking action based on the notification

Cohen

A subscriber (user or program) is not an object

subscriber (user)

subscriber (program)

The user expresses interest in an event by making a ***subscription***, consisting of a ***target*** object, a ***condition***, and an ***action***. For the action, the subscriber may choose either to receive notification, or to supply a program which will be executed when the event occurs.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

1. Objects as Notification Receivers

- Cohen only describes receipt of a notification by a user or a program to be executed. (Cohen, p. 210)
- Therefore, Cohen fails to disclose at least, "receiving the notification by the at least one of the plurality of **objects**".
- Accordingly, the rejection of claims 1 and 8 is in error

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

2. Connection Object

- Claim 8 recites, *inter alia*, "a connection object of an object-oriented operating system."

- Citing to pp. 210-211 of Cohen, the Office Action apparently analogizes the "Event Manager" of Cohen to the connection object of claim 8. (Office Action, p. 4 (citing to section "9. Event Management")).

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

2. Connection Object: store and register

- The claimed connection object must be capable of having connection information ***stored*** and ***registered*** in it (claim 8, elements (b) and (c)).
- According to Cohen (p. 211):
  - Upon subscription, the Event Manager ***passes along*** connection information to a "Type Manager"
  - When an event is subsequently detected, the Type Manager signals the Event Manager, which triggers the action associated with the subscription (i.e. notifying a user or executing a program).
- Accordingly, nothing in Cohen indicates that connection information is ***stored*** or ***registered*** in the Event Manager

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

2. Connection Object: is an *object*

- Furthermore, the Event Manager is not even a connection **object** as claimed.
  - The Event Manager is not object-oriented
  - Furthermore, the type managers are not object-oriented

- Accordingly, the rejection of claims 1 and 8 is in error.

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Cohen is a one-off product

- Gypsy is a narrowly-tailored solution to a very specific problem
  - requires a new type manager for each data type ("***each*** type manager determines the sort of events it will monitor and interprets the condition accordingly") (Cohen, p. 211).
  - requires modification for each problem domain

- In contrast, the notification framework system of the '703 patent was designed to be ***reusable without modification*** in different problem domains.

Conclusion and Summary of Arguments

Sterne Kessler Goldstein Fox
ATTORNEYS AT LAW

Conclusion and Summary

Cohen does not anticipate or render obvious claims 1 and 8 because it does not teach, suggest, or disclose:

1. a **notification receiver object** (e.g. "receiving the notification by the at least one of the ***plurality of objects***"); and

2. a "**connection object**".

Questions and Comments