



US005564048A

# United States Patent [19]

[11] Patent Number: **5,564,048**

Eick et al.

[45] Date of Patent: **Oct. 8, 1996**

- [54] **OBJECT-ORIENTED FUNCTIONALITY CLASS LIBRARY FOR USE IN GRAPHICS PROGRAMMING**
- [75] Inventors: **Stephen G. Eick; Paul J. Lucas**, both of Naperville; **Graham J. Wills**, Lisle, all of Ill.
- [73] Assignee: **Lucent Technologies Inc.**, Murray Hill, N.J.
- [21] Appl. No.: **260,133**
- [22] Filed: **Jun. 15, 1994**
- [51] Int. Cl.<sup>6</sup> ..... **G06F 17/30**
- [52] U.S. Cl. .... **395/600; 395/700; 395/155; 395/156; 395/157; 364/282.3; 364/674; 364/977.2; 364/DIG. 2**
- [58] Field of Search ..... **395/600, 700, 395/375, 650, 155, 156, 157, 158, 159, 160, 161**

Emily Leinfuss, "Managing Class Libraries Takes Discipline", *Software Magazine*, vol. 13, No. 2, Jan. 15, 1993, pp. 15-19.

Allen Holub, "Visual C++: Its Compiler, Language Implementation and Code Quality", *Microsoft Systems Journal*, vol. 8, No. 6, Jun., 1993, pp. 65-75.

Steve Mann, "The Beta Programming Language: an O-O Language With Simula Roots", *Dr. Doob's Journal*, vol. 18, No. 11, Oct., 1993, pp. 56-63.

Ted Faison, "Putting the Owl 2.0 Class Library for Windows Through Its Paces", *Microsoft Systems Journal*, vol. 9, No. 2, Feb., 1994, pp. 45-62.

"Class Libraries for user interface management", Gonzalez, R. E., *Collegiate Microcomputer*, Nov. 1992, USA, vol. 10, No. 4, ISSN 0731-4213, pp. 233-238.

"Zinc Interface Library", Entsminger, G., *Computer Language*, Dec. 1990, USA, vol. 7, No. 12, ISSN 0749-2839, pp. 73-74, 76.

Primary Examiner—Thomas G. Black  
 Assistant Examiner—Paul R. Lintz  
 Attorney, Agent, or Firm—Gordon E. Nelson

- [56] **References Cited**
- U.S. PATENT DOCUMENTS**
- 5,220,500 6/1993 Baird et al. .... 364/408
- 5,379,432 1/1995 Orton et al. .... 395/700
- 5,404,529 4/1995 Chernikoff et al. .... 345/700
- 5,446,902 8/1995 Islam ..... 395/700
- FOREIGN PATENT DOCUMENTS**
- 2661525 10/1991 France .
- OTHER PUBLICATIONS**
- Anonymous, "The World of O-O", *Computer Conference Analysis Newsletter*, No. 296, Mar. 17, 1992, p. 4.
- Korson et al., "Understanding Object-Oriented: a Unifying Paradigm", *Communications of the ACM*, vol. 33, No. 9, Sep., 1990, pp. 40-60.
- Offerman et al., "Objects to the Rescue", *Personal Workstation*, vol. 3, No. 6, Jun., 1991, pp. 50-53.

[57] **ABSTRACT**

A library of C++ classes for use in writing data visualization programs. The library embodies a general design principle for class libraries: that the classes are partitioned into entity classes and functionality classes. Entity classes in the library specify properties of areas in a display and the functionality classes specify functionalities which the areas may have, for example, being responsive to the mouse or the keyboard or being capable of executing drawing commands. The classes are narrowly defined, and when the C++ programmer specifies a class for an object, he or she can give the object exactly the desired properties by selecting from among the area classes and the functionality classes. Because properties are acquired by inheritance, code generated using the libraries is shorter and faster than code generated using conventional class libraries.

**12 Claims, 5 Drawing Sheets**

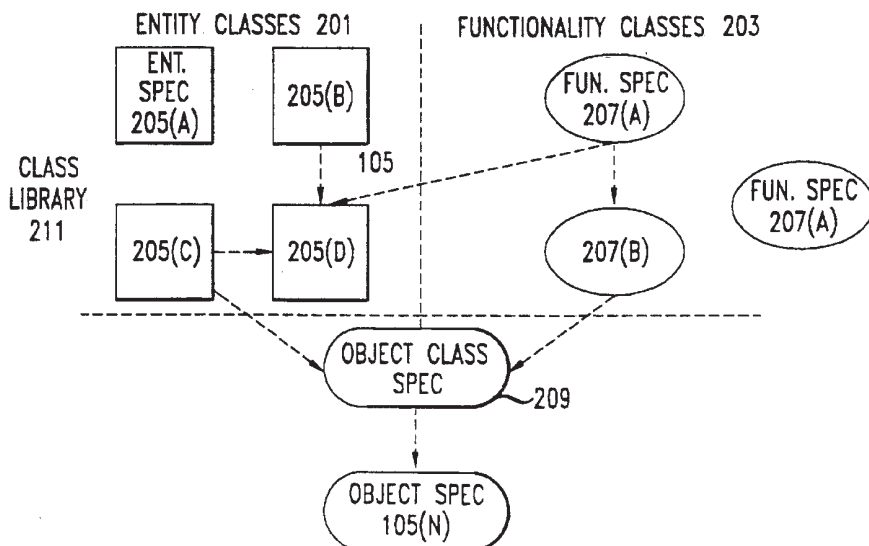


FIG. 1

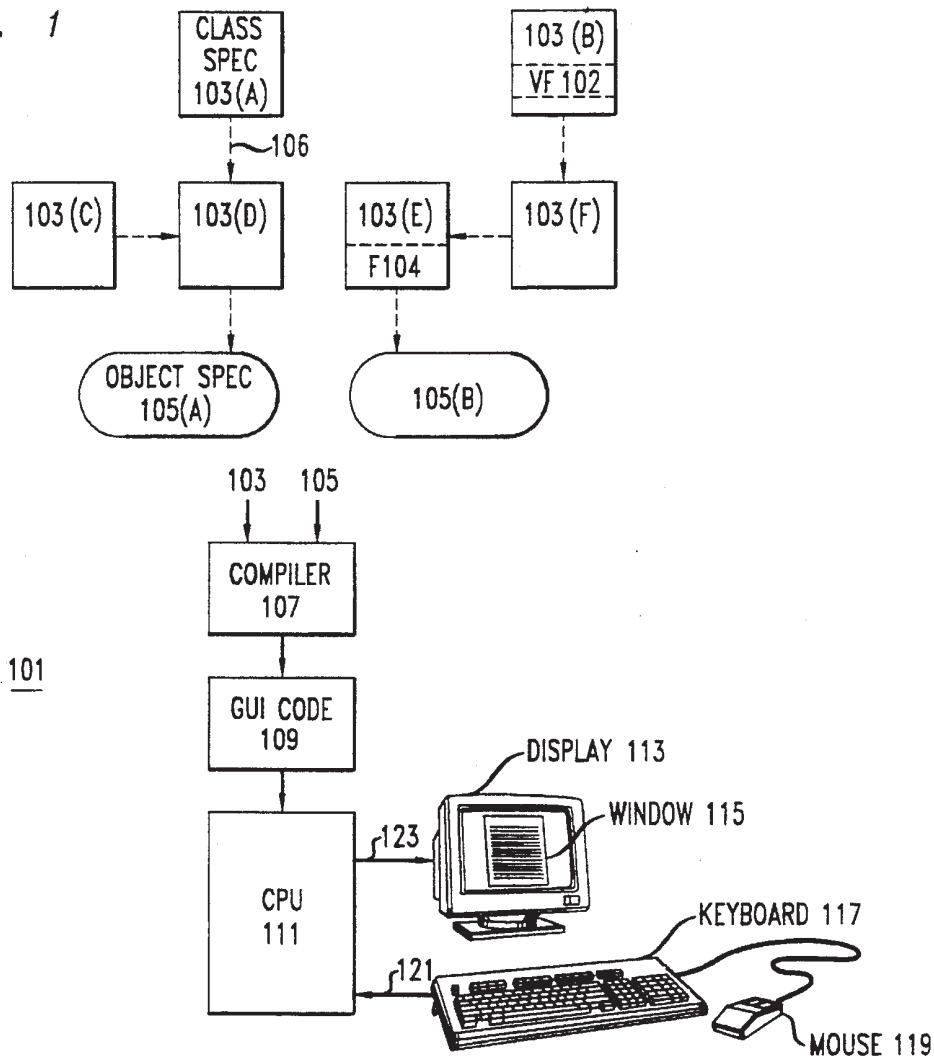
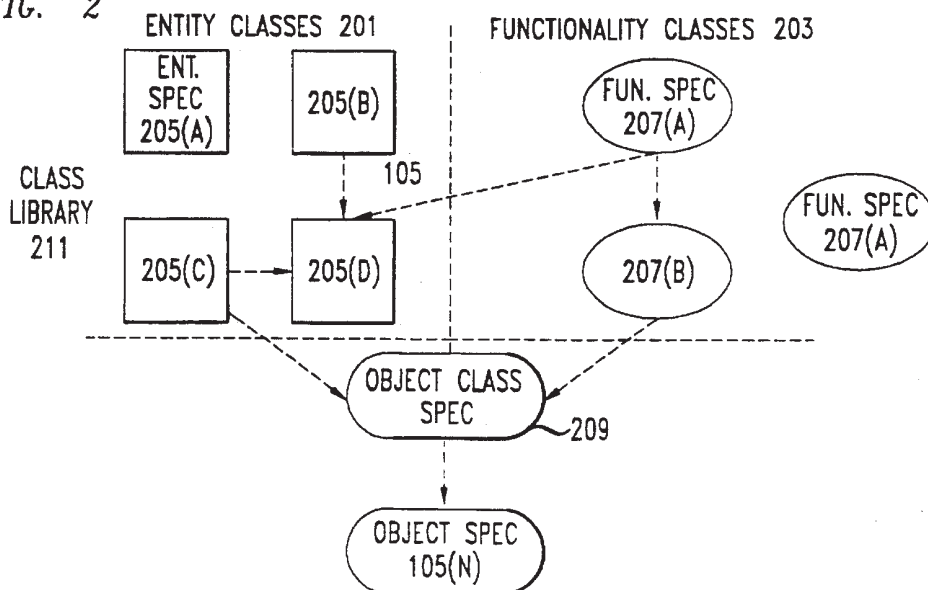


FIG. 2



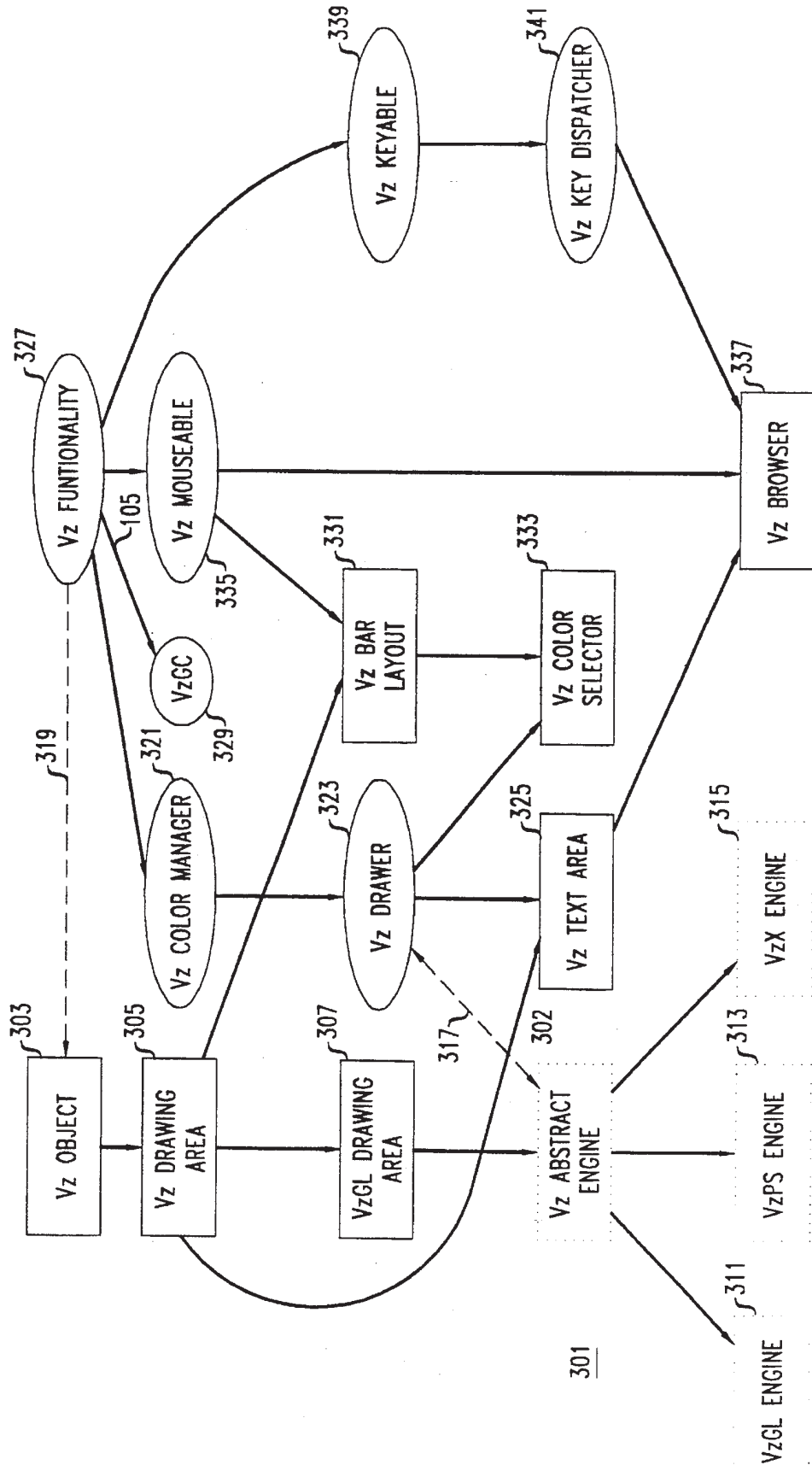


FIG. 3

FIG. 4

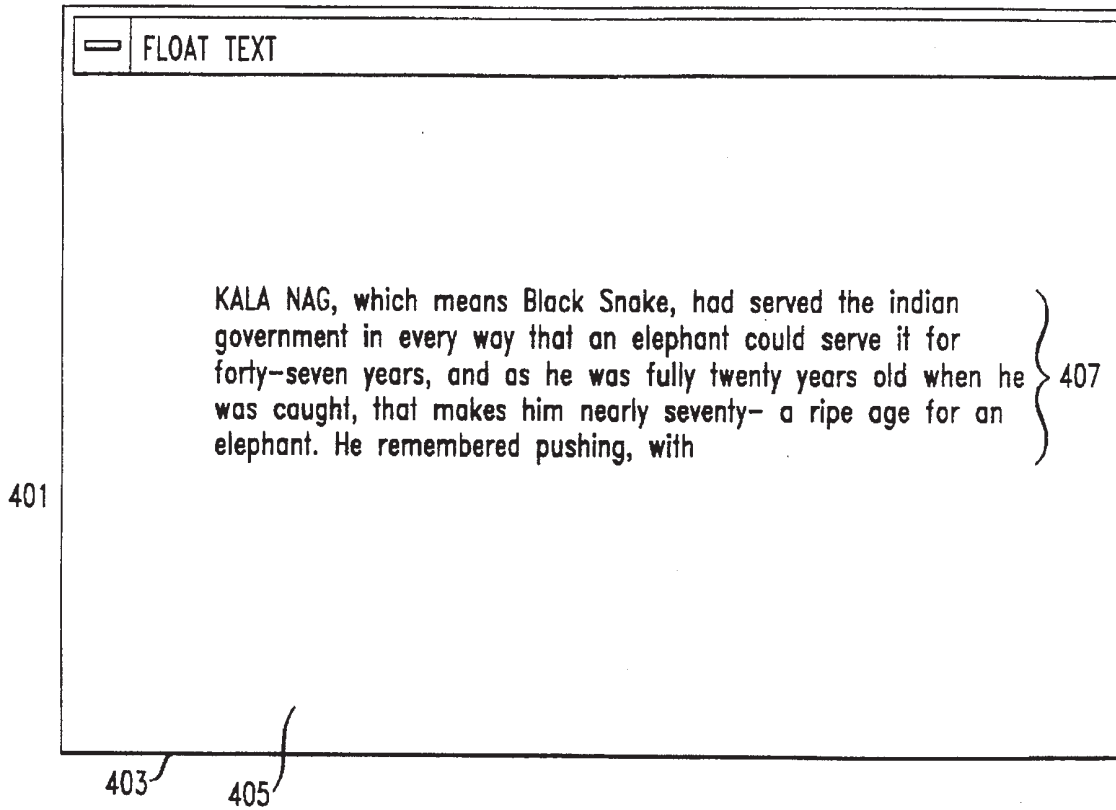


FIG. 5

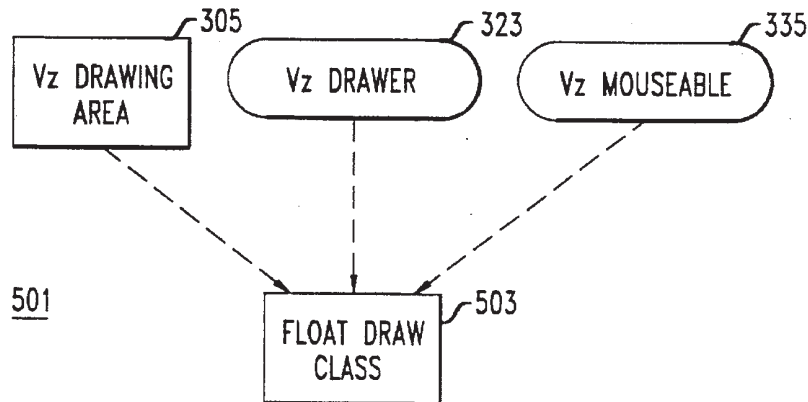


FIG. 6A

Float (toy) - a text browser  
 Graham J. Wills (gwills@research.att.com)  
 Copyright AT&T. All rights reserved

```

#include <string.h>
#include <stream.h>

#include <avector.h>           // Intelligent vectors
#include <charstar.h>         // Intelligent strings

#include <vz/app.h>           603 } // Main window and initialization
#include <vz/drawa.h>         605 } // An area for drawing into
#include <vz/drawer.h>        607 } // Drawing code
#include <vz/mouse.h>         609 } // Mouse handling

static VzColor BACK, DRAWC, HIGHC; // Colors

class FloatDraw : public VzDrawingArea,
                  public VzDrawer,
                  public VzMouseable { } 613
public:

FloatDraw(VzNativeObject w) // Parent Object is this
  : VzDrawingArea("Float", w), // This is a drawing area
    VzMouseable(VzAll), // Want all mouse motions
    VzFunctionality(this) // Annoying C++ requirement
{ } } 615

601 void ReadFile(char *name, char *match); // Do file match 617

// All drawing areas must be able to draw themselves
virtual void DoExpose( int l, int t, int w, int h ); 619

// All mouseables must be able to handle clicks
virtual void DoMouse( VzMouseActions, int, VzNativeEvent const* ); 621

private:
AutoVector<CharString> ln; // Lines of text
AutoVector<short> lnlen; // Their lengths
AutoVector<char> lnmat; // A match?
int maxlen; // Maximum length
int cWid; // Width of each column

```

FIG. 6B

```

Reads file and performs matching
void FloatDraw::ReadFile(char *name, char *match)
{
    ifstream fl(name);

    int maxlen = 0;
    char buffer[9900];
    for(int lines = 0; fl.getline(buffer, 9000); lines++)
    {
        ln[lines] = *buffer ? buffer : " "; // Blank lines become " "
        lnlen[lines] = MeasureString(buffer); // Width in pixels
        if (lnlen[lines] > maxlen) maxlen = lnlen[lines];
        lnmat[lines] = (char) (strchr(buffer, match) != 0);
    }
} } 623

void FloatDraw::DoExpose(int, int, int, int)
{
    int ht = Height()-20;
    cWid = (Width()-20 + 2)/(1 + ln.Count()/ht) - 5;
    if (cWid < 3) cWid = 3;

    int x = 10, y = 10;
    for (int i=0; i<ln.Count(); i++, y++)
    {
        if ( i % (i%ht) ) {
            x += 5*cWid;
            y = 10;
        }
        ForeColor(lnmat[i] ? HIGHC : DRAWC); // Choose appropriate color
        DrawLine( x, y,
                 x+cWid*lnlen[i]/maxlen, y);
    }
} } 625

```

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

## LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

## FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

## E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.