US007240289B2

(12) **United States Patent**
Naughton et al.

(10) **Patent No.: US 7,240,289 B2**
(45) **Date of Patent: Jul. 3, 2007**

(54) **GRAPHICAL USER INTERFACE FOR DISPLAYING AND NAVIGATING IN A DIRECTED GRAPH STRUCTURE**

(75) Inventors: **Patrick J. Naughton**, Palo Alto, CA (US); **Charles H. Clanton, III**, San Francisco, CA (US); **James A. Gosling**, Woodside, CA (US); **Chris Warth**, San Francisco, CA (US); **Joseph M. Palrang**, Sunnyvale, CA (US); **Edward H. Frank**, Portola Valley, CA (US); **David A. LaVallee**, Palo Alto, CA (US); **R. Michael Sheridan**, Oakton, VA (US)

(73) Assignee: **Sun Microsystems, Inc.**, Menlo Park, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 801 days.

(21) Appl. No.: **10/052,747**

(22) Filed: **Jan. 23, 2002**

(65) **Prior Publication Data**

US 2002/0060701 A1 May 23, 2002

(51) **Int. Cl.**
G06F 3/00 (2006.01)
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **715/740**; 715/736; 709/217

(58) **Field of Classification Search** 715/740, 715/844, 716, 744, 839
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,290,142 A 9/1981 Schnee et al.
4,533,910 A 8/1985 Sukonick et al.
4,536,791 A 8/1985 Campbell et al.

(Continued)

FOREIGN PATENT DOCUMENTS

EP 0 239 884 10/1987

(Continued)

OTHER PUBLICATIONS

"Eden Project Proposal", Department of Computer Science, University of Washington, Oct. 1980, Technical Report#80-10-01, cover and Foreword.

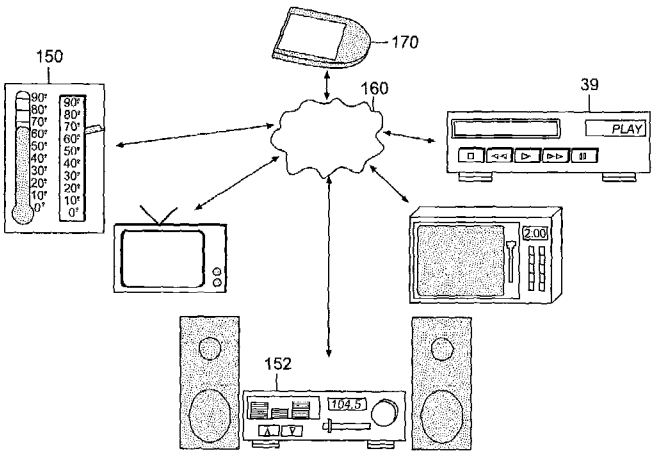
(Continued)

Primary Examiner—Sy D. Luu
(74) *Attorney, Agent, or Firm*—Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.

(57) **ABSTRACT**

An intuitive graphical user interface is based upon a geographic map structure, and includes a system for controlling remote external electronic devices. In the defined graphical user interface, each space of the geographic map structure is rendered on a touch screen display as a graphic image of a geographic space. Within each space are colored cartoon-like icons called "objects" which can be selected and manipulated by the user. Certain objects, referred to as portals, transport the user from one space to another space when Selected. Other objects, referred to as buttons, perform associated actions or functions when Selected. The graphical user interface is displayed on a hand-held display device used to control remote devices. Each remote electronic device transmits a user interface program object that defines a graphical user interface to the display device. The hand-held display devices displays the graphical user interface associated with the remote device when a user selects an icon associated with the remote device on the display device's touch screen display.

8 Claims, 26 Drawing Sheets



U.S. PATENT DOCUMENTS

4,555,775	A		11/1985	Pike
4,573,072	A		2/1986	Freeman
4,622,545	A		11/1986	Atkinson
4,641,205	A		2/1987	Beyers, Jr.
4,653,090	A		3/1987	Hayden
4,706,121	A		11/1987	Young
4,712,191	A		12/1987	Penna
RE32,632	E		3/1988	Atkinson
4,748,618	A		5/1988	Brown et al.
4,750,036	A		6/1988	Martinez
4,772,882	A		9/1988	Mical
4,785,408	A		11/1988	Britton et al.
4,788,538	A		11/1988	Klein et al.
4,812,834	A		3/1989	Wells
4,847,604	A		7/1989	Doyle
4,847,700	A		7/1989	Freeman
4,873,623	A		10/1989	Lane et al.
4,884,223	A		11/1989	Ingle et al.
4,890,320	A		12/1989	Monslow et al.
4,899,136	A		2/1990	Beard et al.
4,914,517	A		4/1990	Duffield
4,914,732	A		4/1990	Henderson et al.
4,931,783	A		6/1990	Atkinson
4,935,865	A		6/1990	Rowe et al.
4,939,507	A		7/1990	Beard et al.
4,959,720	A		9/1990	Duffield et al.
4,977,455	A		12/1990	Young
4,987,486	A		1/1991	Johnson et al.
4,995,078	A		2/1991	Monslow et al.
5,008,853	A		4/1991	Bly et al.
5,014,125	A		5/1991	Pocock et al.
5,021,976	A		6/1991	Wexelblat et al.
5,047,867	A		9/1991	Strubbe et al.
5,062,060	A		10/1991	Kolnick
5,065,347	A		11/1991	Pajak et al.
5,072,412	A		12/1991	Henderson, Jr. et al.
5,086,385	A		2/1992	Launey et al.
5,093,718	A		3/1992	Hoarty et al.
5,107,443	A		4/1992	Smith et al.
5,140,677	A		8/1992	Fleming et al.
5,140,678	A		8/1992	Torres
5,146,556	A		9/1992	Hullot et al.
5,146,598	A	*	9/1992	Takezawa 710/260
5,148,154	A		9/1992	MacKay et al.
5,151,782	A		9/1992	Ferraro
5,151,789	A		9/1992	Young
5,155,806	A		10/1992	Hoeber et al.
5,157,768	A		10/1992	Hoeber et al.
5,177,604	A		1/1993	Martinez
5,195,092	A		3/1993	Wilson et al.
5,204,947	A		4/1993	Bernstein et al.
5,206,722	A		4/1993	Kwan
5,220,420	A		6/1993	Hoarty et al.
5,223,924	A		6/1993	Strubbe
RE34,340	E		8/1993	Freeman
5,233,687	A		8/1993	Henderson, Jr. et al.
5,235,679	A		8/1993	Yoshizawa et al.
5,236,199	A		8/1993	Thompson, Jr.
5,239,540	A		8/1993	Rovira et al.
5,241,671	A		8/1993	Reed et al.
5,247,347	A		9/1993	Litteral et al.
5,253,067	A		10/1993	Chaney et al.
5,295,062	A		3/1994	Fukushima
5,295,243	A		3/1994	Robertson et al.
5,297,253	A		3/1994	Meisel
5,310,349	A		5/1994	Daniels et al.
D348,654	S		7/1994	Branck et al.
5,327,160	A		7/1994	Asher
5,327,161	A		7/1994	Logan et al.
5,333,254	A		7/1994	Robertson
5,333,256	A		7/1994	Green et al.
5,347,628	A		9/1994	Brewer et al.
5,349,658	A		9/1994	O'Rourke et al.
5,357,276	A		10/1994	Banker et al.
5,361,091	A		11/1994	Hoarty et al.
5,363,482	A		11/1994	Victor et al.
5,367,454	A		11/1994	Kawamoto et al.
5,371,553	A	*	12/1994	Kawamura et al. 725/59
5,377,997	A		1/1995	Wilden et al.
5,378,067	A	*	1/1995	Severson et al. 709/237
5,381,158	A		1/1995	Takahara et al.
5,388,993	A		2/1995	McKiel et al.
5,392,389	A		2/1995	Fleming
5,393,073	A		2/1995	Best
5,394,521	A		2/1995	Henderson, Jr. et al.
5,396,546	A		3/1995	Remillard
5,397,678	A		3/1995	Sato et al.
5,410,326	A	*	4/1995	Goldstein 348/734
5,412,377	A		5/1995	Evans et al.
5,416,901	A		5/1995	Torres
5,422,993	A		6/1995	Fleming
5,428,734	A		6/1995	Haynes et al.
5,434,965	A		7/1995	Matheny et al.
5,442,376	A		8/1995	Tannenbaum et al.
5,442,742	A		8/1995	Greyson et al.
5,457,478	A		10/1995	Frank
5,461,710	A		10/1995	Bloomfield et al.
5,463,725	A		10/1995	Henckel et al.
5,463,726	A		10/1995	Price
5,471,190	A	*	11/1995	Zimmermann 340/310.01
5,471,564	A	*	11/1995	Dennis et al. 358/1.15
5,485,197	A		1/1996	Hoarty
5,491,743	A		2/1996	Shiio et al.
5,491,784	A		2/1996	Douglas et al.
5,500,794	A		3/1996	Fujita et al.
5,515,385	A	*	5/1996	Fulling et al. 714/748
5,517,663	A		5/1996	Kahn
5,524,195	A		6/1996	Clanton, III et al.
5,530,865	A		6/1996	Owens et al.
5,543,857	A		8/1996	Wehmeyer et al.
5,555,354	A		9/1996	Strasnick et al.
5,561,444	A		10/1996	Johnston, Jr. et al.
5,638,505	A		6/1997	Hemenway et al.
5,657,221	A	*	8/1997	Warman et al. 700/83
5,666,552	A		9/1997	Greyson et al.
5,689,663	A		11/1997	Williams
5,694,151	A		12/1997	Johnston, Jr. et al.
5,754,178	A		5/1998	Johnston, Jr. et al.
5,886,697	A		3/1999	Naughton et al.
5,995,106	A		11/1999	Naughton et al.
6,061,810	A	*	5/2000	Potter 714/23
6,574,770	B1	*	6/2003	Daudelin 714/776
6,724,732	B1	*	4/2004	Abrams et al. 370/252

FOREIGN PATENT DOCUMENTS

EP	0 438 017	7/1991
EP	0 439 873	8/1991
EP	0 524 103	1/1993
WO	93/11639	11/1992

OTHER PUBLICATIONS

Goldberg et al., "Smalltalk-80—The Language and its Implementation", Xerox Palo Alto Research Center, 1983 (reprinted with corrections, Jul. 1985) pp. 1-720.

Black et al., "Distribution and Abstract Types in Emerald", University of Washington, Technical Report No. 85-08-05, Aug. 1985, pp. 1-10.

Black et al., "Object Structure in the Emerald System", University of Washington, Technical Report 86-04-03, Apr. 1986, pp. 1-14.

Proudfoot, Andrew Blaine, "Replects: data replication in the Eden System", Department of Computer Science, University of Washington, Technical Report No. TR-85-12-04, Dec. 1985, pp. 1-156.

Black et al., "A Language for Distributed Programming", Department of Computer Science, University of Washington, Technical Report 86-02-03, Feb. 1986, p. 10.

Black, Andrew P., "Supporting Distributed Applications: Experience with Eden", Department of Computer Science, University of Washington, Technical Report 85-03-02, Mar. 1985, pp. 1-21.

Black, Andrew P., "The Eden Programming Language", Department of Computer Science, FR-35, University of Washington, Technical Report 85-09-01, Sep. 1985 (Revised, Dec. 1985), pp. 1-19.

Black, Andrew P., "The Eden Project: Overview and Experiences", Department of Computer Science, University of Washington, EUUG, Autumn '86 Conference Proceedings, Manchester, UK, Sep. 22-25, 1986, pp. 177-189.

Black et al., "The Eden Project: A Final Report", Department of Computer Science, University of Washington, Technical Report 86-11-01, Nov. 1986, pp. 1-28.

Pu, Calton, "Replication and Nested Transactions in the Eden Distributed System", Doctoral Disseration, University of Washington, Aug. 6, 1986, pp. 1-179 (1 page Vita).

Holman et al, "The Eden Shared Calendar System", Department of Computer Science, FR-35, University of Washington, Technical Report 85-05-02, Jun. 22, 1985, pp. 1-14.

Jul, Eric, "Object Mobility In A Distributed Object-Oriented System", a Dissertation, University of Washington, 1989, pp. 1-154 (1 page Vita).

Jul et al., "Fine-Grained Mobility in the Emerald System", University of Washington, ACM Transactionis on Computer Systems, vol. 6, No. 1, Feb. 1988, pp. 109-133.

Hsu, Felix Samson, "Reimplementing Remote Procedure Calls", University of Washington, Thesis, Mar. 22, 1985, pp. 1-106.

Almes et al., "Edmas: A Locally Distributed Mail System", Department of Computer Science, University of Washington, Technical Report 83-07-01, Jul. 7, 1983, Abstract & pp. 1-17.

Almes, Guy T., "Integration and Distribution in the Eden System", Department of Computer Science, University of Washington, Technical Report 83-01-02, Jan. 19, 1983, pp. 1-18 & Abstract.

Almes, Guy T., "The Evolution of the Eden Invocation Mechanism", Department of Computer Science, University of Washington, Technical Report 83-01-03, Jan. 19, 1983, pp. 1-14 & Abstract.

Almes et al., "The Eden System: A Technical Review", Department of Computer Science, University of Washington, Technical Report 83-10-05, Oct. 1983, pp. 1-25.

Almes et al., "Research in Integrated Distributed Computing", Department of Computer Science, University of Washington, Oct. 1979, pp. 1-42.

Krasner et al., "SMALLTALK-80: Bits Of History, Words Of Advice", 1983 Xerox Corporation, pp. 1-345.

Hutchinson, Norman C., "Emerald: An Object-Based Language for Distributed Programming", a Dissertation, University of Washington, 1987, pp. 1-107.

Lazowska et al., "The Architecture of the Eden System", Department of Computer Science, University of Washington, Proceedings of the Eighth Symposium on Operating Systems Principles, Dec. 14-16, 1981, ACM, Special Interest Group on Operating Systems, Associataion for Computing Machinery, vol. 15, No. 5, Dec. 1981, ACM Order No. 534810, pp. 148-159.

Alexander, "Visualizing Cleared-Off Desktops," Computerworld, May 6, 1991, p. 20.

JeffreyJ. Blatt, "A Primer on User Interface Software Patents," 9 The Computer Lawyer, vol. 9, No. 4, Apr. 1992.

P. Chan, "Learning Considerations in User Interface Design: The Room Model," Report CS-84-16, University. of Waterloo, Computer Science Dept., Waterloo, Ontario, Canada, Jul. 1984.

Stavros Christodoulakis et al., "Browsing Within Time-Driven Multimedia Documents," Institute for Computer Research, University of Waterloo, Waterloo, Ontario, Canada, Jul. 1988.

Chung et al., "A 'Tiny' Pascal Compiler: the P-Code Interpreter," BYTE Publications Inc., Sep. 1978.

Chung et al., "A 'Tiny' Pascal Compiler Part 2: The P-Compiler," BYTE Publications Inc., Oct. 1978.

Conference Proceedings, The 10th Annual International Symposium on Computer Architecture, Royal Institute of Technnology, Stockholm, Sweden, 1983.

Hoarty, "Multimedia on Cable Television Systems," Oct. 5, 1993, pp. 555-567.

Hsia et al., "Construction and Manipulation of Dynamic Icons," IEEE 78, 1988.

(IBM), "Conferencing Metaphor," Feb. 2, 1993, pp. 63-66.

IBM Technical Disclosure Bulletin, "Creation/Modification of the Audio Signal Processor Setup for a PC Audio Editor," vol. 30, No. 10, Mar. 1988.

IBM Technical Disclosure Bulletin, vol. 34, No. 5, pp. 24-26, Oct. 5, 1991.

IBM Technical Disclosure Bulletin, vol. 36, No. 4, pp. 215-216, Apr. 1993.

Hiroshi Ishii et al., "Clearface: Translucent Multiuser Interface for TeamWork Station," ECSCW, Sep. 1991, pp. 6-10.

Hiroshi Ishii et al., "Toward an Open Shared Workspace," Communications of the ACM, vol. 34, No. 12, Dec., 1991, pp. 37-50.

B. Laurel, "Computers as Theater," Addison Wesley Pub. Co., 1990.

B. Laurel, "Computers as Theater," Addison Wesley Pub. Co., 1993.

B. Laurel, "The Art of Human-Computer Interface Design," Addison Wesley Pub. Co., 1990.

B. Laurel, "The Art of Human Computer Interface Design," Apple Computer Inc., 1991.

Gene McDaniel, "An Analysis of a Mesa Instruction Set," Palto Alto Research Centers, 1982.

Brad A. Meyers, "The Importance of Percent-Done Progress Indicators for Computer-Human Interfaces," ACM 11, 1989.

Mitchell et al., "Mesa Language Manual," Palo Alto Research Centers.

Kenneth A. Pier, "A Retrospective on the Dorado, A High-Performance Personal Computer," Palo Alto Research Centers, Aug. 1983.

George Robertson et al., Communications for the Assoc. for Computing Machinery, "Information Visualization Using 3D Interactive Animation," vol. 36, No. 4, Apr., 1993, pp. 57-71.

George Robertson et al., UIST Proceedings of the ACM Siggraph Symposium on User Interface Software and Technology, "The Cognitive Coprocessor Architecture for Interactive User Interfaces," Nov. 13-15, 1989, Williamsburg, VA, pp. 10-18.

Sega of America, "Sonic the Hedgehog," 1991.

Taingawa, "Personal Multimedia-Multipoint Teleconference System," Apr. 7, 1991, pp. 1127-1144.

Thompson, "Regular Expression Search Algorithm," Communications of the ACM, vol. 11, No. 6, Jun. 1968 at 418 *et seq.*

Y. Tonomura et al., "Content Oriented Interface Using Video Icons for Visual Database Systems,"IEEE 68, 1989.

Hirotada Ueda et al., "Impact: An Interactive Natural-Motion-Picture Dedicated Multi-Medi Authoring System," Communications of the ACM, Mar. 1991, pp. 343-350.

* cited by examiner

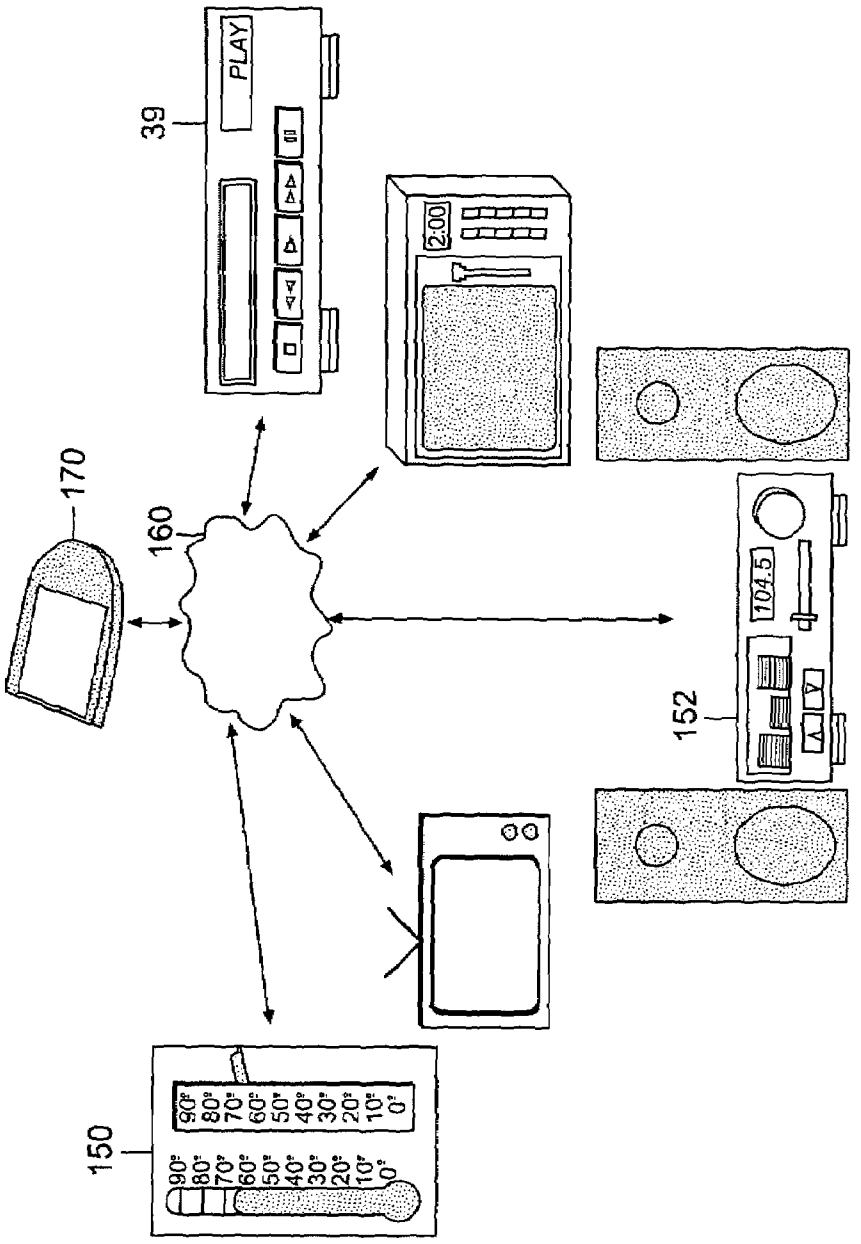


FIG. 1A

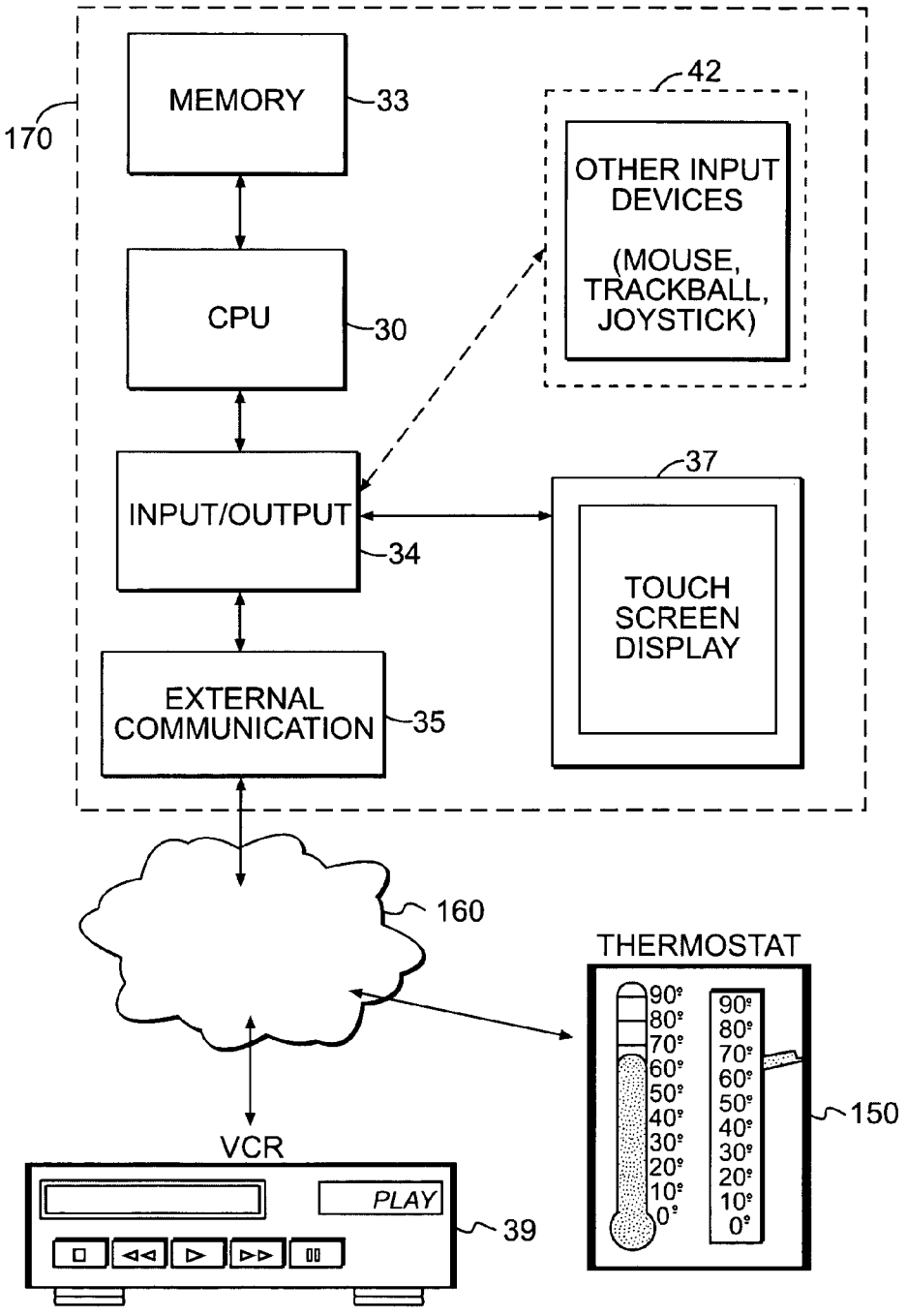

170
MEMORY
33
42
OTHER INPUT DEVICES
(MOUSE, TRACKBALL, JOYSTICK)
CPU
30
INPUT/OUTPUT
34
37
TOUCH SCREEN DISPLAY
EXTERNAL COMMUNICATION
35
160
THERMOSTAT
90°
80°
70°
60°
50°
40°
30°
20°
10°
0°
150
VCR
PLAY
39

FIG. 1B

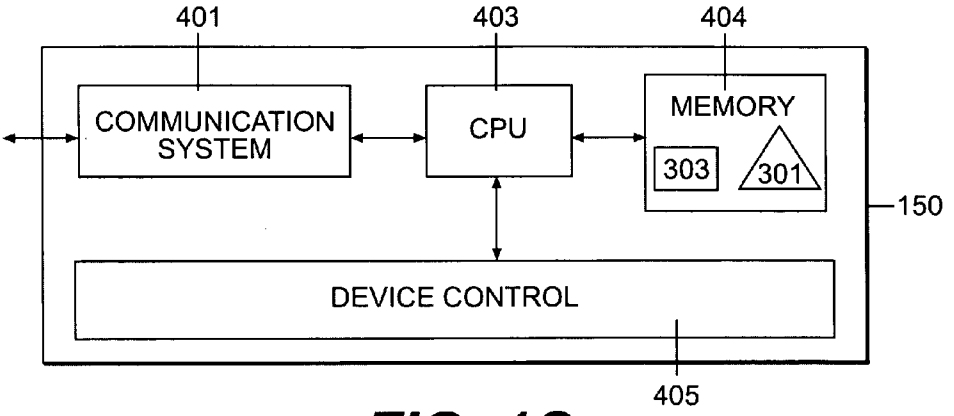


FIG. 1C

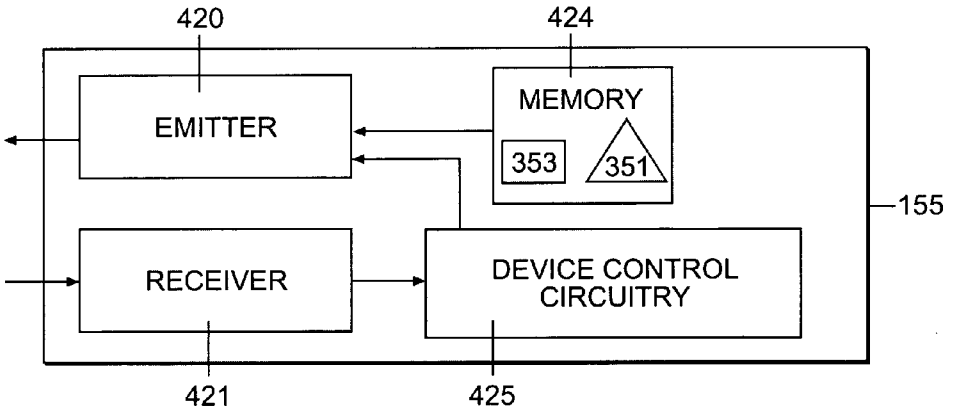


FIG. 1D

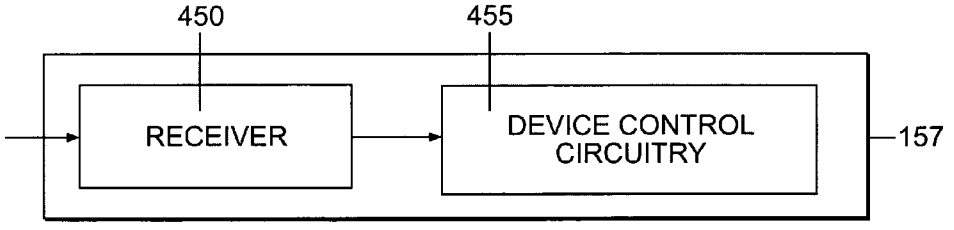


FIG. 1E
PRIOR ART

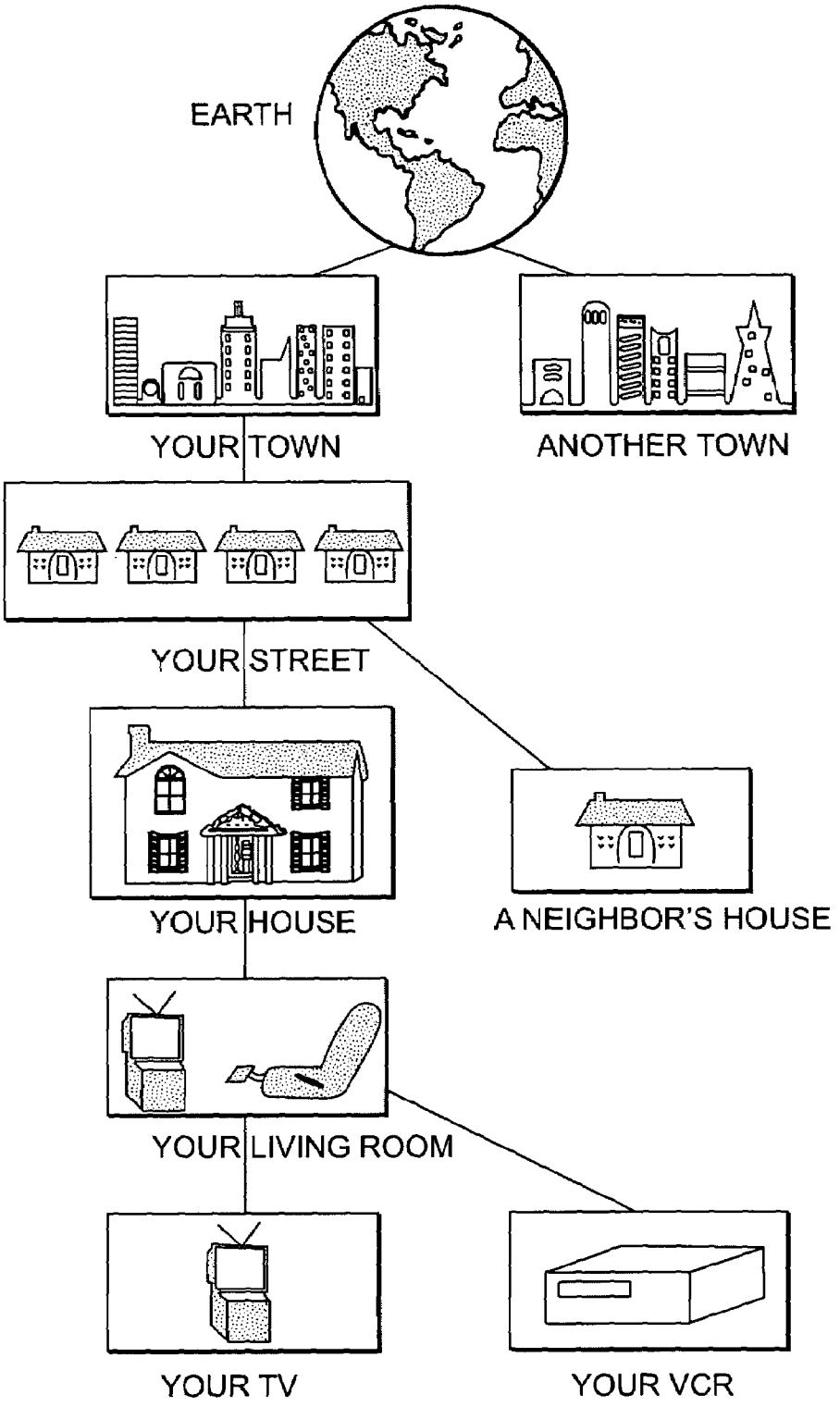

EARTH
YOUR TOWN
ANOTHER TOWN
YOUR STREET
YOUR HOUSE
A NEIGHBOR'S HOUSE
YOUR LIVING ROOM
YOUR TV
YOUR VCR

FIG. 2A

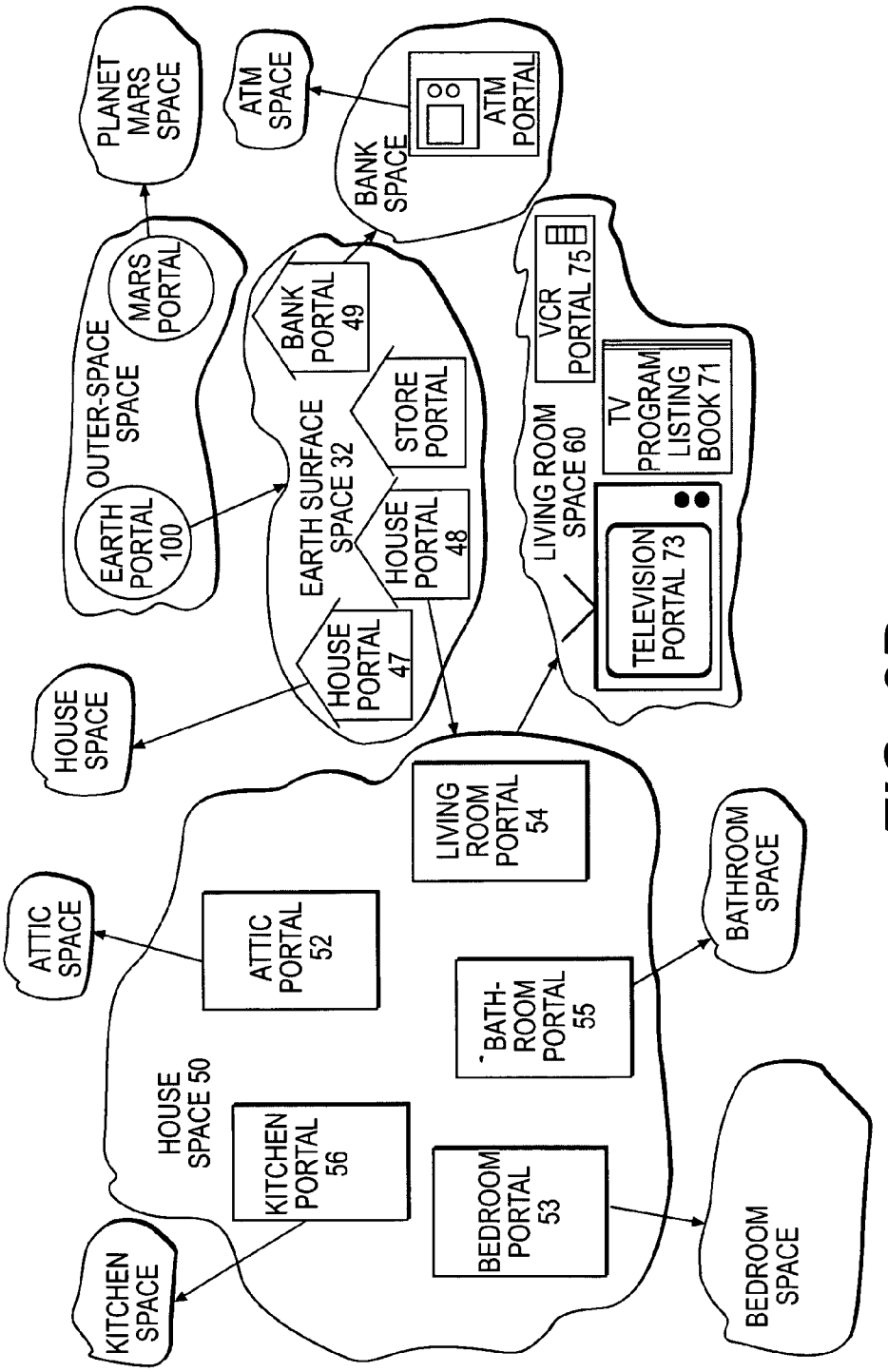

PLANET MARS SPACE
ATM SPACE
BANK SPACE
ATM PORTAL
OUTER-SPACE SPACE
MARS PORTAL
BANK PORTAL 49
EARTH PORTAL 100
EARTH SURFACE SPACE 32
STORE PORTAL
HOUSE PORTAL 48
HOUSE PORTAL 47
VCR PORTAL 75
LIVING ROOM SPACE 60
TV PROGRAM LISTING BOOK 71
TELEVISION PORTAL 73
HOUSE SPACE
LIVING ROOM PORTAL 54
ATTIC SPACE
ATTIC PORTAL 52
BATHROOM SPACE
BATH-ROOM PORTAL 55
HOUSE SPACE 50
KITCHEN PORTAL 56
BEDROOM PORTAL 53
KITCHEN SPACE
BEDROOM SPACE

FIG. 2B

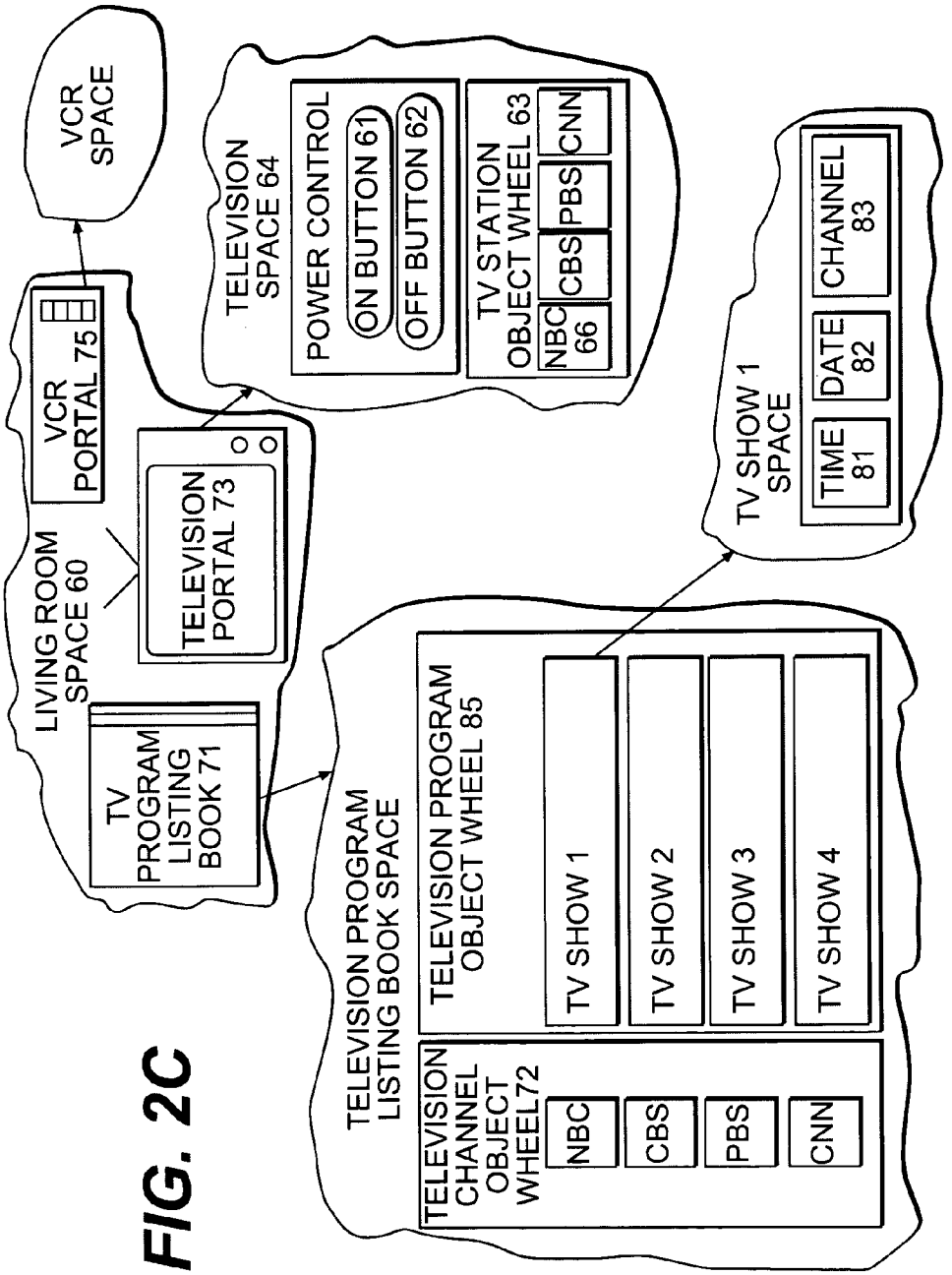

VCR SPACE
VCR PORTAL 75
LIVING ROOM SPACE 60
TELEVISION PORTAL 73
TV PROGRAM LISTING BOOK 71
TELEVISION SPACE 64
POWER CONTROL
ON BUTTON 61
OFF BUTTON 62
TV STATION OBJECT WHEEL 63
NBC 66
CBS
PBS
CNN
TV SHOW 1 SPACE
TIME 81
DATE 82
CHANNEL 83
TELEVISION PROGRAM LISTING BOOK SPACE
TELEVISION PROGRAM OBJECT WHEEL 85
TV SHOW 1
TV SHOW 2
TV SHOW 3
TV SHOW 4
TELEVISION CHANNEL OBJECT WHEEL72
NBC
CBS
PBS
CNN

FIG. 2C

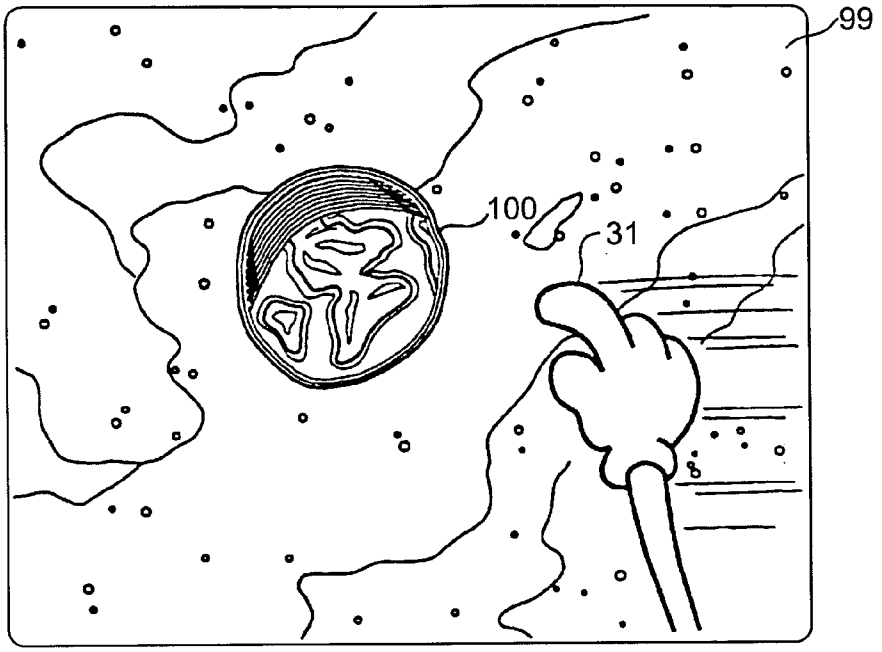


FIG. 3

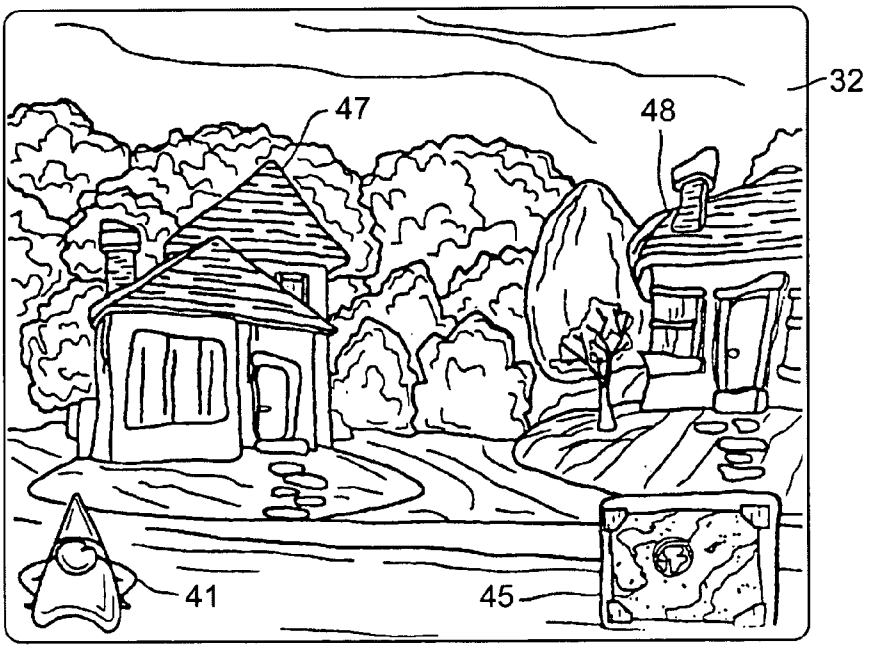


FIG. 4

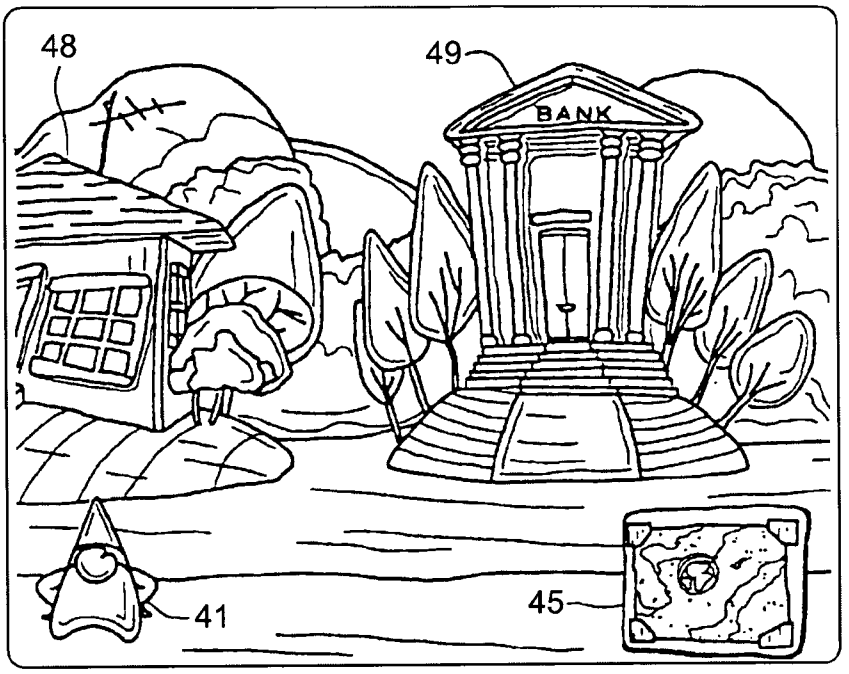


FIG. 5

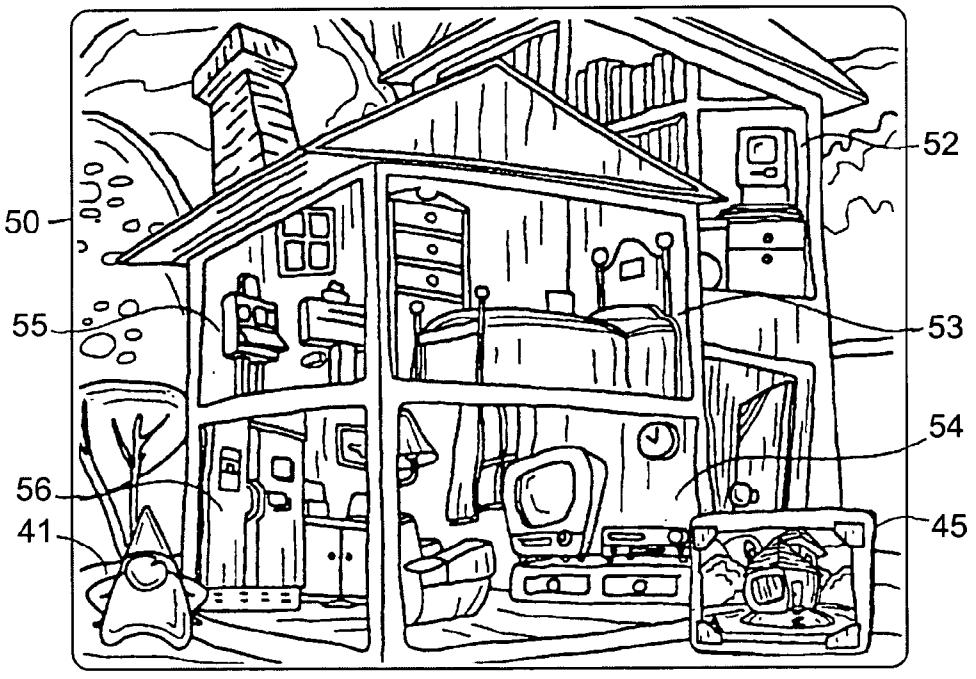


FIG. 6

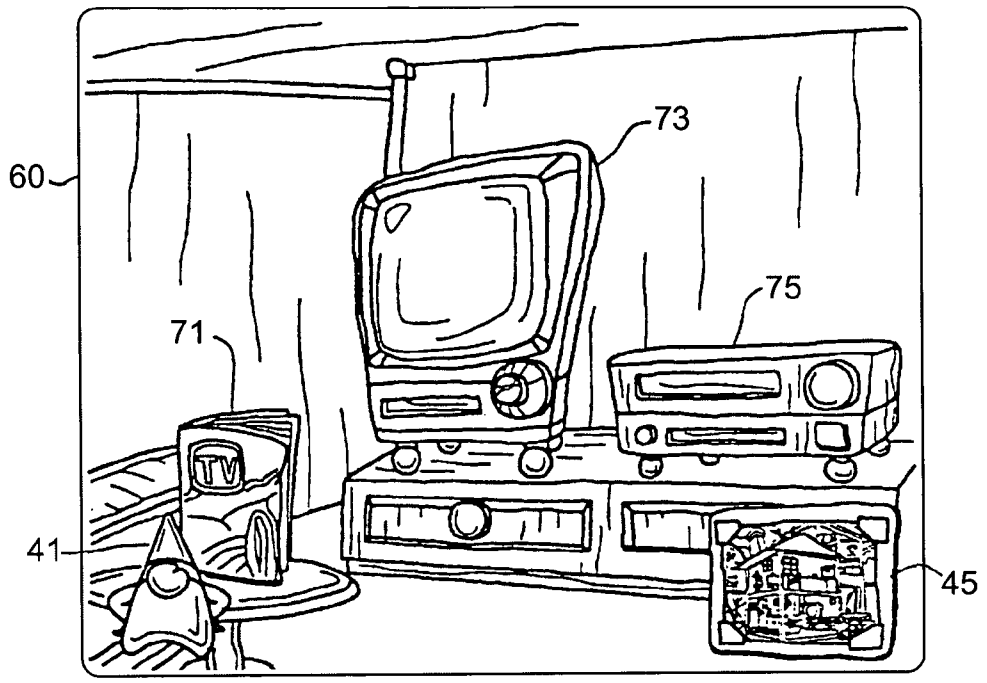


FIG. 7

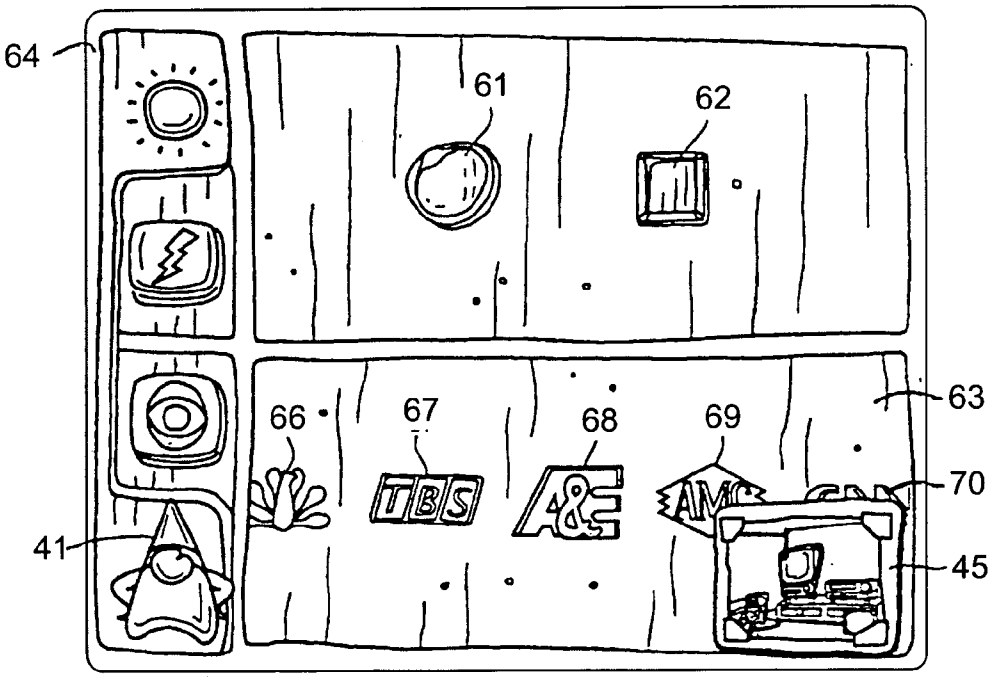


FIG. 8

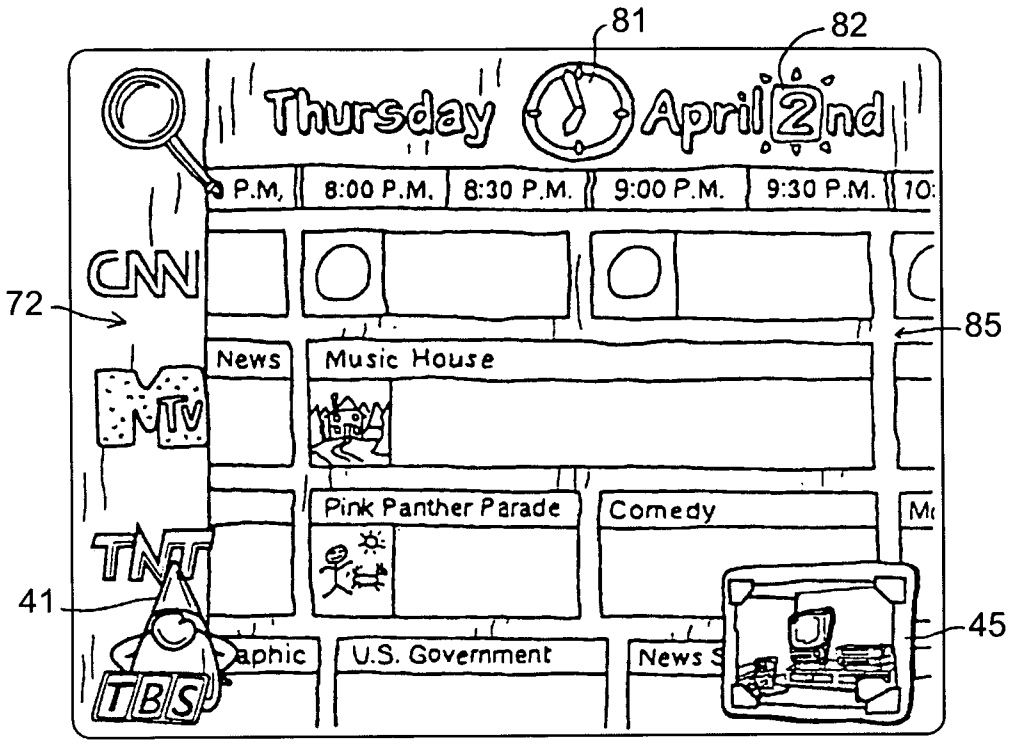

81
82
Thursday
April 2nd
P.M.
8:00 P.M.
8:30 P.M.
9:00 P.M.
9:30 P.M.
CNN
72
85
News
Music House
NTV
Pink Panther Parade
Comedy
TNT
41
45
aphic
U.S. Government
TBS

FIG. 9

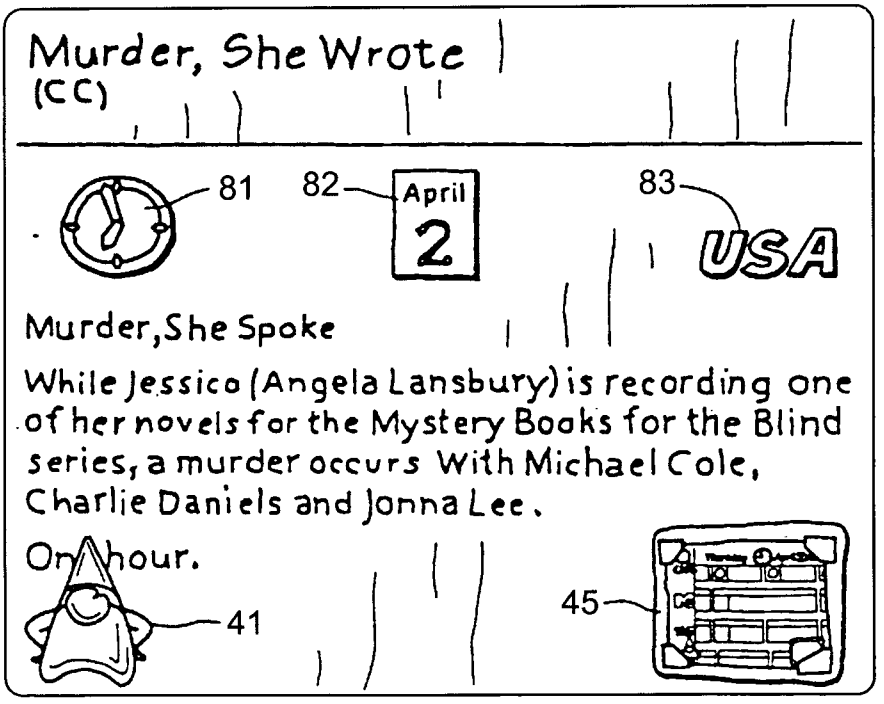

Murder, She Wrote
(CC)
81
82
April
2
83
USA
Murder, She Spoke
While Jessica (Angela Lansbury) is recording one of her novels for the Mystery Books for the Blind series, a murder occurs With Michael Cole, Charlie Daniels and Jonna Lee.
On hour.
41
45

FIG. 10

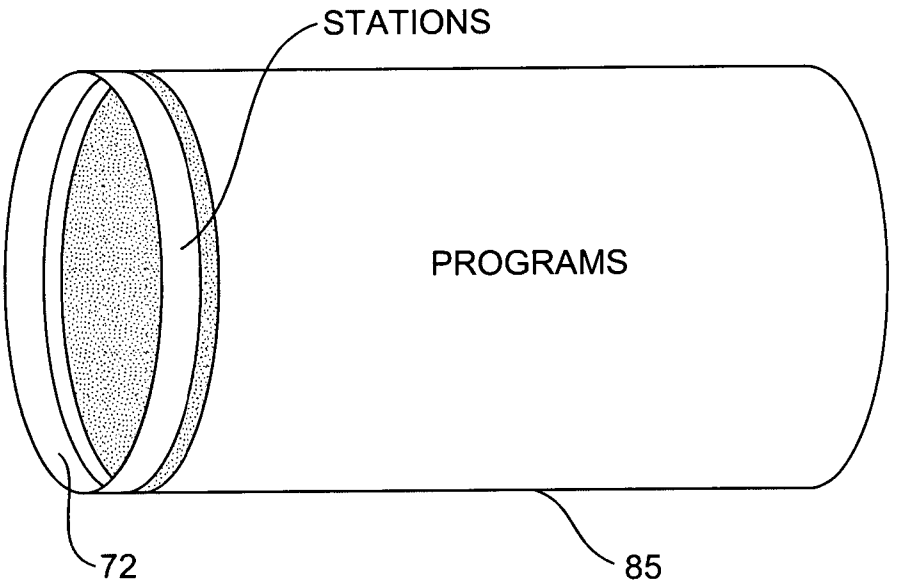


FIG. 11

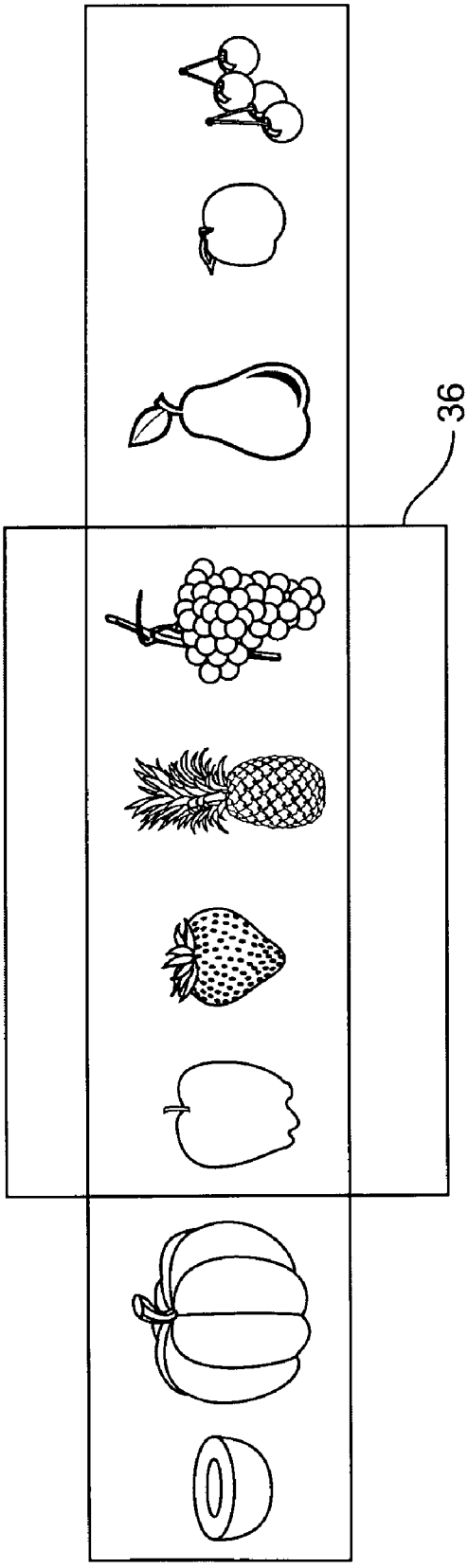


FIG. 12

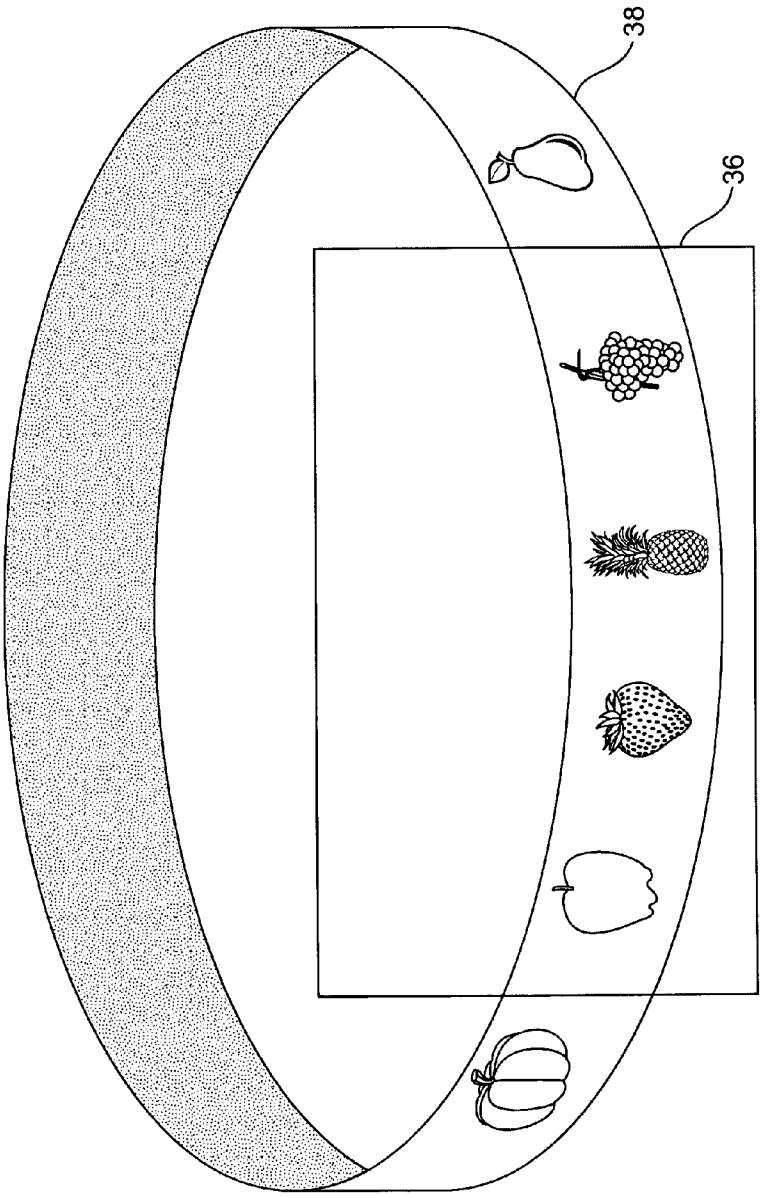


FIG. 13

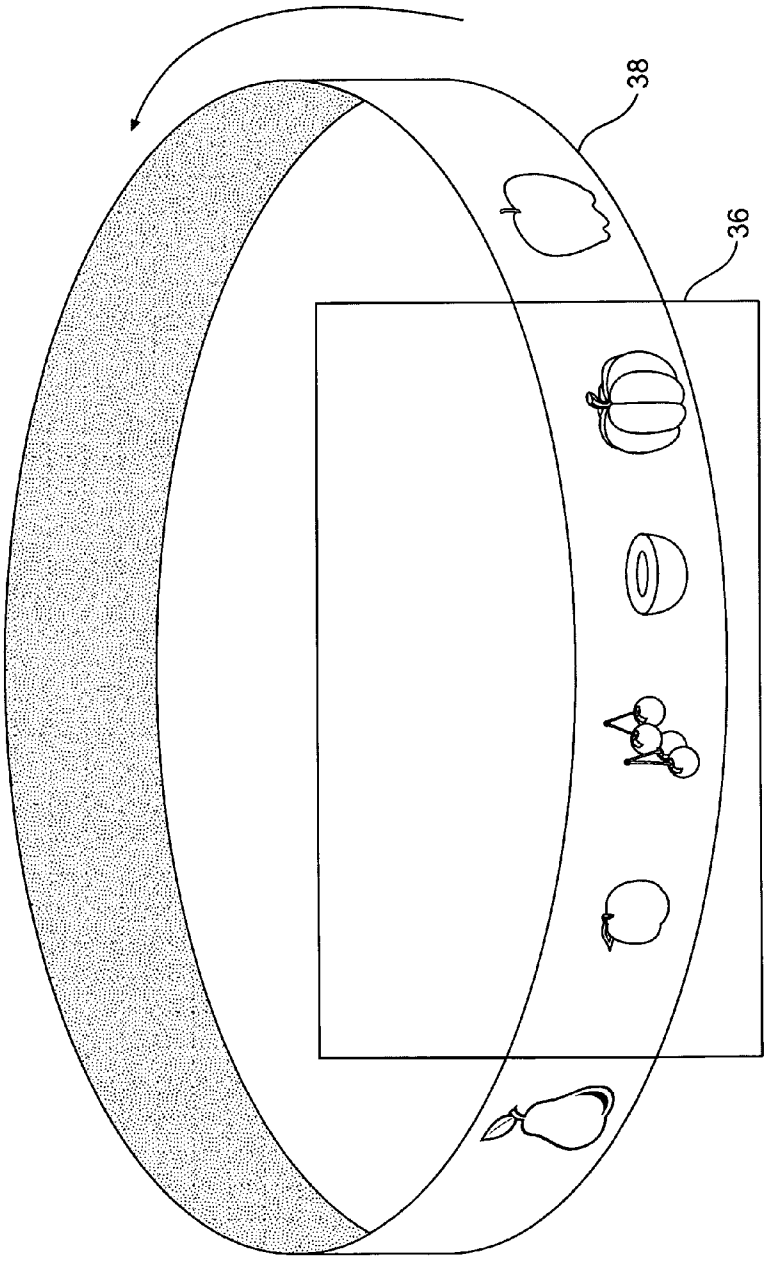


FIG. 14

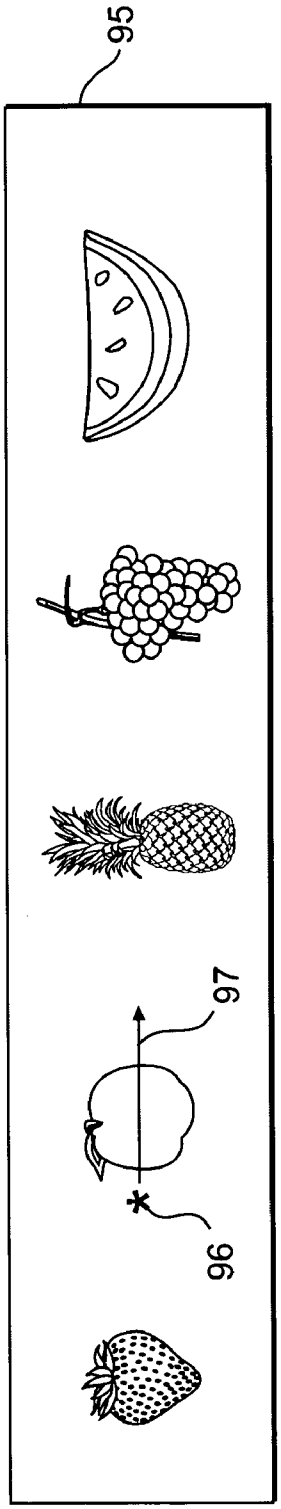


FIG. 15

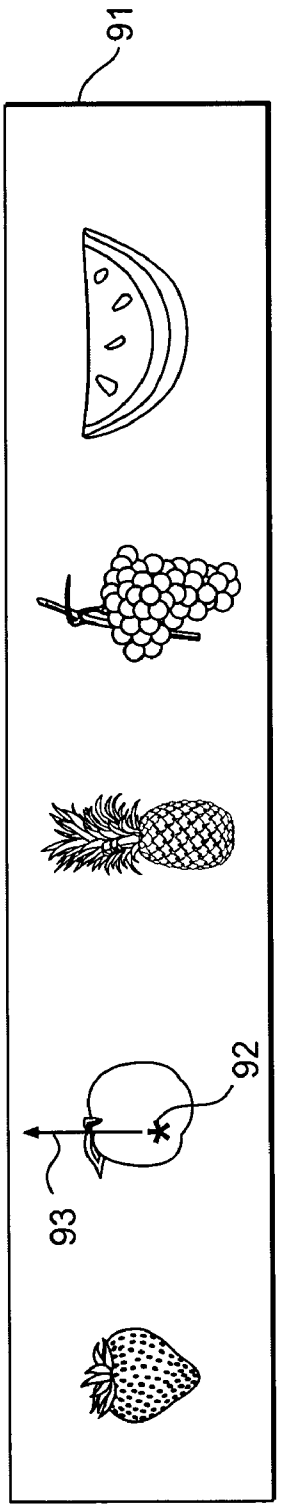


FIG. 16A

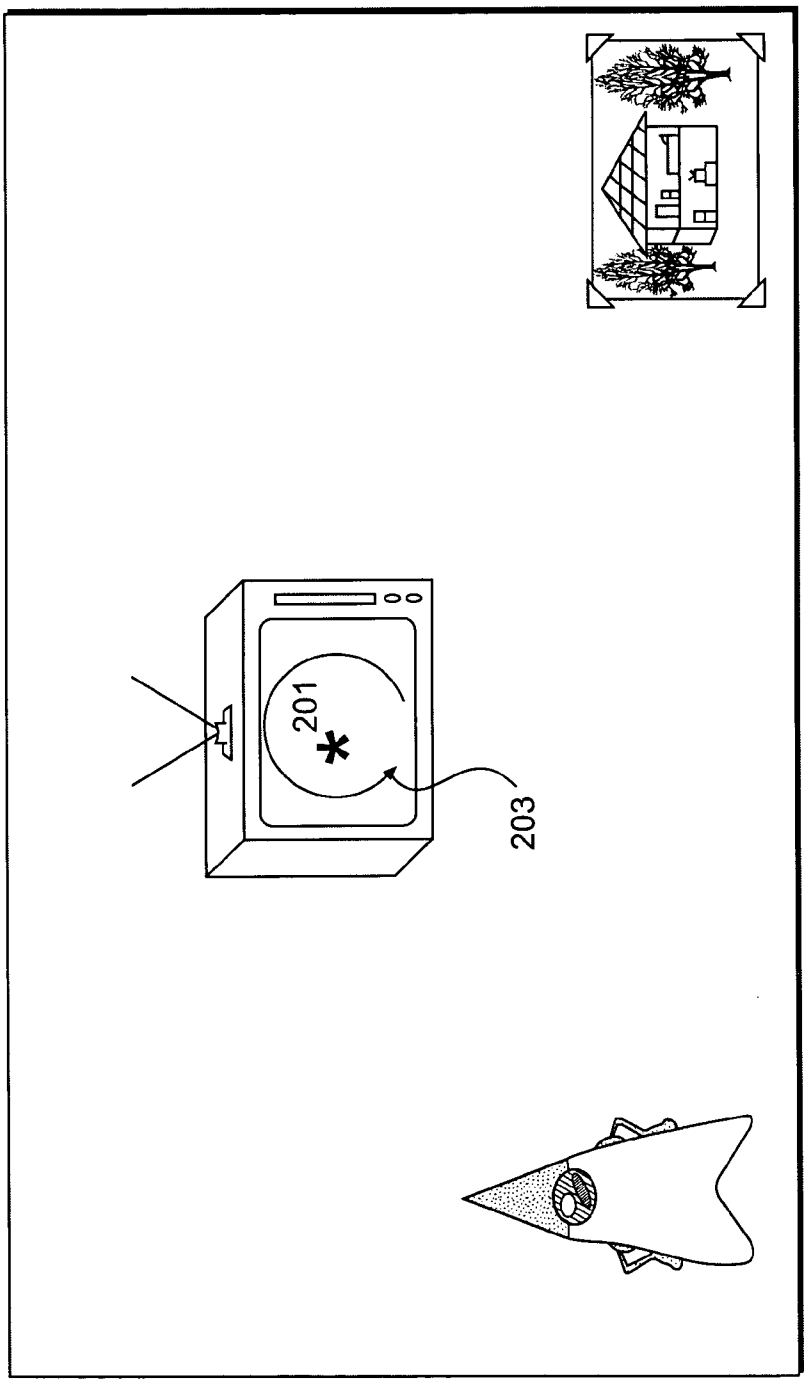


FIG. 16B

FIG. 17A

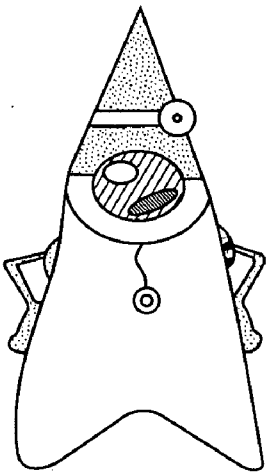


FIG. 17B

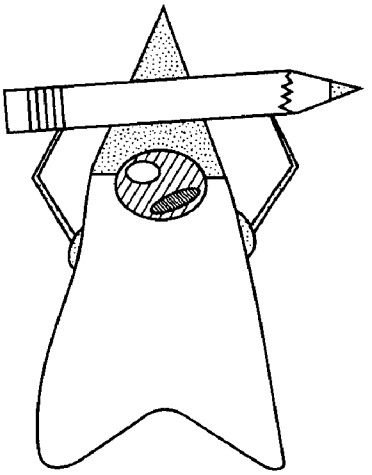
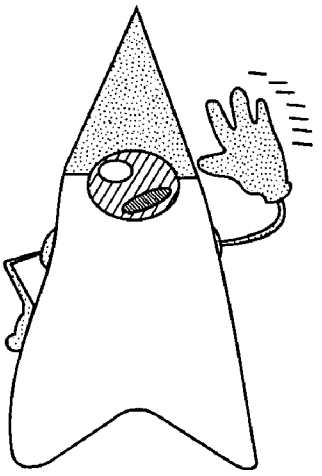


FIG. 17C



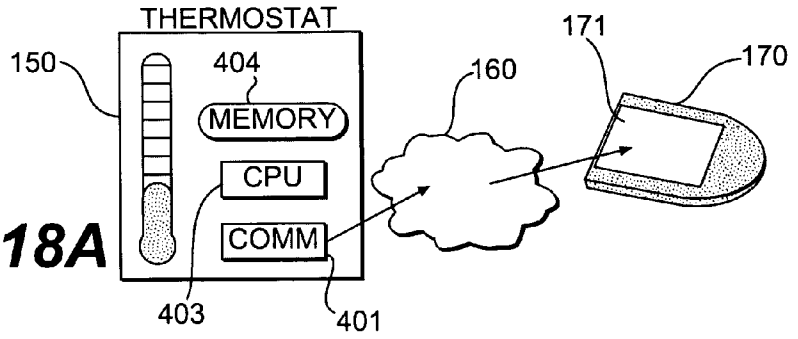


FIG. 18A

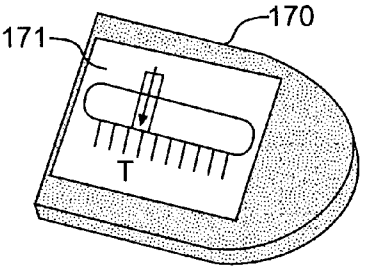


FIG. 18B

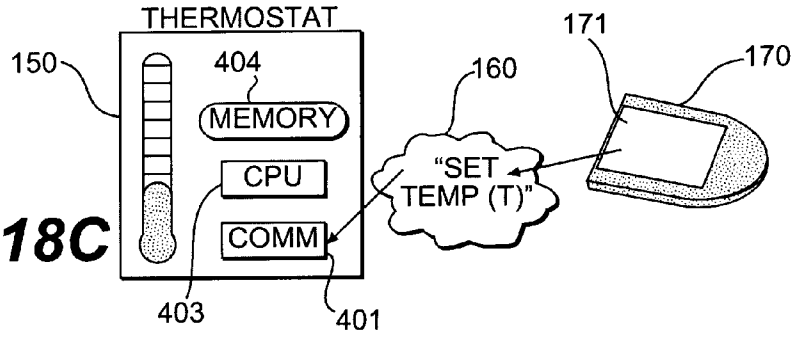


FIG. 18C

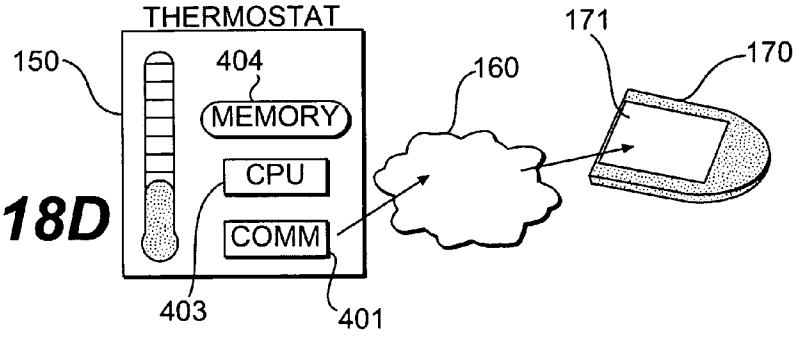


FIG. 18D

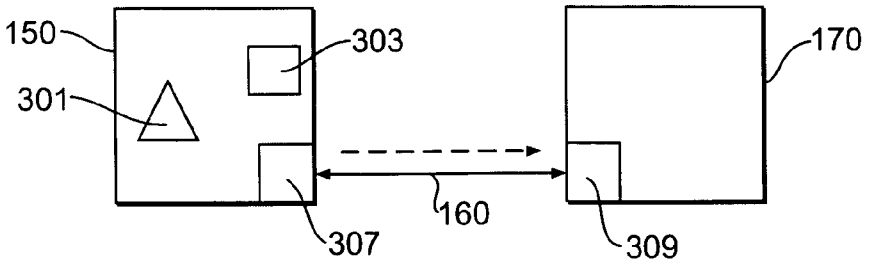


FIG. 19A

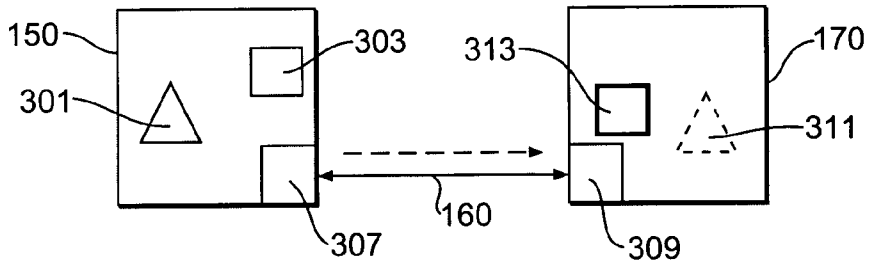


FIG. 19B

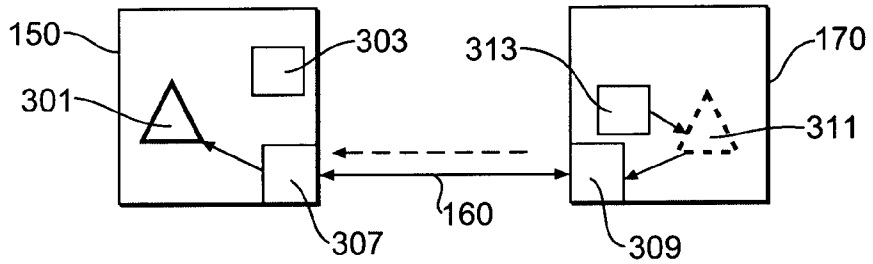


FIG. 19C

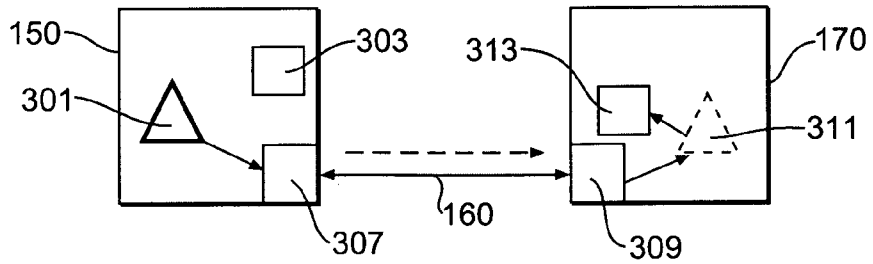


FIG. 19D

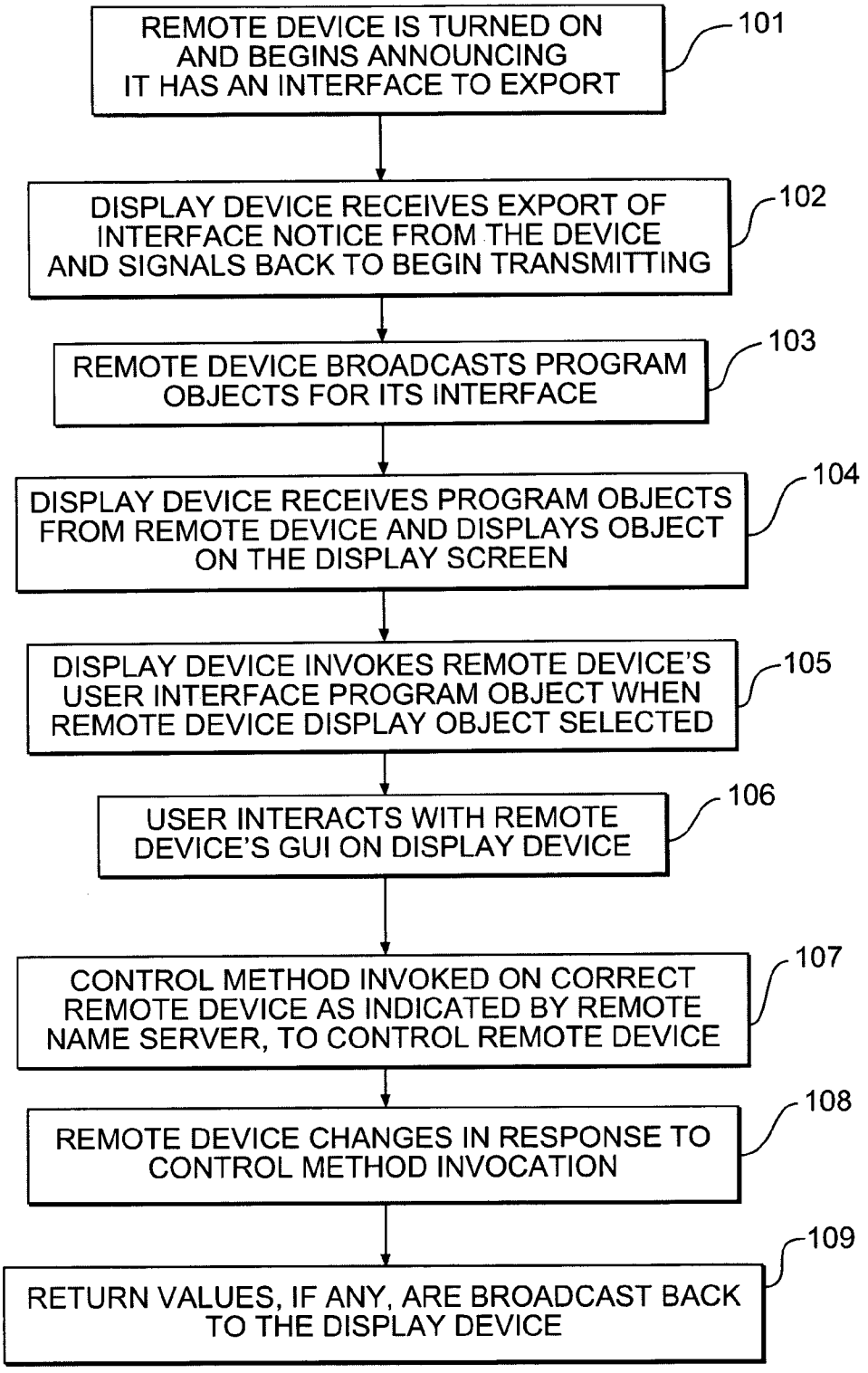

REMOTE DEVICE IS TURNED ON AND BEGINS ANNOUNCING IT HAS AN INTERFACE TO EXPORT
101
DISPLAY DEVICE RECEIVES EXPORT OF INTERFACE NOTICE FROM THE DEVICE AND SIGNALS BACK TO BEGIN TRANSMITTING
102
REMOTE DEVICE BROADCASTS PROGRAM OBJECTS FOR ITS INTERFACE
103
DISPLAY DEVICE RECEIVES PROGRAM OBJECTS FROM REMOTE DEVICE AND DISPLAYS OBJECT ON THE DISPLAY SCREEN
104
DISPLAY DEVICE INVOKES REMOTE DEVICE'S USER INTERFACE PROGRAM OBJECT WHEN REMOTE DEVICE DISPLAY OBJECT SELECTED
105
USER INTERACTS WITH REMOTE DEVICE'S GUI ON DISPLAY DEVICE
106
CONTROL METHOD INVOKED ON CORRECT REMOTE DEVICE AS INDICATED BY REMOTE NAME SERVER, TO CONTROL REMOTE DEVICE
107
REMOTE DEVICE CHANGES IN RESPONSE TO CONTROL METHOD INVOCATION
108
RETURN VALUES, IF ANY, ARE BROADCAST BACK TO THE DISPLAY DEVICE
109

FIG. 20

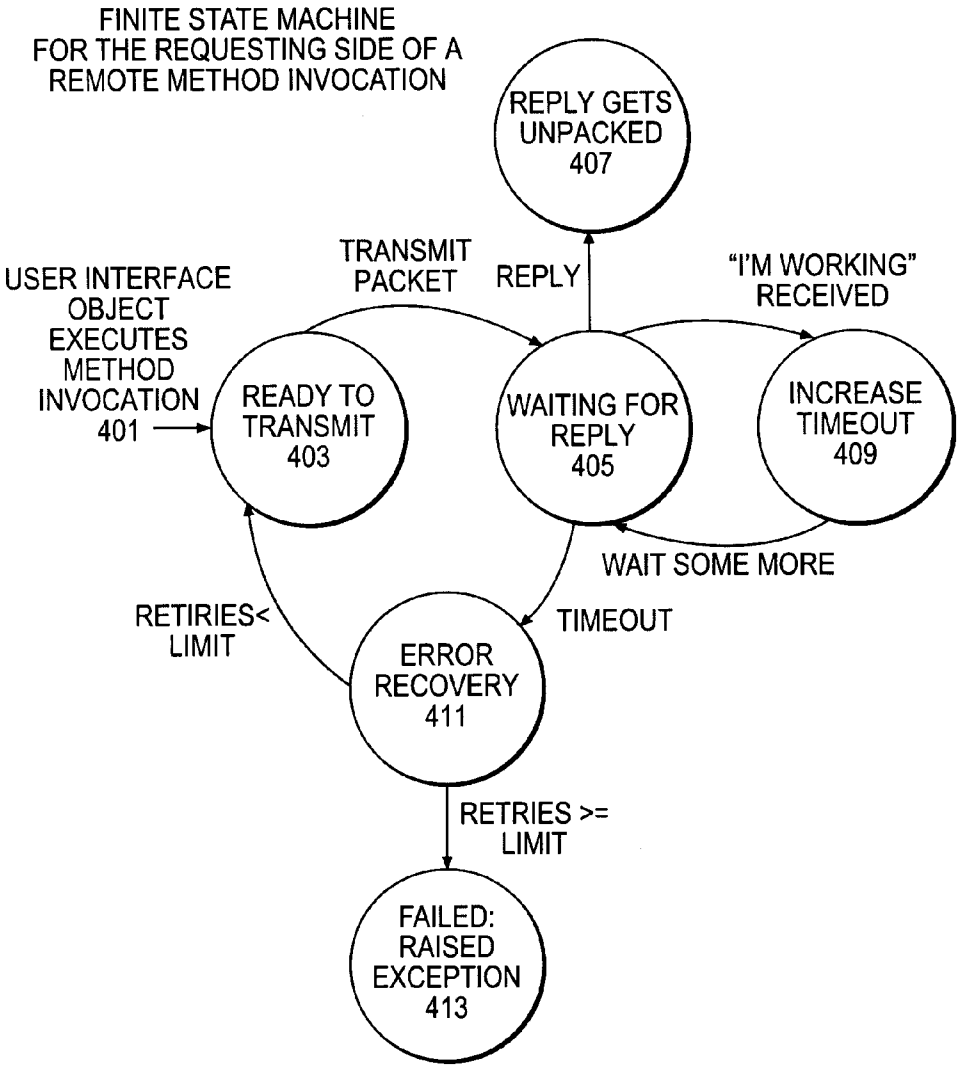

FINITE STATE MACHINE
FOR THE REQUESTING SIDE OF A
REMOTE METHOD INVOCATION
REPLY GETS
UNPACKED
407
TRANSMIT
PACKET
REPLY
"I'M WORKING"
RECEIVED
USER INTERFACE
OBJECT
EXECUTES
METHOD
INVOCATION
401
READY TO
TRANSMIT
403
WAITING FOR
REPLY
405
INCREASE
TIMEOUT
409
WAIT SOME MORE
RETIRIES<
LIMIT
TIMEOUT
ERROR
RECOVERY
411
RETRIES >=
LIMIT
FAILED:
RAISED
EXCEPTION
413

FIG. 21A

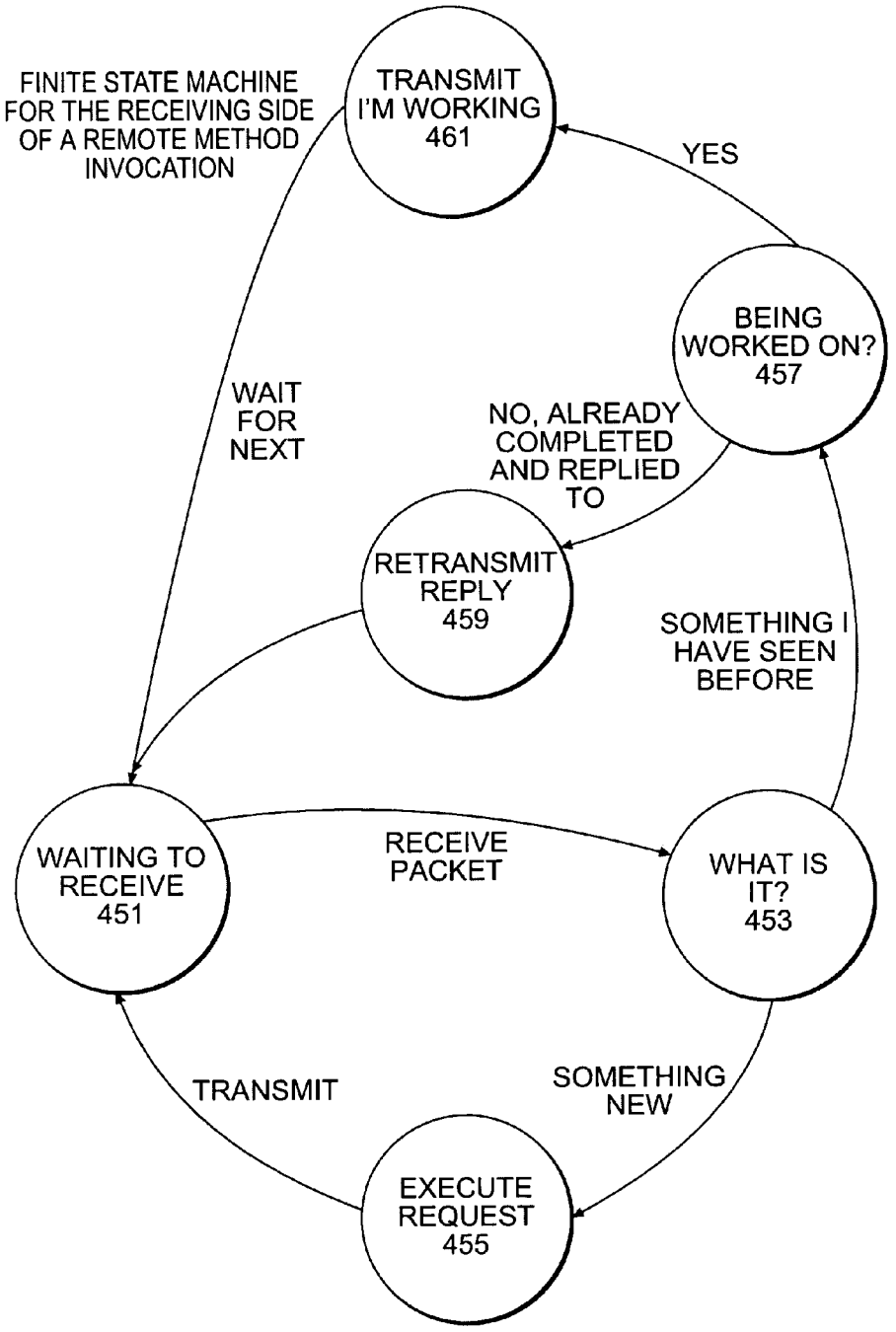

FINITE STATE MACHINE FOR THE RECEIVING SIDE OF A REMOTE METHOD INVOCATION
TRANSMIT I'M WORKING 461
YES
BEING WORKED ON? 457
WAIT FOR NEXT
NO, ALREADY COMPLETED AND REPLIED TO
RETRANSMIT REPLY 459
SOMETHING I HAVE SEEN BEFORE
WAITING TO RECEIVE 451
RECEIVE PACKET
WHAT IS IT? 453
TRANSMIT
SOMETHING NEW
EXECUTE REQUEST 455

FIG. 21B

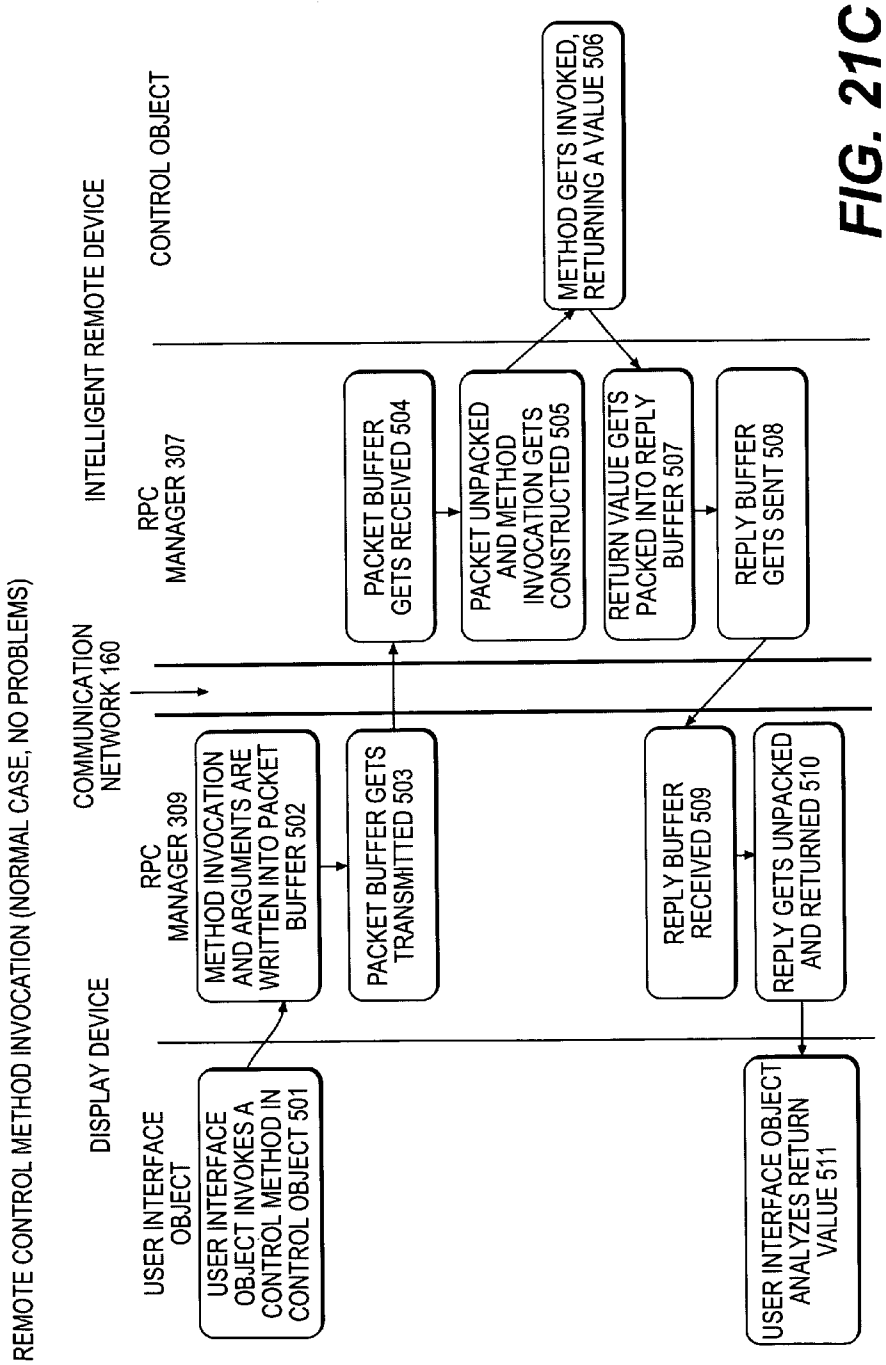

REMOTE CONTROL METHOD INVOCATION (NORMAL CASE, NO PROBLEMS)
DISPLAY DEVICE
COMMUNICATION NETWORK 160
INTELLIGENT REMOTE DEVICE
USER INTERFACE OBJECT
RPC MANAGER 309
RPC MANAGER 307
CONTROL OBJECT
USER INTERFACE OBJECT INVOKES A CONTROL METHOD IN CONTROL OBJECT 501
METHOD INVOCATION AND ARGUMENTS ARE WRITTEN INTO PACKET BUFFER 502
PACKET BUFFER GETS TRANSMITTED 503
PACKET BUFFER GETS RECEIVED 504
PACKET UNPACKED AND METHOD INVOCATION GETS CONSTRUCTED 505
METHOD GETS INVOKED, RETURNING A VALUE 506
RETURN VALUE GETS PACKED INTO REPLY BUFFER 507
REPLY BUFFER GETS SENT 508
REPLY BUFFER RECEIVED 509
REPLY GETS UNPACKED AND RETURNED 510
USER INTERFACE OBJECT ANALYZES RETURN VALUE 511

FIG. 21C

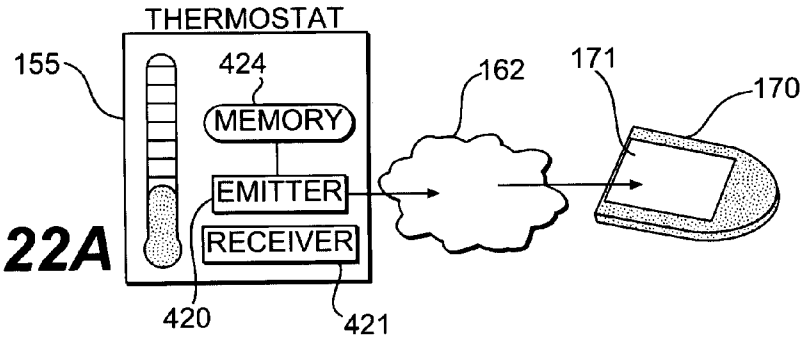


FIG. 22A

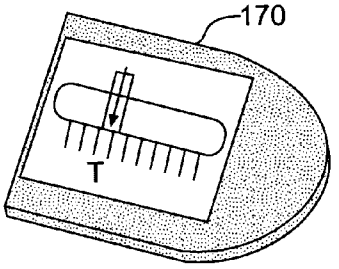


FIG. 22B

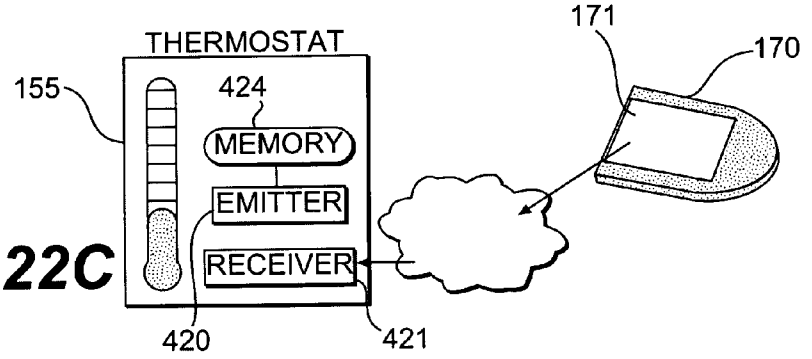


FIG. 22C

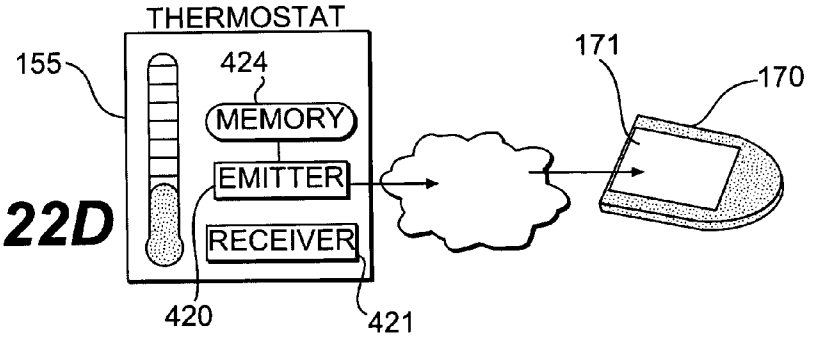


FIG. 22D

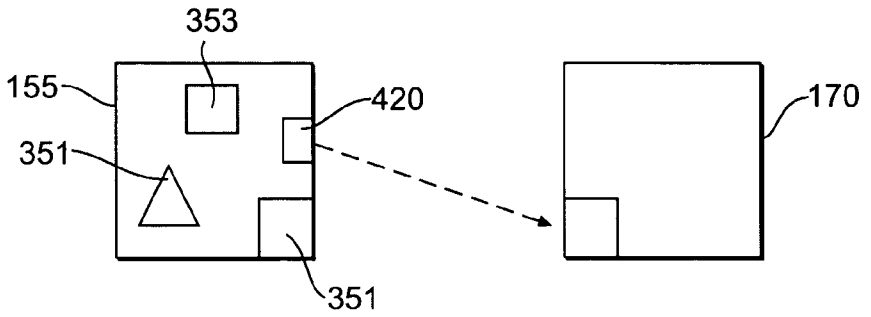


FIG. 23A

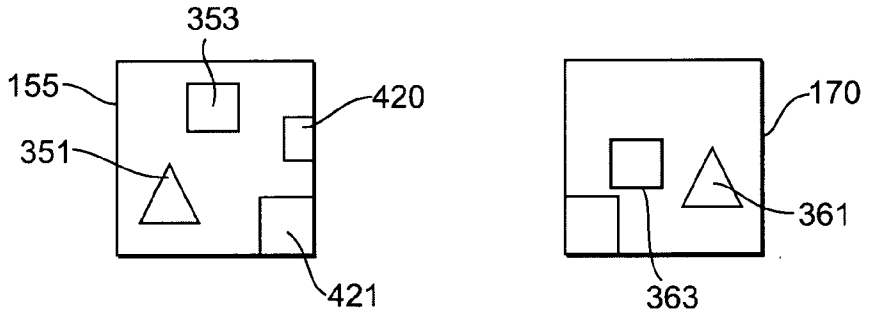


FIG. 23B

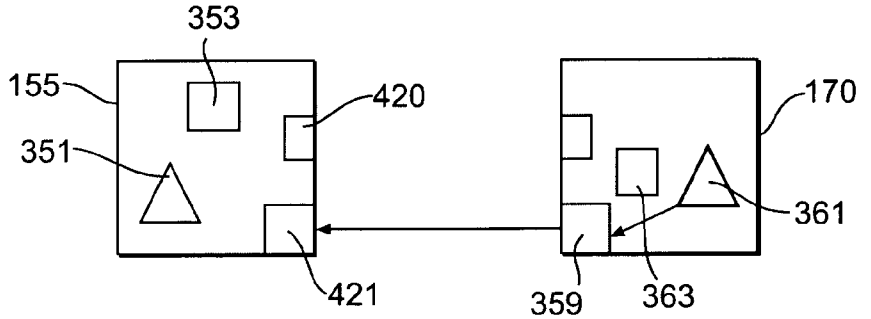


FIG. 23C

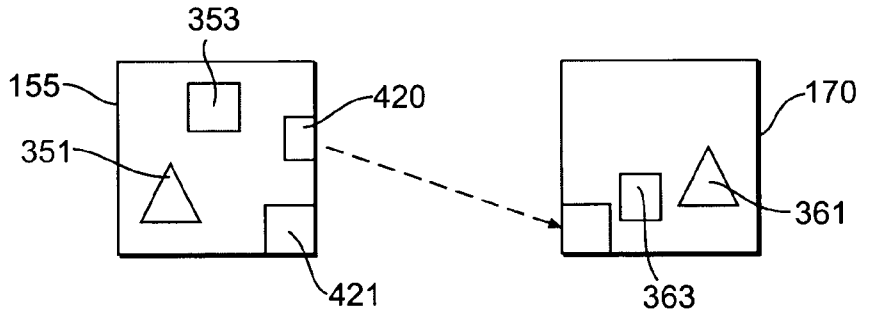


FIG. 23D

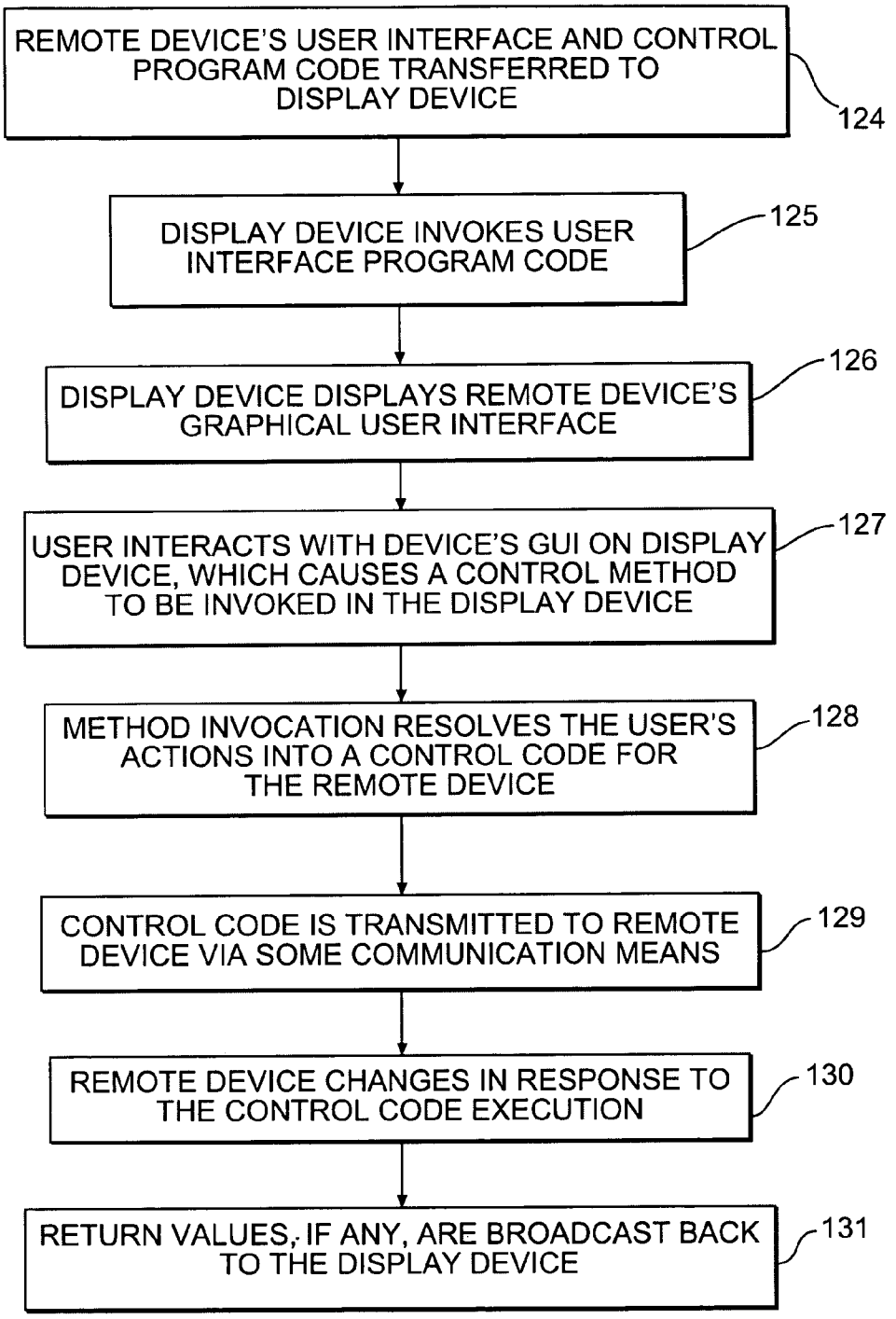

REMOTE DEVICE'S USER INTERFACE AND CONTROL PROGRAM CODE TRANSFERRED TO DISPLAY DEVICE
124
DISPLAY DEVICE INVOKES USER INTERFACE PROGRAM CODE
125
DISPLAY DEVICE DISPLAYS REMOTE DEVICE'S GRAPHICAL USER INTERFACE
126
USER INTERACTS WITH DEVICE'S GUI ON DISPLAY DEVICE, WHICH CAUSES A CONTROL METHOD TO BE INVOKED IN THE DISPLAY DEVICE
127
METHOD INVOCATION RESOLVES THE USER'S ACTIONS INTO A CONTROL CODE FOR THE REMOTE DEVICE
128
CONTROL CODE IS TRANSMITTED TO REMOTE DEVICE VIA SOME COMMUNICATION MEANS
129
REMOTE DEVICE CHANGES IN RESPONSE TO THE CONTROL CODE EXECUTION
130
RETURN VALUES, IF ANY, ARE BROADCAST BACK TO THE DISPLAY DEVICE
131

FIG. 24

GRAPHICAL USER INTERFACE FOR DISPLAYING AND NAVIGATING IN A DIRECTED GRAPH STRUCTURE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of graphical user interfaces for computer systems and, more particularly, to a method and apparatus for providing an intuitive methodology for an operator to control remote devices with a computer controlled object-oriented user interface utilizing animated graphic images.

2. Art Background

Most popular graphical user interfaces that are available today, such as the Xerox Star®, Apple Macintosh®, and Microsoft Windows graphical user interfaces, are based upon the "desktop metaphor." In a desktop metaphor system, the display screen is treated as a virtual desktop. Graphical symbols placed on the virtual desktop (i.e. the display screen) are used to represent common objects found in an office environment such as files, file folders, and printers. These desktop metaphor systems have several limitations which hamper the effectiveness of the man-machine interface between the human operator and the computer system.

One problem with the existing desktop metaphor graphical user interfaces is the use of a hierarchical file system. Most desktop metaphor graphical user interfaces are based upon a computer having a file system in which the computer files are organized in a hierarchical structure. The hierarchical file system structure has a root directory and several sub-directories within the root directory. Each sub-directory may have sub-sub-directories, and so on. The hierarchical file system structure is graphically illustrated on the screen as some type of root file cabinet with file folders inside. The Microsoft Windows and Apple Macintosh® operating systems both use hierarchical file systems illustrated as file folders.

The hierarchical structure of file folders is often very confusing to new users. For example, many users get lost while navigating through hierarchical file systems because the system of keeping files in file folders, which are in other file folders nested to an arbitrary depth is inherently confusing. Users become lost because each file folder looks like every other file folder, and the user is provided with little feedback as to his location within the file system hierarchy. Also, each file folder provides no information as to what the file folder contains other than a user-defined label. For example, except for a short text label, which may have been defined by another user, the user generally is not able to determine whether an open folder is an accounting file folder containing spreadsheet data, or a data base file folder containing a customer data base. Therefore, navigating through the complex file system hierarchy is difficult since the current desktop metaphor graphical user interface offers little feedback as to a user's location within the file hierarchy.

In current graphical user interface systems based upon the desktop metaphor, certain applications often do not fit well into a desktop metaphor environment conceptually. Specifically, non-office related applications may be inappropriate for a system based upon the desktop metaphor. For example, in a desktop metaphor, an application that controls a television would have to be placed into a "paper" file folder on the desktop.

Another common problem with desktop metaphor user interfaces is that displaying a large number of applications

on the desktop results in a cluttered desktop. Many users of desktop metaphor systems keep large numbers of applications and files, usually in icon form, on the display screen. Some attempt is made to use a graphical representation that identifies the type of application or file. However, when a user wishes to open a particular application or file, the user must search the display screen to find the desired icon just as if the user were trying to find a particular paper file on a messy desktop. Icons placed the desktop have no apparent hierarchy, and are all equal. This equality may confuse a user, and may create problems when the user wants to switch from one task to another.

Several systems have evolved to improve the organization of the desktop metaphor graphical user interface. One system is the concept of "Rooms". See, Patrick Chan, "*Learning Considerations in User Interface Design. The Room Model*", Report CS-84-16, University of Waterloo Computer Science Department, Ontario, Canada, July, 1984; Brenda Laurel, "Computers as Theatre", Addison-Wesley Publishing Co., 1991; Brenda Laurel, "The Art of Human-Computer Interface Design", Addison-Wesley Publishing Co., 1990; U.S. Pat. No. 5,072,412, Henderson, Jr. et al., "*User Interface with Multiple Workspaces for Sharing Display System Objects*", and the references cited therein, and also see, Jeffrey J. Blatt, "A Primer on-User Interface Software Patents", Vol. 9, No. 4, The Computer Lawyer (1992).

In systems based upon the Rooms concept, the user creates different rooms where the user performs a single task or a group of related tasks. Each room represents an individual virtual desktop with a group of associated tasks. Each room is given a label that identifies the room with the room's functions. For example, a user may create a mail room, a typing room with a word processor, and an accounting room with a spreadsheet. Each room has only those icons and windows that are associated with the particular task of the room, as well as any windows shared by all rooms, such as a clock.

To switch between tasks in a system that uses the Rooms concept, a user moves between the different rooms that are associated with the different tasks. For example, to go from a spreadsheet to retrieve email, a user moves from the accounting room to the mail room. To move between the two rooms, a user selects an icon that represents a door connecting the current room to the other room. Users can create doors between any of the available rooms.

Although the Rooms concept helps solve the problem of the "cluttered desktop," the Rooms concept still relies upon the common desktop metaphor. For example, the Rooms interface concept does not help simplify the confusing hierarchical file structure used to store information. The Rooms concept also does little to solve the form and function inconsistencies that are apparent when applications which do not fit into the desktop metaphor are added to the system.

In prior art graphical user interfaces, each time a new piece of hardware is introduced to the system a new piece of control software must also be added by the user. For example, if a user of an Apple® Macintosh® or a user of Microsoft® Windows desires to add a new printer to the system, the user must also install a new printer driver to control the added printer. Therefore, each time a user desires to add a new device to the system, the user must also follow a complicated procedure to add the appropriate software.

As will be described, the present invention provides an improved graphical user interface that overcomes several of the problems associated with desktop metaphor systems. A

method and apparatus for obtaining the user interface of a remote device without requiring the user to load the related software is also disclosed. The remote device can be controlled using the remote device's user interface as displayed on the present invention's display screen. Although the user interface is described with reference to the control of remote devices, it will be appreciated from the discussion that follows that the user interface may be applied to a variety of applications in computer workstations, portable computers, and hand held control systems.

SUMMARY OF THE INVENTION

The present invention provides an intuitive graphical user interface and a method and apparatus for controlling remote devices. The graphical user interface of the present invention is displayed on a hand-held display device. The hand-held display device comprises at least one central processing unit (CPU) coupled to a graphic display system. In the present embodiment, the graphic display system is touch sensitive to permit the user to input information using a fingertip without the need for additional input devices such as a keyboard. A communications circuit is also coupled to the CPU transmitting information over some communication medium. The communications circuit is used to control and to communicate with remote devices.

The graphical user interface of the present invention creates a world of spaces organized into a geographic map structure that is simple to navigate through since each space provides familiar geographic surroundings. Each space is represented by a background image which provides navigational landmarks that identify where the space is located within the geographic map structure. For example, the background image of a living room space contains items typically found in a living room such as a lamp, a chair, and a table. Therefore, navigation through this "world" is made easier because the user can orient himself by looking around and examining the surroundings just as a person would do in the real world.

Within each space, the user interface provides graphic objects that can be selected and manipulated by the user. The graphic objects are rendered as animated cartoon-like graphic objects that are easily identifiable as real world objects such as televisions and VCRs. There are at least three types of objects that exist in the spaces of the user interface: data objects, buttons, and portals. Data objects are objects that are associated with a particular piece of data. Buttons are objects that are associated with a particular function or action; when a user selects a button, the user interface performs the function or action associated with the button. Portals are objects that are displayed in a first space but are associated with a second space; when a user selects a portal the user passes through the portal and moves to the second space associated with that portal. The user environment is further enhanced by the use of sound, coordinated with every user action and supplementing the visual feedback provided.

To control real world remote devices, many of the graphic objects presented on the screen are associated with remote devices in the real world. To control a remote device, a user selects the graphic object pictured on the display screen associated with the particular remote device. After the user selects a graphic object associated with a remote device, the hand-held display device displays a user interface for controlling the remote device. The user can then interact with the remote device's user interface. To control the remote device, the present invention invokes a device driver method for the remote device. The device driver method may be located within the remote device or within the local hand-held display device, depending on the remote device's sophistication.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features and advantages of the present invention will be apparent to one skilled in the art, in view of the following detailed description in which:

FIG. **1***a* illustrates an overview of the remote control functionality provided by the present invention.

FIG. **1***b* illustrates an overview of the hardware used to implement the hand-held display device of the present invention.

FIG. **1***c* illustrates a block diagram of an intelligent remote device that is controlled by the present invention.

FIG. **1***d* illustrates a block diagram of a simple remote device that is controlled by the present invention.

FIG. **1***e* illustrates a block diagram of conventional electronic device that is controlled remotely by the present invention.

FIG. **2***a* illustrates a structure of different geographic spaces.

FIGS. **2***b*-**2***c* illustrate a sample geographic map structure of spaces used to provide a user interface.

FIG. **3** illustrates a view of the outer-space space of the user interface based upon the geographic map structure in FIGS. **2***b* and **2***c*.

FIG. **4** illustrates a first view of the Earth's surface space in the user interface based upon the geographic map structure in FIGS. **2***b* and **2***c*.

FIG. **5** illustrates a second view of the Earth's surface space in the user interface based upon the geographic map structure in FIGS. **2***b* and **2***c*.

FIG. **6** illustrates a view of the first house space of the user interface based upon the geographic map structure in FIGS. **2***b* and **2***c*.

FIG. **7** illustrates a view of a living room space in the user interface based upon the geographic map structure in FIGS. **2***b* and **2***c*.

FIG. **8** illustrates a view of a television space in the user interface based upon the geographic map structure in FIGS. **2***b* and **2***c*.

FIG. **9** illustrates a view of a television program listing book space in the user interface based upon the geographic map structure in FIGS. **2***b* and **2***c*.

FIG. **10** illustrates a view of a single television program description space of the user interface based upon the geographic map structure in FIGS. **2***b* and **2***c*.

FIG. **11** illustrates a conceptual view of the television program listing book space of FIG. **9**.

FIG. **12** illustrates a view of a plurality of objects which do not fit on a single screen display.

FIG. **13** illustrates a conceptual view of the plurality of objects placed on an object wheel.

FIG. **14** illustrates a conceptual view of the object wheel of FIG. **13** where the object wheel has rotated revealing new objects.

FIG. **15** illustrates the Spin gesture being used to spin an object wheel.

FIG. **16***a* illustrates a Pickup gesture being used to pickup an object off an object wheel.

FIG. **16***b* illustrates a Pickup gesture being used to pickup an object from a space.

FIG. **17***a* illustrates the Agent Portal of the present invention wearing a doctor's uniform to symbolize a diagnostic mode.

FIG. **17***b* illustrates the Agent Portal of the present invention holding a pencil to symbolize a writing mode.

FIG. **17***c* illustrates the Agent Portal of the present invention attempting to alert the user about a recent event.

FIG. **18***a* illustrates an intelligent device transmitting its graphical user interface to a display device.

FIG. **18***b* illustrates a display device displaying the graphical user interface of an intelligent controllable device.

FIG. **18***c* illustrates the display device transmitting user interactions with the user interface to the intelligent device.

FIG. **18***d* illustrates the intelligent device transmitting return values or feedback to the display device.

FIG. **19***a* illustrates a conceptual diagram of the program objects in the intelligent device being sent to the display device.

FIG. **19***b* illustrates a conceptual diagram of the program objects of the intelligent device in the display device.

FIG. **19***c* illustrates a conceptual diagram of the display device invoking a method in an object within the remote intelligent device.

FIG. **19***d* illustrates a conceptual diagram of the method in an object within the remote intelligent device returning status.

FIG. **20** illustrates a flowchart of the transfer of an intelligent device's user interface.

FIG. **21***a* illustrates a state diagram of the RPC manager for a client requesting a remote method invocation.

FIG. **21***b* illustrates a state diagram of the RPC manager for a server handling a remote method invocation.

FIG. **21***c* illustrates the sequence of a remote method invocation.

FIG. **22***a* illustrates the simple device transferring its graphical user interface to a display device.

FIG. **22***b* illustrates a display device displaying the graphical user interface of the simple controllable device.

FIG. **22***c* illustrates the display device transmitting simple control codes to the simple device.

FIG. **22***d* illustrates the simple device transmitting return values or feedback to the display device.

FIG. **23***a* illustrates a conceptual diagram of the program objects in a simple device being sent to the display device.

FIG. **23***b* illustrates a conceptual diagram of the program objects of the simple device in the display device.

FIG. **23***c* illustrates a conceptual diagram of the display device invoking a method in a program object and transmitting control codes to the simple device.

FIG. **23***d* illustrates a conceptual diagram of the remote simple device returning status information.

FIG. **24** illustrates a flowchart of the transfer of a simple device's user interface.

NOTATION AND NOMENCLATURE

The detailed descriptions which follow are presented largely in terms of display images, algorithms, and symbolic representations of operations of data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art.

An algorithm is here, and generally, conceived to be a consistent sequence of steps leading to a desired result. These steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared selected, chosen, modified, and otherwise manipulated. It proves convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, images, terms, numbers, or the like. It should be borne in mind, however, that all of these, as well as similar terms, are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

In the present case, the operations are machine operations performed in conjunction with a human operator. Useful machines for performing the operations of the present invention include general purpose digital computers or other similar devices. In all cases, the distinction between the method operations of operating a computer and the method of computation itself should be kept in mind. The present invention relates to method steps for operating a computer and processing electrical or other physical signals to generate other desired physical signals.

The present invention also relates to apparatus for performing these operations. This apparatus may be specially constructed for the required purposes or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. The algorithms presented herein are not inherently related to any particular computer or other apparatus. In particular, various general purpose machines may be used with programs in accordance with the teachings herein, or it may prove more convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these machines will appear from the description given below. Machines which may perform the functions of the present invention include those manufactured by the Assignee, FirstPerson, Inc., as well as other manufacturers of computer systems, including Sun Microsystems, Inc. of Mountain View, Calif.

CODING DETAILS

No particular programming language has been indicated for carrying out the various procedures described herein. This is due in part to the fact that not all languages that might be mentioned are universally available. Each user of a particular computer will be aware of a language which is most suitable for his immediate purposes. In practice, it has proven useful to substantially implement the present invention in a high level language which is then compiled into machine executable object code. Because the computers and the monitor systems which may be used in practicing the instant invention consist of many diverse elements, no detailed program listing has been provided. It is considered that the operations and other procedures described herein and illustrated in the accompanying drawings are sufficiently disclosed to permit one of ordinary skill to practice the instant invention.

DETAILED DESCRIPTION OF THE INVENTION

A computer based remote control system with an intuitive graphical user interface is disclosed. In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required to practice the present invention. In other instances, well

known circuits, functions, processes, and devices are shown in block diagram and conceptual diagram form in order not to obscure the present invention unnecessarily.

System Overview

FIG. **1***a* illustrates an overview of the computer based remote control system of the present invention. The main component of the present invention consists of a hand-held display device **170** as illustrated in FIG. **1***a*. In the present embodiment, the hand-held display device **170** comprises a computer system with a color graphic touch screen display. The hand-held display device **170** is designed to control any compatible remote device such as thermostat **150**, video cassette recorder **39**, and stereo system **152**. The hand-held display device **170** communicates with the remote devices using a communications medium **160**.

The remote control system of the present invention as illustrated in FIG. **1***a* differs significantly from prior art remote control systems since it is based upon modular object-oriented software. The object-oriented software provides features previously not available in prior art remote control systems. For example, when an operator (sometime referred to herein as a "user") adds a new compatible remote device, such as thermostat **150** to the system, the new remote device transmits information about itself to the hand-held display device **170**. The transmitted information includes a user interface object that can be used to render a graphical user interface for the remote device. The present invention uses the user interface object to display a graphical user interface for the new remote device on the hand-held display device **170**. The transmitted information may also include a device driver object that is used to control the new remote device. To organize all the available user interface objects, the hand-held display device **170** creates a "world" of geographic Spaces that can be easily be navigated using a set of gestures.

This Specification will first disclose the hardware required to construct the hand-held display device **170** and the compatible remote devices of the present invention. The graphical user interface comprising geographic Spaces and objects which is displayed on the hand-held display device **170** will then be described. A set of gestures used by operators to navigate around the user interface of the present invention will also be defined. To combine the user interface elements, a tour through an embodiment of the user interface of the present invention will be provided to illustrate how the navigational gestures are used to interact with the objects displayed on the hand-held display device **170**. Finally, the system of obtaining information from the remote devices is defined, such that the hand-held display device **170** may communicate with and control remote external devices.

Due to the limitations of a written Specification to describe animated events on an interface, the reader is referred to a video tape entitled "Green Project User Interface," submitted by the applicant concurrently with the filing of the application on which this patent is based. In addition it must also be noted that the features of the present invention are illustrated in black and white within the accompanying figures. However, in the presently preferred embodiment, and as described below, objects and features are displayed in color.

Hardware Configuration

The hand-held display device **170** of the present invention comprises a computer system, such as that illustrated in FIG. **1***b*. The computer system **170** of FIG. **1***b* includes a central processing unit (CPU) **30**, a memory unit **33**, and an input/output (I/O) unit **34**. These elements are those typically found in most general purpose computers, and in fact, the hand-held computer system **170** is intended to be representative of a broad category of data processing devices capable of generating graphic displays.

Coupled to the input/output (I/O) unit **34** of computer system **170** is a color graphic screen display **37**. The screen display **37** comprises, in part, a color raster display device such as a cathode ray tube (CRT) monitor or a liquid crystal display (LCD). The screen display **37** must be of sufficient resolution such that screen display **37** can render graphic images.

In the preferred embodiment, the screen display **37** further comprises a touch screen display system. The touch screen display includes a feature that can detect the presence of a finger touching the display screen **37**. Furthermore, the touch screen can identify the precise location of the finger touching the display screen **37** such that the display screen can sense finger gestures made by a user's finger on the display screen **37**. The touch screen display **37** may comprise one of a variety of touch sensitive display screens commercially available on the market.

Alternatively, the hand-held computer system **170** may be constructed with a conventional graphic display system without a touch sensitive display screen. Such a system requires an alternative input system to obtain the user's interactions. FIG. **1***b* illustrates an alternative input device **42**. Although the user interface of the present invention may be used with alternative input devices **42** such as mice, trackballs, and joysticks, the system will be described within this specification as the preferred embodiment with a touch screen display **37**.

In operation, the CPU **30** generates and displays a graphical user interface on touch screen display **37** through the execution of computer code stored in memory **33**. The computer code stored in memory **33** is not disclosed herein since the description of the present invention in this Specification is sufficient for one skilled in the computer art to utilize the teachings of the invention in a variety of computer systems using one of many computer languages.

Coupled to the input/output (I/O) unit **34** of computer system **170**, in FIG. **1***b*, is an external communication circuit **35**. The external communication circuit **35** provides a system for transmitting and receiving information to and from the hand-held computer system **170**. The present invention uses the external communication circuit **35** in computer system **170** to communicate with remote devices such as a video cassette recorder **39** and a thermostat **150**, across communications medium **160**. The communication circuit **35** may use one of many well-known communication technologies such as radio waves, low power, cellular, infrared signals, or data signals carried on existing power lines.

To be controlled by the present hand-held display device **170** of the present invention, a remote device must have at least one method for receiving information transmitted by the hand-held display device **170**. Ideally, the remote devices are intelligent devices that can both transmit and receive information. However, the hand-held display device **170** can control any remote device that fits within any of the three following remote device "types".

The first type of remote device that can be controlled by the present invention is an "intelligent remote device" specifically designed to operate with the hand-held display device **170**. In the present invention, an "intelligent remote device" is defined as a remote device that contains a micro-

processor, a communication system, and a memory. FIG. **1***c* provides a block diagram of the components in an intelligent remote device **150**. As illustrated in FIG. **1***c*, the intelligent remote device **150** comprises a CPU **403**, a memory **404**, a communications circuit **401**, and a device control circuitry **405**.

The hand-held display device **170** of the present invention communicates with an intelligent remote device **150** through the intelligent remote device's communications circuit **401**. The memory **404** stores programs and data used by the CPU **403**, including a user interface object **303** and a device driver object **301**. The CPU **403** in the intelligent remote device invokes methods in the device driver object **301** to control the remote device.

The second type of remote device that can be controlled by the present invention is a "simple remote device" specifically designed to operate with the hand-held display device **170**. In the present invention, a remote device is defined as a "simple remote device" if the remote device does not have a microprocessor or the remote device contains such a primitive processor that the hand-held display device **170** can not invoke methods in a device driver object within the remote device.

FIG. **1***d* provides a block diagram of the components in a typical simple remote device **155**. As illustrated in FIG. **1***d*, the simple remote device **155** comprises an emitter **420**, a receiver **421**, a memory **424**, and device control circuitry **425**. The emitter **420** in the simple remote device **155** is used to transmit the contents of the memory **424** to the hand-held display device **170**. Therefore, the simple remote device **155** transmits both the user interface object **353** and the device driver object **351** in the memory **424**, such that the hand-held display device receives both objects.

To control the simple remote device **155**, the display device **170** invokes a method within the device driver object **351** to translate the user's interactions into simple predefined control codes. The display device **170** then transmits the simple predefined control codes to the simple remote device **155** which receives the control codes though receiver **421** and passes the information to the device control circuitry **425**.

Finally, the third type of remote device that can be controlled by the present invention is a conventional electronic device that can be controlled by transmitted signals such as infrared signals. Therefore, the present invention is backward compatible with most existing devices that can be controlled remotely. FIG. **1***e* illustrates a typical electronic device **157** that can be controlled by coded signals. The device has a receiver **450** for receiving transmitted coded control signals and device control circuitry **455** for responding to the coded control signals. The hand-held display device **170** controls the electronic device **157**, by transmitting coded control signals. The remote electronic device **157** receives the transmitted signals and responds accordingly.

The Display Device User Interface

In order to create a new product that is easy to use, product designers attempt to make the new product operate in a manner similar to an old product that the intended operator already knows how to operate. For example, audio cassette tape recorders have been available for such a long time that most people know how to operate an audio cassette tape recorder. The play, stop, rewind, fast-forward, and record buttons on cassette tape recorders are therefore familiar to most people. Video cassette recorder manufacturers produced easy-to-use video cassette recorders by taking advantage of this common knowledge. The video cassette recorders are easy to use since the video cassette recorders used the same buttons, i.e., the play, stop, rewind, fast-forward, and record buttons, that audio cassette recorders utilize for operation.

To make the hand-held display device **170** easy to use, the present invention displays an intuitive graphical user interface on the hand-held display device **170**. This section defines the graphical user interface which is provided to users on the hand-held display device **170**.

To design a graphical user interface that is easy to navigate, designers should organize the graphical user interface with a structure that most users already understand. Humans live in a three dimensional geographic world. To survive in our three dimensional geographic world people must learn how to navigate around the local geographic area, including such geographic spaces their house, their neighborhood, and their town. In order to do this, each person creates a mental map of geographic spaces of their relevant local world.

The mental maps of geographic spaces people create are de-facto geographic map structures. FIG. **2***a* illustrates a typical geographic map structure. For example, every person on this planet knows that they live on the Earth. Each person also lives in a certain town; yet, each person is aware that other towns exist on the Earth. Each person lives at a particular house on a particular street; yet, each person is aware that there are other streets in the town with other houses on each street. The identification of one particular house on one street within one town on Earth is a naturally occurring geographic map structure as illustrated in FIG. **2***a*.

The geographic map structure can be extended into the house. For example, the living room is one room of many within the house. The television, television program schedule and the video cassette recorder are objects within the living room. Each object can have its own geographic space. The map structure can also be extended to abstract spaces as well as geographic spaces. Each television program listed in a television program schedule has a program description. The program description can be considered a space within the television program schedule. Most people understand both the structure of geographic spaces and the detail that will be found within each geographic space. Therefore the de-facto geographic map structure of spaces represents a structure that is familiar to most people.

The present invention uses the familiar geographic map structure of our three dimensional world in a graphical user interface. The present invention describes a technique of user navigation in an arbitrarily large and complex geographic map structure through graphic background images that depict geographic spaces. Such geographic map structures improve a user's memory of complex data organization. The advantage of including such geographic map structures is that the familiar structures enable a wider range of people to use information storage facilitated by a computer.

FIGS. **2***b* and **2***c* illustrate a sample geographic map structure utilized by the present invention. In the present invention, each space in the geographic map structures of FIGS. **2***b* and **2***c* is represented as a background image on the display screen of hand-held display device such as the one illustrated in FIG. **1***b*. Located within each geographic space are objects that can be selected and manipulated by the user.

User Interface Display Elements

The user interface environment of the present invention consists of two main display elements: "spaces" and selectable "objects." Each area of the geographic map structure of FIGS. **2***b* and **2***c* represents a space. The space provides a geographic reference that informs a user of the user's current location within the geographic map structure. Within each space are one or more "objects." A user may select and manipulate the objects within each space. In the present embodiment of the hand-held display device **170** with touch screen display **37**, a user may select objects by touching the objects displayed on the touch screen display **37**. Pointing devices other than the user's finger and input devices, other than a touch screen, might handle selection and user input in a different manner.

Spaces

Each space in FIGS. **2***b* and **2***c* has an associated background image that contains appropriate imagery for the space. When a user enters a space, the space's background image is rendered on the display screen **37**. Each background image contains landmarks that help the user identify where the user is. For example, the Earth's surface space **32** of FIG. **2***b* is illustrated as the background images in FIG. **4** and FIG. **5**. The Earth's surface background image of FIGS. **4** and **5** resembles a street on the Earth's surface, thereby informing a user that the user is at the Earth's surface space of the map structure. The system renders the background image using desaturated or dimmed colors such that the background image can easily be distinguished from Selectable objects. The background image for a space may be larger than the display screen **37**; therefore, the present invention provides a gesture for "panning" such that a user may move around within a space as will be explained later.

Objects

Objects are individual graphical icons that are rendered within the background images of spaces. Each object is actually an object-oriented programming type of object containing data methods which can be invoked to act on the data. A user can select and manipulate any of the objects which appear within the background image of a space. Each object is brightly colored and outlined in black. The bright colors distinguish the objects in a space from the space's background image that is drawn using desaturated colors. When a user points to an object, the user interface generates a "Selectable Object" sound that informs the user that the object can be selected and manipulated. Referring again to FIG. **4**, in the present example, the houses **47** and **48** are objects that can be selected and manipulated by a user.

An "anticipation animation" is used in the present invention to inform the user when some action is taking place. The anticipation animation is a short animation displayed on the display screen **37** that is generated by the CPU **30**. In most cases, the anticipation animation consists of enlarging an object's icon such that the icon appears to move closer to the user when the object is selected.

For example, referring to FIG. **3**, when the Earth portal **100** Is selected, the Earth portal **100** expands such that it appears to move closer to the user. After the Earth portal **100** anticipates, the user interfaces move the user to the Earth's surface space **32** that is associated with the Earth portal **100** as illustrated in FIG. **4**.

When a user selects an object by pointing to the object for a predetermined minimum period of time, a sound is generated, the object anticipates, and the CPU invokes a method within the object which performs an action associated with the object.

There are at least three types of objects that exist in the user interface of the present embodiment: data objects, buttons, and portals. The simplest object is a data object. Data objects are objects associated with a set of data. Buttons are objects that provide a means for accomplishing functions. Portals provide a method for moving between different spaces. Each type of object will be explained individually below.

Data Objects

A data object is a simple object used to represent some piece of information. For example, a data object may represent a color or a television channel. The data objects have no methods that can be invoked to perform an associated action or function when the data object is selected. However, a data object can be dropped onto other objects that use the information in the data object. For example, a television program listing data object can be dropped onto a video cassette recorder object to program the video cassette recorder to record the television program.

Buttons

A button object is a special object that has an associated action or function. When a user selects a button, the CPU **30** invokes the action method in the button object to cause the execution of the action or function associated the button. When the button is selected, the CPU also anticipates the button and generates a "button action" sound providing feedback to the user that the button has been selected. Certain buttons provide additional visual feedback to inform the user of the results of the method invoked by the CPU to perform the action associated with the button.

For example, referring to FIG. **8**, a television power "on" button **61** and a television power "off" button **62** are displayed. When a user selects the television power "on" button **61**, the television power "on" button **61** anticipates and the CPU **30** generates the "button action" sound. The CPU **30** also invokes a method in the button object that transmits the necessary information to turn on a real television set associated with the television control panel illustrated in FIG. **8**.

If the real television set is an "intelligent remote device", the television set may return a result value to the display device informing the user whether or not the television was turned on. The user interface may then light up the television power "on" button **61** and generate a "hum" sound to inform the user that the real television was actually turned on.

Buttons can be copied and moved to different spaces so that a user can customize the "world" of the user interface. For example, the television power "on" **61** and the television power "off" **62** buttons could be copied, and the button copies moved from the television control space to any other space. A user may use the copies of the television power "on" **61** and the television power "off" **62** buttons in the new space to control the television.

Portals

Portals are objects that provide a simple method of moving from the current space to another space associated with the portal. A user moves from the current space to another space associated with a portal by simply Selecting the portal. After a user selects a portal, the portal anticipates, generates a "movement through portal sound," and the CPU invokes a method that moves the user to the space associated with the portal.

For example, referring to the geographic map of FIG. **2***b*, to move from the Earth's surface space **32** to the house space **50**, a user simply selects the house portal **48**. Referring to the Earth's surface space **32** as illustrated in FIG. **4**, a user performs the move by selecting the house portal **48** within the Earth's surface space **32**. After being selected, the house portal **48** anticipates, generates a "movement through portal sound," and then the user is moved to the house space **50** as illustrated in FIG. **6**.

Portals can be duplicated and moved such that a user can customize the "world" of the user interface. For example, in FIG. **7** a television program schedule portal **71** is illustrated within the living room space **60**. The TV program schedule portal **71** is associated with a space that contains TV program schedule information. A user can make a copy of the television program schedule portal **71** and move the copy to another space where the user would like to be able to access the television program schedule. The user can then use the copy of the television program schedule portal to access the same information that could be accessed using the original television program schedule portal. Thus, a user would be able to access the same information from different spaces.

There are two special portals that are almost always available to an operator of the user interface: a wayback portal and an Agent portal.

The Wayback Portal

The wayback portal provides a way for a user to trace back a path that the user has traveled along between different spaces. The wayback portal **45** consists of a small rectangular frame that encloses a snapshot of the previous space along the path of spaces that the user traveled along. Since the background image for a space can be larger than what can be seen in a single view, the wayback portal shows a snapshot of the view of the previous space exactly as it appeared when the user went through a portal in that previous space. In the present embodiment, the wayback portal is available to the user in every space in the lower right-hand corner of the display screen. If desired, the user can move the wayback portal to a new location on the display screen such that the wayback portal will then appear at the new location on the screen in every space.

To move one step backward along the current path, the user Selects the wayback portal **45**. When Selected, the CPU **30** anticipates the wayback portal **45**, generates a "move through wayback portal" sound, and moves the user to the space pictured within the wayback portal. In the present embodiment, the "move through wayback portal" sound is actually the same sound as the "move through portal" sound except that it is played backwards. By playing the "move through portal" sound backward when moving through the wayback portal, the movement backward along the current path is emphasized to the user.

As users go through portals they leave a trail of invisible markers. Each marker represents a visit to a space. The wayback manages these markers and maintains them in the order that the spaces are visited. The markers and the markers' order define the path that the user has taken through the world of the user interface. Thus, when a user goes though the wayback, the path defined by the markers is retraced one marker at a time. Each time the user goes through the wayback to a space, a marker is removed from those currently managed by the wayback.

For example, referring to geographic map structure illustrated in FIG. **2***b*, if a user starts in the outer-space space **99**, moves to the Earth's surface space **32**, and then moves to the house space **50**; a path will be generated from the outer-space space **99** to the Earth's surface space **32** to the house space **50**. After traversing this path, the display screen **37** will display the house space **50** as depicted in FIG. **6**. In the lower right corner of the house space **50** is the wayback portal **45**. As illustrated in FIG. **6**, the wayback portal **45** displays a miniature snapshot of the previous space along the space path. In this case, the previous space along the space path is the Earth's surface space **32**; therefore, the wayback portal **45** displays a miniature snapshot of the Earth's surface space **32** as it appeared when the user went through the house portal. If the user then Selects the wayback portal **45**, the wayback portal **45** anticipates, generates the "move through wayback portal" sound, and then moves the user to the Earth's surface space **32** as depicted in FIG. **4**. At the Earth's surface space **32** as depicted in FIG. **4** the wayback now displays a miniature snapshot of the outer-space space **99**, which was the first space along the path. The wayback portal **45** therefore provides a "way back" along the path of spaces the user has traveled.

The present invention implements the wayback portal **45** as a stack of data structures, wherein each data structure contains a pointer and a pair of coordinates. The pointer in each data structure on the stack points to a space that the user has previously visited. The pair of coordinates in each data structure on the stack specifies exactly where the user was within the space pointed to by the pointer when the user left the space through a portal. Each time the user moves through a portal from a particular space, the user interface "pushes" a data structure, containing the pointer to the space and the coordinate pair which defines exactly where the user was within the space when he went through the portal, onto the wayback stack. The wayback portal **45** displays a miniature snapshot of the space that is pointed to by the pointer in the data structure on the top of the stack. To trace back along the path of spaces, each time the user Selects the wayback portal **45**, the user interface "pops" a data structure off the stack and moves to the space indicated by the pointer in the data structure.

The Agent Portal

The Agent is an anthropomorphic character that aids the user in navigating around the user interface and in dealing with events that occur in the "world" created by the user interface. Referring to the Earth's surface space of FIG. **4**, the Agent **41** is illustrated in the lower left corner. The Agent **41** carries objects around for the user. This carrying function is represented by a briefcase, which the Agent also carries. A user can access objects that the Agent is carrying by Selecting the Agent portal **41** thereby entering an Agent space associated with the Agent portal **41**. When a user Selects the Agent portal **41**, the Agent anticipates by appearing to pull out the briefcase and open the briefcase up. The CPU then generates the "move through portal" sound and moves the user to the Agent space.

To place an object into the Agent's briefcase, a user picks up the object and then "Drops" the object onto the Agent portal **41**. The object Dropped onto the Agent portal **41** is then always accessible to the user from any space by entering the Agent's space. The Agent is discussed more fully below in the section entitled "The Agent."

Object Wheel

It is often necessary that a user make choices among an large, arbitrary number of objects. Therefore, the user interface must provide both a method of displaying a large number of objects and a method of allowing the user to Select any of the displayed objects.

For example, referring to the television control space **64** illustrated in FIG. **8**, the user interface of the present invention must provide a method of displaying an arbitrary number of television channels for the user to choose from. In any system, screen space is at a premium, so when the user must choose among many possible options, the options must be presented compactly. In the present invention, if there are 500 television channel objects for a user to select from, presenting them all at once would not only take up significant screen space, but the visual context of the list (i.e., why is the user looking at this list in the first place) would be lost. Thus, a method for allowing the user to browse through the available objects is required.

The user interface of the present invention introduces an "object wheel" to provide a method for allowing a user to choose from a large group of objects. In the object wheel, a few immediately accessible objects are displayed in an object wheel on the display screen while the remaining objects in the group are not rendered on the screen. To access the objects not rendered on the screen, the user "Spins" the object wheel to move the previously inaccessible objects into view on the display screen.

Referring to FIG. **12**, a large number of fruit objects are available for a user to choose from, yet only four fruit objects fit horizontally on the display screen **37** at once. To provide the user with access to all the fruit objects available, the present invention puts the strip of fruit objects onto an "object wheel" **38** as displayed in the conceptual diagram of FIG. **13**. A few of the objects on the object wheel **38** are immediately accessible to a user on the display screen **37**. To access the objects that are not displayed on the display screen **37**, the user rotates the object wheel **38**. As the object wheel **38** rotates, new objects are brought into view and other objects disappear from view as illustrated in FIG. **14**. An operator uses the navigational gestures Point, Spin, and Pickup to manipulate the object wheel presented on the touch screen display **37**. These three navigational gestures are explained in the following section.

In order to enhance the user's feeling that a real wheel is being spun, object wheels obey the laws of physics. For example, the object wheels have mass and therefore inertia; object wheels spin faster as the user applies more "force" when spinning the wheel; object wheels respond to friction since a spinning object wheel gradually stops spinning.

While an object wheel is spinning and the objects are moving across the screen, the current embodiment of the user interface generates a "clicking" sound. The "clicking" sound generated by the user interface simulates the sound made by detents on a slot machine wheel which forces the wheel to stop and be aligned. Like a real wheel, the object wheels of the present invention always stop aligned, and never stop in between object choices. A user can stop a spinning object wheel at any time by Pointing to a moving object on the object wheel while the object wheel is spinning.

An object wheel can be aligned vertically or horizontally on the display screen. When placed vertically, the object wheel can be spun both up and down. When placed horizontally, the object wheel can be spun both left and right.

User Interface Navigational Gestures

Navigation and object manipulation in the user interface are performed using a screen pointing device. In the preferred embodiment, the hand-held display device **170** is constructed with a touch screen display **37** and the pointing device used is a user's finger. Alternatively, the user interface may be implemented using any other type of screen pointing device such as a mouse, a trackball, or a joystick. This specification, however, will only describe the preferred embodiment that uses the user's finger as a pointing device on a touch screen.

In the preferred embodiment of the present invention, the hand-held display device **170** comprises, in part, a touch screen display **37** as illustrated in FIG. **1***b*. The touch screen display **37** senses when a finger has been placed on the display screen and identifies where the finger has been placed. Users select objects on the touch screen and manipulate the objects using a set of six navigational gestures: Point, Select, Pan, Spin, Pickup, and Drop. Each navigational gesture used by the user interface is explained individually below. (Since the navigational gesture names are the same as common words, this specification will capitalize all references to the navigational gestures.)

Point

Point is the most simple gesture in the user interface of the present invention. An operator uses the Point gesture to pick objects rendered on the display screen. For example, if the user interface of the present invention requests the user to choose one of many objects rendered on the display screen, the user Points to an object to choose the object.

A user Points to an object by simply placing a finger on the object displayed on the touch screen. When a user has successfully Pointed to an object, the user interface generates a "Selectable Object" sound. The Selectable object sound provides feedback to the user indicating that the user has successfully Pointed to the object.

Select

The Select gesture is very similar to the Point gesture. An operator uses the Select gesture to have an object invoke a method to perform an function or action associated with the object. For example, a Selected portal invokes a method that moves the user to the space associated with the portal and a Selected button invokes a method that performs the action or function associated with the button.

To Select an object, a user first Points to an object by touching the object with a finger. If the user continues to Point to the object for a predetermined minimum period of time, the object becomes Selected. After a user Selects an object, the object "anticipates," generates a sound for user feedback, and then a method within the object is invoked. If the object is a button, then the method invoked performs a function associated with the button. If the object is a portal, then the method invoked will move the user to the space associated with that portal.

Alternatively, a user can Select an object on the display screen by quickly Pointing to the object twice in succession. When a user Selects an object by quickly Pointing to the object twice, the object does not anticipate. Instead, the user interface only generates the appropriate feedback sound and then performs the action associated with the object. This provides the user a method of Selecting an object quickly.

Pan

To move around within a space a user uses the Pan navigational gesture. A user Pans though a space by placing a finger on the background image of a space displayed on the touch display screen **37** and sliding the finger in the desired direction of movement. The user interface of the present invention responds by generating a "Pan Active" sound and moving background image of the space and all the objects within the space in the opposite direction of the finger's

movement. By moving the space in the opposite direction, the user interface simulates movement of the users' point of view in the other direction.

FIGS. **4** and **5** illustrate the use of the Pan gesture. If a user places his finger on the background of FIG. **4** and then slides the finger on the touch screen to the right, the view will correspondingly pan to the right by moving the background image of the space and all the objects within the space to the left. FIG. **5** illustrates the result of the Pan right operation where house **48** has moved from the right side of the display screen over to the left side of the display screen.

The background image for each space in the present invention can be arbitrarily large. If the background image of a space has finite size, the user interface of the present invention provides feedback that the furthest extent of the background image has been reached. When a user Pans in a direction all the way to the end of a background image, the Pan gesture will not move the user any further in that direction. To inform the user of this fact, the user interface generates an "end of extent" sound instead of the "Pan Active" sound when a user attempts to Pan past the end of a background image. The "end of extent" sound sounds like a "thud" made when something bumps into a wall. The "end of extent" sound thereby informs the user that the user has hit the end of the background image and can go no further.

Spin

An operator uses the Spin gesture to browse through several objects placed on an object wheel. When a user Spins an object wheel, the visible objects on the object wheel move across the display screen and disappear off one end of the screen and previously invisible objects move onto the display screen at the other end.

A user Spins an object wheel by placing a finger on the object wheel and then moving the finger across the display screen in a direction parallel to the object wheel's movement. FIG. **15** illustrates the Spin gesture. In FIG. **15**, a user first places a finger on the object wheel (as shown by the asterisk **96**). The user then slides the finger across the object wheel (as shown by the arrow **97**). This finger gesture causes the object wheel to Spin in the direction of the finger's movement. The faster the user slides the finger across the object wheel on the display, the faster the object wheel spins. Once the object wheel is spinning, the objects on the object wheel appear at one end of an object wheel, move across the display screen, and then disappear off the other end of the object wheel similar to figures which appear on a slot machine wheel.

Pickup

The Pickup gesture is used to pickup an object located within a space or on an object wheel such that a user can carry the object around the "world" defined by the user interface of the present invention. In the present embodiment, different methods are used to Pickup an object from a space or an object wheel. Each method will be explained individually below. However, it will be appreciated that the same gesture could be used to pickup objects in both situations.

The gesture used to Pickup an object from an object wheel is illustrated in FIG. **16***a*. FIG. **16***a* illustrates an object wheel **91** with a number of fruit objects to choose from. To Pickup one of the fruit objects disposed on the object wheel **91**, a user first places a finger on the object (as designated by the asterisk **92**) and moves the finger in a direction perpendicular to the motion of the object wheel. The movement of the finger in a direction perpendicular to the motion of the object wheel is illustrated by arrow **93**. The perpendicular finger motion "tears" the object off the object wheel such that the user can move the object. When the object comes loose from the object wheel, the user interface generates a "tearing" sound that informs the user that the object has been removed from the object wheel. When an object has been picked up, the object casts a shadow onto the background image of the space and moves the object along with the user's finger.

The gesture used to Pickup an object from a space is illustrated in FIG. **16***b* . FIG. **16***b* illustrates a space with an object in the center. To Pickup the object, a user circles the object with the finger as illustrated by arrow **203** and then touches the object as illustrated by asterisk **201**. When the object comes loose from the space, the user interface generates the tearing sound that informs the user that the object has been picked up. After an object has been picked up from a space by a user, a ghost image of the object remains at the object's original location. The ghost image of the object is drawn in desaturated colors indicating to the user that the ghost object cannot be Selected.

After a user has picked up an object, either from an object wheel or a space, the user interface generates a "carrying object" sound that informs the user that an object is being carried. The user can move the carried object within a space, from space to space, or Drop the carried object onto another object.

Drop

The final gesture used to interact with the user interface of the present invention is Drop. After a user has used the Pickup gesture to pick up an object from a space or from an object wheel, the user must continue to keep the finger pressed against the screen to carry the object. The user moves the carried object by sliding the finger across the touch screen display system. After moving the carried object to a desired location, the user then uses the Drop gesture to place the object at the desired location. To drop the object the user lifts the finger off the touch screen display. When the user Drops an object, the object falls into the background image of a space as if pulled by gravity, and the object no longer casts a shadow.

A user of the present invention uses the Drop gesture to perform a number of different functions. The Drop gesture can be used to copy an object by picking up the object and Dropping the object into a portal such that a copy of the object is created in the space associated with the portal. The Drop gesture can be used to move an object to a new location by picking up the object and Dropping the object at the desired location. The Drop gesture can also be used to provide a data object containing information to another object which uses the information to perform a function. Each different use of the Drop gesture will be explained below.

To make a copy of an object using the Drop gesture, a user must first Pickup the object and then Drop the object onto a portal. An object that has been picked up is carried over the portal by sliding the finger across the screen. Once the carried object is over the portal, the portal becomes highlighted such that it provides feedback to the user signaling that the carried object is over the portal. In the present embodiment, the portal emits a green glow. To Drop the carried object into the portal, the user lifts the finger away from the touch screen display. The object is then displayed such that it passes through the portal with an appropriate animation. During the animation, the CPU **30** generates a "sucked through portal" sound effect. When the object has completely disappeared through the portal, the ghost image

of the object drawn in desaturated colors in the present space changes back into a brightly colored object. This provides visual feedback to the user that the original object remains in the present space even though a copy of the object has been sent through the portal to another space.

Users can move objects either within a single space or from one space to another. To move an object within a single space a user first uses the Pickup gesture to pick up an object, moves the object to the desired new location, and then Drops the object using the Drop gesture. After the object is Dropped, the ghost image of the object merges with the Dropped object at the new location to form a single brightly colored image of the object at the new location. This informs the user that the object has been moved to a new location and that a copy has not been made.

Moving an object to another space is accomplished by carrying the object through a portal to another space. To move an object to another space, a user must first Pickup the object and then carry the object over a portal. Once the carried object is over the portal, the portal becomes highlighted such that it provides feedback to the user signaling that the carried object is over the portal, and the portal emits a green glow. To move the carried object through the portal, the user holds the object over the portal for a predetermined minimum amount of time. The portal eventually anticipates moving the user and the carried object to the space associated with the portal. The ghost image of the carried object drawn in desaturated colors is also moved along, with the user, and the carried object to the destination space. When the user Drops the carried object by lifting his finger off of the touch screen display system, the carried object and the ghost image merge into a single brightly colored image of the object. This provides feedback to the user that the object has been moved to the space and has not been copied.

The Drop gesture can also be used to provide information to an object. A user provides information to an object by Dropping an object containing data (i.e. a data object) onto the object which will use that data.

To provide information to an object, a user must first Pickup a data object that contains the necessary information. The user then carries the data object over the object which can utilize the information. When the carried data object is over the object that can utilize the information, the object that can utilize the information becomes highlighted if the carried data object is of the correct class. By becoming highlighted, the object that can utilize the information provides feedback to the user that the carried data object is of the correct class. To Drop the data object into the object that can utilize the information, the user lifts his finger used to carry the data object away from the touch screen. The data object is then passed through and into the portal with appropriate animation and sound effects. When the data object has completely disappeared into the receiving object which can use the information, the receiving object then performs the appropriate function on the data provided.

For example, the Drop gesture can be used to provide television program information to the video cassette recorder object which is associated with a physical video cassette recorder, such that the video cassette recorder programs itself to record the television program. To program the video cassette recorder, a user first uses the Pickup gesture to pickup a data object containing television program information. The user then carries a television program data object over the video cassette recorder object. The video cassette recorder object then modifies its appearance to indicate it is a "program video cassette recorder" mode. This informs the user that the video cassette recorder "understands" the information that is over it (i.e., the video cassette recorder will be programmed to record the television program if the user Drops the television program data object onto the video cassette recorder).

When the user Drops the television program object into the video cassette recorder, the video cassette recorder object recognizes that the television program data object contains information it can use. The video cassette recorder then extracts the time and channel information from the television program data object such that the video cassette recorder object programs itself to record the television program.

A Tour of the Present Embodiment in Operation

To best convey the look and feel of the user interface of the present invention, a "tour" through a sample environment is provided with reference to FIGS. **3-10**. The sample environment explored in the tour is based upon the geographic map structure of spaces diagrammed in FIGS. **2***b* and **2***c*. The tour through the sample environment described herein is not intended to limit the scope of the present invention, but instead to provide an example of one possible embodiment of the present invention. Due to the limitations of a written Specification in describing the present invention's animated user interface, the reader is encouraged to view a video tape entitled "Green Project User Interface," submitted by the applicant concurrently with the filing of the application on which this patent is based.

Referring to FIG. **3**, upon first entering the user interface of the present invention, the user interface initially positions the user in an outer-space space **99**. This space is illustrated in the geographic map structure of spaces diagrammed in FIG. **2***b* . Although, the user is initially placed in the outer-space space **99**, the user interface will later start the user at the last space the user was in when the user quit using the hand-held display device. Referring again to FIG. **3**, at the outer-space space **99**, the user interface renders a brightly colored Earth portal **100** against a dark outer-space background image on the display screen.

Soon after entering the user interface, an outline of a hand **31** appears on the display screen to teach the user how to navigate around the user interface. The hand **31** that appears on the display screen is the hand of the Agent. The Agent's hand **31** extends a finger to touch the screen and then slides the finger across the screen causing the space and objects on the screen display to Pan and the user interface to emit the "Panning active" sound. Referring to FIG. **3**, the outline hand **31** of the Agent is shown moving across the screen. By touching the display screen and moving a finger across the display screen, the Agent's hand **31** demonstrates to a new user how to use the Pan gesture to move around the view of a space.

After demonstrating how to use the Pan gesture to move around in a space, the hand **31** of the Agent next demonstrates how to use the Point and Select gestures. The Agent demonstrates how to use the Point gesture to Point to an object by placing a finger on the Earth portal **100**. After the Agent Points to the Earth portal **100**, the user interface emits the "Selectable Object" sound to provide feedback which informs the user that the Earth portal **100** is a Selectable Object. The Agent then demonstrates how to use the Select gesture to Select an object by continuing to Point to the Earth portal **100**. After the Agent's hand **31** Points to the Earth portal **100** for the predetermined minimum amount of time, the Earth portal **100** anticipates and the CPU generates the "move through portal" sound. The Earth portal **100**

anticipates by expanding rapidly such that the Earth portal **100** appears to move closer to the user. The screen then fades and a new background image appears displaying the Earth's surface space **32** associated with the Earth portal **100**, as illustrated in FIG. **4**.

At the Earth's surface space **32**, the Agent portal **41** appears to "jump" onto the touch screen display **37** and moves to the lower left corner of the display screen. The Agent **41** will always be accessible to the user because the agent appears in every space, except the Agent's space. If desired, the user can move the Agent from the lower left corner to a different location on the display screen. This movement of the Agent will change the Agent's location in every space.

Referring to FIG. **4**, in the lower right corner of the display is the wayback portal **45**. The wayback portal **45** displays a framed snapshot of the outer-space space **99** from FIG. **3** which the user just came from. If the user Selects the wayback portal **45**, the user interface of the present invention moves the user back along the traveled path to the space illustrated within the wayback portal **45**. For example, if the user desires to return to the outer-space space **99** displayed in the wayback portal **45**, the user simply Selects the wayback portal **45**.

The Earth's surface space **32** depicted in FIG. **4** is comprised of a number of houses and community buildings such as stores, banks, schools, etc. Each building in the Earth's surface space **32** represents a portal that can be Selected by the user to move the user to a space associated with the portal. For example, in the Earth's surface space depicted in FIG. **4**, the first house **47** and the second house **48** are portals which can be Selected by the user. If the user Selects either the first house **47** or the second house **48**, the user will be moved to an associated house space.

While within the Earth's surface space depicted in FIG. **4**, the user can move around in the space by using the Pan gesture on the touch screen as demonstrated by the Agent's hand **31** in FIG. **3**. To Pan, the user simply places a finger on the screen and moves the finger in the direction the user wishes to move. The user interface responds to the user's finger motion by generating the "Panning Active" sound and moving the background image and the objects on the screen in the opposite direction of the finger's movement, thereby bringing other objects into the view.

For example, if a user viewing the Earth's surface space **32** as depicted in FIG. **4** wishes to move to the right, the user simply places a finger onto a background area of the Earth's surface space **32** and slides the finger to the right. The user interface reciprocally responds to the user's finger motion by moving the space's background image and the objects in the space to the left. FIG. **5** depicts the view of the Earth's surface space **32** after the user has placed a finger onto a background area of FIG. **4** and slid the finger to the right. A first house portal **47** displayed in FIG. **4** has moved out of the view and is not visible in FIG. **5**. A second house portal **48** displayed on the right side of the view in FIG. **4** has moved to the left side of the view in FIG. **5** and now is only partially visible. A bank portal **49** that was not visible in FIG. **4** has moved into view from the right as illustrated in FIG. **5**. The Agent **41** and wayback portal **45** remain in the view in the same lower left and lower right positions respectively.

When the user Selects any portal from the Earth's surface space **32** as depicted in FIGS. **4** and **5**, the Selected portal anticipates to provide feedback to the user and the user interface moves the user to a space associated with the Selected portal. For example, when a user at the Earth's surface space **32** depicted in FIG. **4** Selects the first house portal **47**, the house portal **47** anticipates and then the user interface moves the user to a house space associated with the house portal **47**. FIG. **6** illustrates a house space **50** associated with the house portal **47** of FIG. **4**.

The house space **50** displayed in FIG. **6** consists of a house with the outer walls removed such that the individual rooms inside the house are visible. All of the rooms inside the house, such as the living room **54**, bedroom **53**, bathroom **55**, attic **52**, and kitchen **56** depicted in FIG. **6**, are portals that can be Selected by the user. As illustrated in FIG. **6**, the wayback portal **45** contains a miniature snapshot of the Earth's surface space **32**, as it appeared when the user left the previous space. Thus the wayback portal **45** depicted in FIG. **6** not only contains a miniature view of the Earth's surface space, but a view of that portion of the Earth's surface space **32** that was in view when the user Selected the house portal **47** of FIG. **4**. To move back up to the Earth's surface space of FIG. **4**, the user Selects the wayback portal **45** that contains a snapshot image of the Earth's surface space.

Referring to FIG. **6**, if the user Selects the living room portal **54**, the living room portal **54** anticipates, the "move through portal" sound is generated, and the user interface moves the user to a living room space associated with the living room portal **54**. Referring to FIG. **7**, an embodiment of a living room space **60** is depicted. Within the living room space **60** of FIG. **7** is a television **73**, a video cassette recorder **75**, and a television program schedule book **71**. The television **73**, video cassette recorder **75**, and television program schedule book **71** are each portals that can be Selected by the user. The television **73** and the video cassette recorder **75** portals are associated with a real television and video cassette recorder respectively.

When a user Selects the television portal **73** displayed within the living room space **60** of FIG. **7**, the television portal **73** anticipates and the user interface displays a graphical user interface associated with the real television. FIG. **8** depicts the television control space **64** which comprises the television's graphical user interface in the present embodiment. The objects available in the television space **64** include a green "on" button **61**, a red "off" button **62**, and several television channel buttons **66-69**. The television channel buttons **66-69** are on a horizontally aligned object wheel **63** such that a user can Spin the object wheel **63** left or right in order to reveal additional television channel buttons that are not currently displayed on the display screen.

The on button **61**, off button **62**, and television channel button **66-69** objects available in the television space **64** are not portals that give access to other spaces in the space map of FIG. **1***b*. Instead, the on button **61**, off button **62**, and television channel buttons **66-69** are buttons that are associated with functions. For example, the green "on" button **61** and red "off" button **62** are associated with the functions "turn the television on" and "turn the television off" respectively. When a user Selects either the on or off button, the button anticipates to provide feedback to the user that the button was Selected. The hand-held display device then transmits the appropriate commands that turn the television on or off. Similarly, if the user Selects any of the television channel buttons **66-69**, the television channel button anticipates and the hand-held display device transmits the appropriate commands to the real television thereby tuning the television to the Selected channel. The process of transmitting commands to the physical television set is explained in detail in the following section.

Referring back to FIG. **7**, if a user Selects the television program schedule book portal **71**, the television program schedule book portal **71** anticipates, and the user interface moves the user to the television program schedule space. FIG. **9** depicts the television program schedule book space of the present embodiment. In the television program schedule book space of FIG. **9**, the user is presented with a two dimensional array of television program listings. The array organizes the television program listings vertically by television channel and horizontally by time. Each television program listing is a portal that can be Selected by the user. Each television program listing can also be carried by the user and Dropped onto the video cassette recorder portal such that the hand-held display device programs the video cassette recorder to record the television program.

The television program schedule book space of FIG. **9** is actually organized as a pair of connected vertically oriented object wheels. Referring to FIG. **11**, a conceptual diagram of the television program schedule book space is illustrated. The television channels of FIG. **9** are placed on a first object wheel **72** as illustrated in FIG. **11**. The television program listings are placed a second object wheel **85**. The two object wheels **72** and **85** are connected such that if the user Spins either object wheel, the other object wheel will spin at the same rate such that the television channels and television program listings are always aligned.

Each television program listing on the television program schedule object wheel **85** is a portal. When a user Selects one of the television program listing portals, the television program listing portal anticipates and the user interface moves the user to a television program information space. FIG. **10** illustrates a typical television program information space. The television program information space includes a clock **81**, a date **82** and television channel **83** which inform the user when the television program is on and what channel the television program is on. The television program information space also includes a description of the television program as illustrated in FIG. **10**.

The Agent

The Agent is an anthropomorphic character designed to help a user of the present invention in a number of ways. The Agent assists users by providing help, providing state information, alerting the user to certain events, and acting as a repository. Each function performed by the Agent will be explained individually.

Help

As previously described in the tour of the present embodiment, with reference to FIG. **3**, when a user first enters the user interface of the present invention, the Agent demonstrates how to use the Point gesture and the Select gesture. Furthermore, the Agent demonstrates how to use the Pan gesture to move around within a space. The Agent can also demonstrate how to use other features of the user interface as necessary.

To further aid users, the Agent carries a help manual in its briefcase that is available to the user at any time. To access the help manual, the user simply Selects the Agent portal. The Agent anticipates by providing an animation of the Agent opening up its briefcase. The user then Selects the help manual from within the Agent's briefcase.

State and Mode Information

The Agent is also used to express the current state or mode of the user interface. To express the current state or mode the Agent changes its clothing, changes its expression, or changes the props it is carrying.

For example, if the user interface of the present invention is placed into a diagnostic mode, the clothing that the Agent is wearing is changed to convey this fact. Referring to FIG. **17***a*, the Agent is shown in a doctor's uniform to symbolize that the user interface is in a diagnostic mode.

FIG. **17***b* provides another example of how the Agent's appearance can convey state information. When the user is in a writing state the Agent is given a pencil to symbolize that the user interface is in a writing state.

Alerts

While a user is using the hand-held display device of the present invention, certain events may occur that require the user to be informed immediately. Such events are usually referred to as alerts. For example, if the hand-held display device is coupled to an electronic messaging service, the hand-held display device should alert the user when a message has been received that is addressed to the user.

In the user interface of the present invention, the Agent is used to obtain the user's attention in order to inform the user about any alerts. To get the user's attention the Agent waves its hand. Additionally, if the alert is very important, the Agent may wave both arms or produce sounds to further attract the user's attention.

For example, when an event occurs that requires the user's attention, the Agent begins waving as illustrated in FIG. **17***c*. To explain what caused the alert, the Agent's appearance may be further altered as demonstrated in the previous section in order to convey information about the specific alert. For example, when an urgent message arrives, the Agent can wave with one hand while holding a telegram in the other hand.

Since alerts are generally more important than state or mode information, alerts will take precedence over state or mode information. Therefore, if the image of the Agent is currently being used to convey state or mode information when an important event occurs, the image of the Agent will change to get the user's attention and inform the user about the alert. For example, if the Agent is holding a pencil while in writing mode, as illustrated in FIG. **17***b*, when an urgent message arrives, the Agent will immediately change to the waving animation as illustrated in FIG. **17***c* to inform the user about the message.

Remote Device Control

The hand-held display device of the present invention is designed to control remote devices. Accordingly, the user interface defined in the previous section for the hand-held display device is designed to be an ideal graphical user interface for controlling remote devices. For example, to control a device in the user's living room, a user first moves to the living room space **60** of the user interface as illustrated in FIG. **7**. The user then Selects an object corresponding to a device that the user wishes to control. For example, if the user desires to control a television, the user Selects the television portal **73** associated with a real television. When the user Selects the television portal **73**, the television portal **73** anticipates and then displays a graphical user interface for controlling the physical television in the user's living room. FIG. **8** illustrates one possible embodiment of a set of television controls for controlling a television. The user then controls the television using the set of television controls displayed on the hand-held display device.

In order for a remote device to be controlled by the hand-held display device, the hand-held display device must have certain information about the remote device. Specifically, the hand-held display device must be provided with an object-oriented programming type of object that is associated with the remote device. The object comprises a plurality of methods including a method for displaying the object on the display screen of the hand-held display device and a method for displaying a graphical user interface for the remote device.

To simplify the operation of the present invention, each compatible remote device automatically transmits the required object-oriented programming type of object to the hand-held display device. When a remote device transmits the object to the hand-held display device, the hand-held display device invokes the method for displaying the object on the display screen. The user can then move the object to an appropriate space. If the user Selects the object, the Selected object anticipates and then the method for displaying the graphical user interface is invoked. The method displays the remote device's graphical user interface on the hand-held display device's touch screen display. The user can then control the remote device by interacting with the remote device's graphical user interface displayed on the touch screen display.

The hand-held display device of the present invention automatically obtains a graphical user interface from two types of compatible remote devices. The first type of compatible remote device is the intelligent remote device with a dedicated computer control system. The second type of compatible remote device is a simple remote device that can only respond to simple coded commands.

The hand-held display device interacts differently with the two types of compatible remote devices. The hand-held display device communicates with intelligent remote devices using peer-to-peer communication with remote method invocation for remote device control. The hand-held display device communicates with simple remote devices using only simple control codes. However, from the user's perspective the differences between the intelligent remote devices and the simple remote devices cannot be seen, and the user's interaction with the graphical user interface is identical for both types of remote devices. The implementation detail of whether a user's actions cause a remote method invocation, or transmission of control codes through some communication mechanism is hidden from the user.

This section will explain how the two types of compatible remote devices transmit information to the hand-held display device. Furthermore, a third type of remote device will explained. The third type of device consists of any electrical device that can be controlled by coded infrared signals.

Within this section, reference will be made to object-oriented programming type objects. Each object-oriented programming type of object contains data and methods that can be invoked to perform functions. To simplify the terminology used, the term "program object" will be used when referring to an object-oriented programming type of object. To eliminate confusion, the objects displayed on the touch screen display in the user interface disclosed in the previous user interface will be referred to as "display objects" in this section.

Peer to Peer Communication with Remote Method Invocation

In the present invention, a remote device is defined as an "intelligent remote device" if that remote device contains a microprocessor, a communication system, and a sufficient amount of memory such that it can invoke methods in a program object. FIG. **1***c* illustrates, in block diagram form, the components in an intelligent remote device. The hand-held display device of the present invention communicates with intelligent remote devices in a peer-to-peer fashion.

The intelligent remote device has an associated graphical user interface comprising touch screen control elements such as dials, switches, sliders, etc. The intelligent remote device's graphical user interface is defined within a user interface program object **303** stored in the intelligent remote device's memory **404**. When an intelligent remote device is first turned on, the intelligent remote device automatically transmits its user interface program object **303** to the hand-held display device. The hand-held display device can then display the user interface for the intelligent remote device as defined by the user interface program object.

When a user interacts with the intelligent remote device's graphical user interface as displayed on the hand-held display device, the hand-held display device invokes methods in a device driver program object **301** within the intelligent remote to control the remote device. Thus the software that actually "controls" an intelligent remote device is located within the intelligent remote device itself.

The procedure of transmitting the user interface object **303** from an intelligent remote device to the hand-held display device will be disclosed with reference to FIGS. **18***a*-**18***d*, **19***a*-**19***d* and the flowchart in FIG. **20**. FIGS. **18***a*-**18***d* illustrate, on a high level, how the intelligent remote device **150** and the hand-held display device **170** of the present invention communicate with each other. FIGS. **19***a*-**19***d* illustrate, in conceptual form, how the user interface program object **303** in the intelligent remote device **150** is transmitted to the hand-held display device **170**.

Referring to step **101** of FIG. **20**, when a new intelligent remote device **150**, such as the intelligent thermostat control device in FIG. **18***a*, is first turned on, the new intelligent remote device **150** broadcasts across the communications network **160** that it has a user interface program object to export. As set forth in the hardware section, the communication on the communications network **160** can take place using radio waves, low power, cellular, infrared signals, or electrical signals on existing power lines in the case of an Echelon® base LON® system. In step **102**, a hand-held display device **170** receives the interface export message. The hand-held display device **170** signals back across the communications network **160** to the new intelligent remote device **150** requesting the new intelligent device **150** to begin transmitting the user interface object **303**.

The intelligent remote device **150** then transmits the user interface object **303** to the display device **170** as specified in step **103** of FIG. **20** and illustrated in FIG. **18***a*. Referring to FIG. **19***a*, the intelligent remote device **150** also transmits a pointer (as indicated by dotted lines) to a device driver program object **301** within the intelligent remote device **150**. The user interface program object **303** defines how the user interface for the intelligent remote device **150** should appear on a display screen. The user interface program object **303** invokes methods in a device driver program object **301** to handle user interface interactions. The device driver program object **301** contains methods that control the actual intelligent remote device **150**. Referring to FIG. **19***b*, the display device **170** now has a copy of the intelligent device's user interface program object **313** and a device driver program object **311** pointer.

After receiving the user interface program object **313**, the hand-held display device **170** invokes a method in the user

interface program object **313** that renders a display object in the current space as stated in step **104**. The new intelligent device **150** also registers with a remote name server connected to the communications network **160**.

When a user Selects the display object associated with the new intelligent remote device **150**, the hand-held display device **170** invokes a method in the user interface program object **313** as stated in step **105** (the object with an invoked method is shown in bold). The method invoked in the user interface program object **313** displays the remote intelligent device's graphical user interface on the display screen **171** of the display device **170** as illustrated in FIG. **18***b* . A user can interact with the remote intelligent device's graphical user interface displayed on the display device **170** as stated in step **106**.

The user controls the intelligent remote device **150** by interacting with the graphical user interface displayed on the hand-held display device **170**. To resolve the user's interactions, the user interface program object **313** requires control methods in the intelligent remote device's device driver program object **301**. To access the control methods in the intelligent remote device's device driver program object **301**, the hand-held display device **170** uses the device driver program object pointer **311**. The device driver program object pointer **311** invokes the methods in the real device driver program object **301**. The intelligent remote device's device driver program object **301** is invoked by sending a message over the communication network **160** as illustrated in FIGS. **18***c* and **19***c* (again, the object with an invoked method is shown in bold). The methods invoked in the intelligent remote device's device driver program object **301** control the intelligent device **150** as stated in step **108**.

Finally, in step **109**, any return values or feedback generated by the device driver program object **301** in intelligent remote device **150** are sent back to the hand-held display device **170** over the communications network **160** as illustrated in FIGS. **18***d* and **19***d*. The user interface program object **313** changes the graphical user interface on the display screen **171** in response to the return values or feedback received from device driver program object **301** in the intelligent remote device **150**.

A robust communication protocol across the communications network **160** is required to communicate remote method invocations between the hand-held display device **170** and the intelligent remote device **150**. Both the hand-held display device **170** requesting the method invocation and the intelligent remote device **150** that executes the method must be able to handle a broken communication path. Therefore, both the hand-held display device **170** and the intelligent remote device **150** have a dedicated remote procedure call (RPC) manager that handles the remote method invocation. Referring to FIGS. **19***a*-**19***d*, the remote procedure call manager **309** transmits the remote method invocation request for the hand-held display device **170**. On the other side, a server remote procedure call manager **307** processes the received remote method invocation in an intelligent remote device **150**.

FIGS. **21***a* and **21***b* illustrate state diagrams used to control the remote procedure call (RPC) managers that handle the remote method invocation on both the hand-held display device **170** side and the remote device side. Taken together, state diagrams in FIGS. **21***a* and **21***b* disclose a handshake used between the hand-held display device **170** and an intelligent remote device **150** to ensure reliable communications.

FIG. **21***a* illustrates a state machine for the remote procedure call manager **309** in the hand-held display device **170** that handles the display device's side of a remote method invocation. In the present invention, the user interface program object **313** in the hand-held display device **170** requests remote method invocations to the device driver program objects **301** located in remote intelligent devices. Referring to FIG. **21***a*, when a user interface object requests a remote method invocation, the hand-held display device's RPC manager enters the "ready to transmit" state **403**. The display device's RPC manager then sets an error countdown timer and transmits a packet containing a message with a method selector and associated data. After transmitting the packet, the display device's RPC manager enters a "waiting for reply" state **405**. If the proper reply message is sent back, the method invocation is complete and the reply message is analyzed at state **407**.

Remote method invocations do not always complete so easily. If the device driver program object **301** in the intelligent remote device **150** needs more time to complete the method, it will send back a message requesting more time. The display device's RPC manager responds by moving to state **409** where it increases the timeout value. If the display device's RPC manager does not receive a reply by the time the error countdown timer expires, the display device's RPC manager moves to an "error recovery" state **411**. If the error countdown timer expires less than a predetermined number of times, the display device's RPC manager returns to the "ready to transmit" state **403**. If the error countdown timer reaches a predetermined limit, the display device's RPC manager abandons the remote method invocation and enters a failed state **413**. The user is then usually informed of the communications problem.

FIG. **21***b* illustrates a state machine for a remote procedure call manager **307** in an intelligent remote device that handles the remote device's side of a remote method invocation. The remote device's RPC manager **307** starts in a "waiting to receive" state **451**. When a packet is received, the remote device's RPC manager analyzes the packet at state **453**. If the received packet contains a new message requesting a method invocation, the remote device's RPC manager gives the packet to an object that performs the method of the request at state **455**. The remote device's RPC manager then transmits a result and returns to the waiting state **451**.

If the received packet contains a message that the remote device has already seen, the RPC manager of the intelligent remote device moves to state **457** where the intelligent remote device determines whether the method invocation request was already completed or whether it is still working on the request. If the intelligent remote device has already completed the request, the subsystem retransmits the results at state **459** and returns to the waiting state **451**. Otherwise, if the intelligent remote device is still working on the method invocation, the remote device's RPC manager moves to state **461** where the remote device's RPC manager requests more time to complete the invoked method and then returns to the waiting state **451**.

FIG. **21***c* illustrates the sequence of a remote method invocation where there are no communication problems. At step **501**, a user interface object in the hand-held display device requests a remote method invocation. Therefore, the user interface object sends a message with a method selector and argument to the display device's RPC manager. The display device's RPC manager packs the message received into a packet at step **502** and transmits the packet across the communications network at step **503**.

The intelligent remote device's RPC manager receives the transmitted packet at step **504**. The remote device's RPC

manager unpacks the message from the packet at step **505** and directs the packet to the device driver program object within the intelligent remote device. The device driver program object invokes the method specified by the method selector in the message at step **506**. The device driver program object generates a return message that is passed back to the remote device's RPC manager. At step **507**, the remote device's RPC manager packs the return message into a reply buffer. The remote device's RPC manager sends the reply buffer at step **508**.

The display device's RPC manager receives the reply buffer at step **509** and unpacks the reply buffer to extract a return value at step **510**. The display device's RPC manager delivers the return value to the requesting user interface program object that invoked the method in the device driver program object. Finally, the user interface program object analyzes the return value at step **511**.

As illustrated, the present invention can control an intelligent remote device by remotely invoking methods in a device driver program object located within the intelligent remote device. In addition, it will be appreciated by the reader that the remote device may also remotely invoke methods in the display device. For example, the device driver object in a remote device may invoke a method in a user interface object in the display device to display a message.

Many remote devices are not sophisticated enough to have internal control methods that can be invoked by the display device. Therefore, the present invention provides other methods for controlling simple devices.

Simple Device Control

In the present invention, a remote device is defined as a "simple device" if the remote device does not have a microprocessor or the remote device contains such a primitive processor that the display device can not invoke control methods within the remote device. FIG. **1***d* illustrates, in block diagram form, the components in a simple remote device. To control a simple device, the hand-held display device of the present invention transmits simple predefined control codes instead of invoking control methods within the simple remote device. Therefore, to control a simple remote device, the hand-held display device must obtain both a user interface program object and a device driver program object from the simple remote device.

An example of obtaining user interface program object and a device driver program object from a simple remote device is given with reference to FIGS. **22***a*-**22***d*, **23***a*-**23***d* and the flowchart in FIG. **24**. FIGS. **22***a*-**22***d* illustrate on a high level how the remote simple device **155** and the hand-held display device **170** of the present invention communicate with each other. FIGS. **23***a*-**23***d* illustrate, in conceptual form, how the user interface program object **353** and the device driver program object **351** of the remote external device **150** are transported the hand-held display device **170** and used.

Referring to FIG. **22***a*, a thermostat implemented as a simple remote device **155** is illustrated. As specified by step **124** of FIG. **24**, the simple remote device **155** of FIG. **22***a* transmits both a user interface program object and a device driver program object to the hand-held display device **170**. Illustrated conceptually in FIG. **23***a*, the simple device **155** transmits the user interface program object **353** and the device driver program object **351**. The transmission of the program objects is done using some communications means **162** that may be a communications network or any other means of transferring the information into the display device **170**.

The hand-held display device **170** receives the copy of the user interface program object **363** that describes the simple device's graphical user interface. The hand-held display device **170** then invokes a method in the user interface program object **363** that renders a display object within a space shown on the hand-held display device **170**. When a user Selects the display object associated with the simple remote device **155**, the display device **170** invokes a method within the user interface program object **363** as stated in step **125** of FIG. **24** and illustrated in FIG. **22***b*. The method invoked in the user interface program object **363** displays a graphical user interface associated with the simple remote device on the display screen **171** of the display device **170**. The user can then interact with simple remote device's graphical user interface on display device **170** as stated in step **127**.

The user's interactions with the simple device's graphical user interface are analyzed by the simple device's device driver program object **361** within the display device **150**. The device driver program object **361** resolves the user's interactions into control codes that the simple remote device **155** will respond to. As illustrated in FIG. **23***c*, the device driver program object **361** located within the display device **170** analyzes the user's actions.

Referring to FIGS. **22***c* and **23***c*, the hand-held display device **170** transmits the control codes to the simple remote device **155**. The simple remote device **155** changes in response to the control codes received as stated in step **130**. Finally, if the simple remote device **155** and the display device **170** are coupled together by a communications network, the simple device **155** can transmit any return values or feedback generated by the simple remote device **155** as illustrated in FIG. **22***d*.

Conventional Device Control

Existing remote devices that are not designed for the hand-held display device of the present invention do not have a user interface program object to transmit to the hand-held display device. For example, FIG. **1***e* illustrates a typical existing remote device that can only respond to coded command signals such as infrared signals. In such cases, the hand-held display device must be manually informed about the remote device. The hand-held display device then uses a generic user interface program object for the type of remote device.

For example, if a user has a conventional television that is controlled by infrared signals, the user enters a model number of the television to inform the hand-held display device about the conventional television. The hand-held display device then accesses a database to obtain a list of the control codes that the conventional television will understands. A television display object is then created to control the conventional television. If the user selects television display object, the hand-held display device will invoke a method that displays a generic television user interface. When the user interacts with the generic television user interface, the display device will transmit the proper infrared control codes to control the television.

SUMMARY

The foregoing has described an intuitive graphical user interface based upon a geographic map structure and a system for controlling remote external electronic devices. In

the user interface of the present invention, each space of the geographic map structure is rendered on a touch screen display as a space. Within each space are colored icons called "objects" that can be Selected and manipulated by the user. Certain objects, referred to as portals, transport the user from one space to another space when Selected. Other objects, referred to as buttons, perform associated actions or functions when Selected. The user interface is displayed on a hand-held display device that can control remote electronic devices. Each remote electronic device transmits a user interface program object that defines a graphical user interface to the display device. The hand-held display device displays the graphical user interface associated with the remote device when a user selects an icon associated with the remote device on the display device's touch screen display. It is contemplated that changes and modifications may be made by one of ordinary skill in the art, to the materials and arrangements of elements of the present invention without departing from the scope of the invention.

We claim:

1. A method of providing an interactive display on a display device in a network that controls a remote device in the network, the method comprising:

selecting a display object on a device display in the network;

generating, at the display device, a request associated with the selected display object;

initializing, at the display device, a communications manager that monitors communications with the remote device;

sending the request to the remote device in the network so that a corresponding remote method is invoked;

receiving, at the display device, a reply from the remote device when the communication manager fails to detect communications errors; and

displaying an interactive graphical interface on the display device when the received reply corresponds to the selected display object.

2. The method of claim **1**, further comprising:

analyzing the received reply to determine whether the reply corresponds to the selected display object.

3. The method of claim **1**, further comprising:

adding time to a timer of the communications manager when the remote device requests additional time for sending the reply.

4. The method of claim **1**, wherein monitoring communications with the remote device comprises:

setting a countdown timer to a predetermined value;

resetting the countdown timer to the predetermined value when the countdown timer expires;

monitoring a number of times the countdown timer expires before the reply is received from the remote device; and

terminating communications with the remote device when the number of times the countdown timer expires reaches a predetermined limit.

5. A method, of providing an interactive display associated with a remote device in a network and based on a selected display object on a display device in the network, the method comprising:

initializing a communications manager of the display device;

sending a request from the controller to the remote device;

processing the request at the remote device so that an associated method is invoked;

returning a reply to the display device based on the invoked method when the communications manager maintains a network connection with the remote device;

displaying, on the display device, a graphical interface associated with the returned reply; and

analyzing the returned reply to determine whether the reply corresponds to the selected display object;

wherein maintaining a network connection comprises:

setting a countdown timer of the communications manager to a predetermined value;

resetting the countdown timer to the predetermined value when the countdown timer expires;

adding additional time to the countdown timer when the remote device requests additional time for sending the reply; and

monitoring a number of times the countdown timer expires before the reply is received from the remote device.

6. A computer readable medium including instructions for directing a display device in a network to perform a method of providing an interactive graphical display based on a selected display object, said instructions comprising:

a module for generating a request associated with the selected display object;

a module for initializing a communication manager so that communication errors are detected;

a module for sending the generated request to a remote device in the network;

a module for receiving a reply from the remote device when no communication errors are detected; and

a module for displaying the interactive graphical display when the reply corresponds to the selected display object.

7. The computer readable medium instructions of claim **6**, further comprising:

a module for analyzing the received reply to determine whether the reply corresponds to the selected display object.

8. The computer readable medium instructions of claim **6**, further comprising:

a module for setting a countdown timer to a predetermined value;

a module for resetting the countdown timer to the predetermined value when the countdown timer expires;

a module for monitoring a number of times the countdown time expires before the reply is received from the remote device; and

a module for terminating communications with the remote device when the number of times the countdown timer expires meets a predetermined limit.

* * * * *