

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
11 October 2001 (11.10.2001)

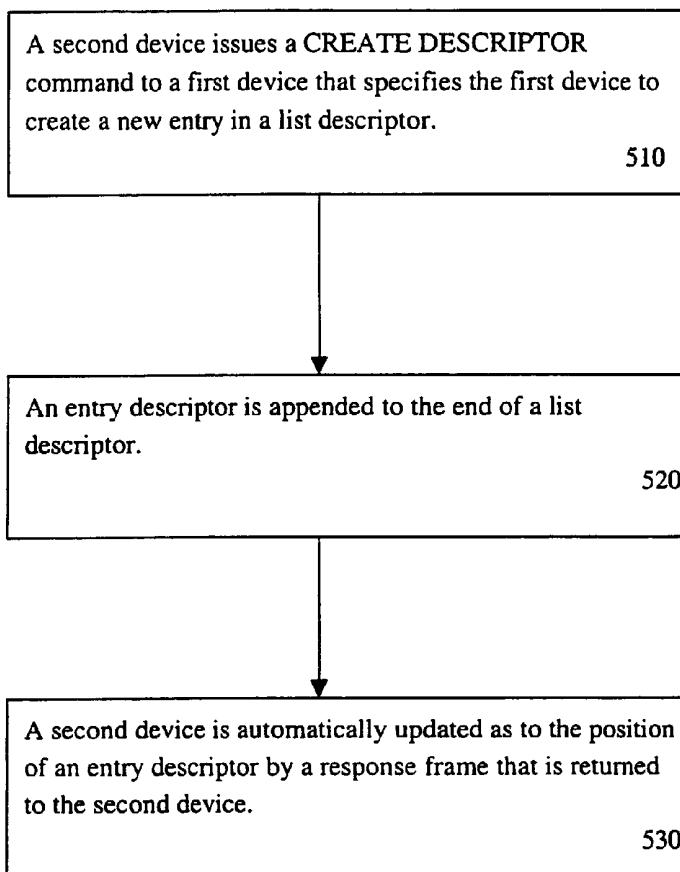
PCT

(10) International Publication Number
WO 01/75611 A1

- (51) International Patent Classification⁷: G06F 13/00
- (21) International Application Number: PCT/US01/10792
- (22) International Filing Date: 2 April 2001 (02.04.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
 - 60/194,238 3 April 2000 (03.04.2000) US
 - 09/693,688 20 October 2000 (20.10.2000) US
- (71) Applicant: SONY ELECTRONICS, INC. [US/US]; 1 Sony Drive, Park Ridge, NJ 07656 (US).
- (72) Inventor: BRELIN, Jon, Ebbe; 267 Watson Drive, #2, Campbell, CA 95008 (US).
- (74) Agents: SOBRINO, Maria, McCormack et al.; Blakely, Sokoloff, Taylor & Zafman, 12400 Wilshire Boulevard, 7th floor, Los Angeles, CA 90025-1026 (US).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: A SYSTEM AND A METHOD FOR CREATING AND ACCESSING DATA



(57) Abstract: In one embodiment, a list descriptor is created (510) in which the list descriptor has a first end and a second end. An entry descriptor is created (520) at the second end of the list descriptor. Another embodiment relates to creating a list in a descriptor. A response frame is returned to a controller in which the response frame includes a list ID for the list.



WO 01/75611 A1



Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

A SYSTEM AND A METHOD FOR CREATING AND ACCESSING DATA

BACKGROUND OF THE INVENTION

This application claims the benefit of the earlier filing date of co-pending provisional application of Jon E. Brelin entitled "*Returning the Position and ID of a Newly Created Descriptor*," Serial No. 60/194,238, filed April 3, 2000 and incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates to creating data in audio, video or audio/video interconnected systems and, more particularly, to a system and method for creating and accessing data in such systems.

BACKGROUND

The conventional method for creating and accessing data in a high speed serial bus such as an Institute of Electrical and Electronics Engineers (IEEE) 1394 standard serial bus, std 1394-1995, Standard For A High Performance Serial Bus, (August 30, 1996) (hereinafter referred to as the "IEEE 1394 standard serial bus") is thought to be inefficient. This inefficiency is revealed by the process involved in creating and then accessing data structures such as entry descriptors, root lists, and child lists.

Definitions for these data structures are provided below followed by a description of the conventional process used to create and access these data structures.

An entry descriptor, typically maintained in a list descriptor, describes information related to a subunit's objects such as a picture from a camera. A root list, a type of list descriptor that is typically located at the beginning of a descriptor hierarchy, serves the purpose of providing a directory, for example, of other list descriptors in a descriptor hierarchy. A child list, a type of list descriptor that is typically subordinate to a parent descriptor, contains information blocks regarding the parent descriptor. For example, a parent descriptor may contain information about a song whereas the child list typically

contains more detailed information about that song such as the author of the lyrics.

To create an entry descriptor, a child list, or a root list, an AV/C controller (“controller”), a device at a serial bus node, sends an AV/C command such as a AV/C CREATE DESCRIPTOR command to a remote AV/C device. The conventional CREATE DESCRIPTOR command, provided below, causes the AV/C device (or subunit) to create (or reject) to create a specified type of descriptor in a certain location of a descriptor hierarchy.

	length	msb					lsb
opcode	1	CREATE DESCRIPTOR (C ₁)					
operand	1	result					
operand	1	subfunction 1					
operand	1	reserved					
operand	v	subfunction_1_specification					
:							
:							

CREATE DESCRIPTOR command frame

As noted above, the CREATE DESCRIPTOR command includes an opcode, the command to create a descriptor, the most significant bits (“msb”), and the least significant bits (“lsb”) of the command. The opcode is used to map the fields to the operation code such as the CREATE DESCRIPTOR command. To create an entry descriptor, an address is generally specified such as FF₁₆ in the AV/C CREATE DESCRIPTOR.

A controller that created the entry descriptor occasionally requires access to the new entry descriptor; however, the position of the entry descriptor is generally unknown to the controller. Therefore, the position of the entry descriptor must be determined by reading and parsing the list descriptor’s number_of_entry_descriptors field. However, reading the list descriptor in order to locate the entry descriptor is inefficient.

Another example of creating and accessing data relates to creating a root list descriptor. A root list descriptor is a type of list descriptor that is generally at the beginning of a descriptor hierarchy of list descriptors. When a root list is created by specifying the root list type (e.g., root_list_type) in a CREATE

DESCRIPTOR command, the target device ("target") creates the root list descriptor containing the root list and updates the subunit identifier descriptor ("SID") with this information. In order for the controller to read the root list, the controller must find and read the SID to determine the new root list ID through root_list_ID field so that the controller may access the new root list. This process is inefficient since the controller must find and read the SID.

Another example of creating and accessing data in an inefficient manner relates to a controller creating a child list by using the CREATE DESCRIPTOR command to create a list type such as through the child_list_type field. The target creates the child list and updates the parent entry by placing the new child list ID (*i.e.*, child_list_ID) into the parent entry. In order to access the child list, the controller must read the parent entry to determine the new child list ID. This process is inefficient since the controller is required to read the parent entry. It is therefore desirable to have a system and a method that overcomes the disadvantages associated with conventional methods.

SUMMARY

A method and apparatus related to creating and accessing data structures on a bus. One embodiment involves creating a list descriptor. An entry descriptor is created and appended to the list descriptor. A response frame is returned to a device such as a controller. The response frame includes the location of the entry descriptor in the list.

Another embodiment relates to creating a root list in a descriptor. A response frame that includes a list ID of the root list is returned to a device such as a controller. This information allows a device to directly access the root list.

In yet another embodiment, a child list is created in a descriptor. The child list ID is returned in a response frame to a device such as a controller. This information allows a device to directly access the child list. Other features and advantages of the invention will be apparent from the accompanying drawings and from the detailed description that follows below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limited in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

Figure 1 is a block diagram of one embodiment for a multimedia network including various consumer electronic devices coupled through a high speed serial bus;

Figure 2 is a block diagram of one embodiment of a controller AV/C device and a target AV/C device with a descriptor hierarchy residing in the target node of the IEEE 1394 standard serial high speed serial bus;

Figure 3 illustrates a flow diagram of one embodiment for creating an entry descriptor;

Figure 4 illustrates one embodiment of a flow diagram for creating a root list; and

Figure 5 illustrates one embodiment of a flow diagram for creating a child list.

DETAILED DESCRIPTION

The invention relates to a process for efficiently creating and accessing entry descriptors, root lists, and child lists. In one embodiment, a system is disclosed in which a device such as a controller is coupled by a bus such as a IEEE 1394 standard serial bus to another device such as a target device that contains shared information.

A list descriptor is created and then an entry descriptor is appended to the list descriptor. A response frame is returned to the controller indicating the position of the entry descriptor. By implementing this process, the controller may easily access the entry descriptor without reading the list descriptor.

In another embodiment, a root list is created in a root list descriptor. A response frame that includes the list identifier (*e.g.* root list ID) is returned to a controller. The list identifier indicates the identifier (ID) of a list such as a root list. By implementing this process, the controller may directly access the root list

by using the root list identifier without finding and reading the subunit identifier descriptor ("SID").

In yet another embodiment, a child list is created in a child list descriptor. A response frame that includes the list ID of the child list is returned to the controller. By implementing this process, the child list may be directly accessed by using the list identifier without reading the parent entry.

In the following description, numerous specific details are set forth to provide a thorough understanding of the invention. However, it will be understood by one of ordinary skill in the art that the invention may be practiced without these specific details. In other instances, well known structures and techniques have not been shown in detail to avoid obscuring the invention.

Figure 1 is a block diagram of one embodiment for multimedia network 100, including various consumer electronic devices coupled through a high speed serial bus 160. The high speed serial bus 160 may be, for example, an IEEE 1394 standard serial bus. Although not limited as such, in one embodiment, the multimedia network 100 may be located in one physical space, such as a home or an office. Multimedia network 100 may include digital video camera ("DVC") 110, digital video monitor 120, personal computer 130, personal digital assistant 135, digital video cassette recorder ("VCR") 140, set top box 142, and printer 145. Computer 130 and personal digital assistant 135 contain processors which execute instructions stored on a media, such as memory (volatile, non-volatile, removable), or storage media, etc. High speed serial bus 160 supports communication of digital audio/video data and computer transmission data between the network devices. DVC 110, digital VCR 140, printer 145, or any other consumer electronic device typically do not have direct Internet access whereas computer 130 and personal digital assistant 135 may be configured to have Internet access. Network 210 that includes networks such as an intranet or a global network such as the Internet, interfaces with computer 130 and personal digital assistant 135 through, for example, a telephone line (not shown) or wireless communication.

Figure 2 is a block diagram in which an entry descriptor, a root list, and a child list, are created through AV/C devices. In this embodiment, two AV/C devices such as computer 130 and DVC 110 of **Figure 1** are shown residing in two nodes of the IEEE 1394 standard serial bus. System 400 includes controller (also referred to herein as a second device) 410 connected to a target device (also referred to herein as a first device) 420.

Target device 420 may be, for example, DVC 410. Target device 420 includes memory 430 for storing data and several descriptors (*e.g.*, data structures) 405, 450, 460, and 470. Each descriptor 405, 450, 460, and 470 has a structured data interface containing information pertaining to features of target device 420 which in this embodiment is DVC 110. Descriptor 460 is a child descriptor. Child descriptors are typically created to the right of its parent descriptor such as descriptor 450. The identification of child descriptor 460 such as the child list ID, is maintained within the descriptor 450.

Figure 2 is a block diagram of one embodiment of a controller AV/C device and a target AV/C device with a descriptor hierarchy residing in the target node of the IEEE 1394 standard serial high speed serial bus. Specifically, **Figure 2** illustrates descriptors such as entry descriptive 470, root list descriptor 450, and child list descriptor 460 created by target device 420. In one embodiment, controller 410 issues a CREATE DESCRIPTOR command to target device 420 to create an entry descriptor. The CREATE DESCRIPTOR command is accepted by target device 420 and causes target device 420 to create entry descriptor 470. Entry descriptor 470, created at the end of a child list descriptor, includes information about a subunit's objects such as a picture from DVC 110. Target device 420 then returns information such as the position of the entry descriptor to controller 410 through a response frame of the CREATE DESCRIPTOR command. The response frame of the CREATE DESCRIPTOR command is automatically returned to the controller that originally sent the CREATE DESCRIPTOR command. Conventional CREATE DESCRIPTOR commands lack such a response frame that contains the position information.

In order to understand the difference between the new CREATE DESCRIPTOR command and the conventional CREATE DESCRIPTOR

command, some of the features of the CREATE DESCRIPTOR command are provided in Table I. Thereafter, additional embodiments are presented using the CREATE DESCRIPTOR command.

Table I - Field values in the CREATE DESCRIPTOR command
ACCEPTED/REJECTED/INTERIM response frames for subfunction_1
00₁₆

Field	Command	Response		
		ACCEPTED	REJECTED	INTERIM
result	FF ₁₆	00 ₁₆	FF ₁₆	FF ₁₆
subfunction_1	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆
descriptor_specifier_where	Descriptor specifier of the SID or an entry	← ¹	←	←
descriptor_specifier_what	Descriptor specifier of the new list or entry	← ²	←	←

¹ When appending an entry using FF, the response frame returns the resultant object position.

² When creating a list, the accepted response frame returns the list ID using descriptor_specifier 10₁₆.

The fields of the CREATE DESCRIPTOR command are presented with several columns: a column for field names, the field values in the command frame, and a response for the various response frames. The field and command columns relate to the values in the command sent by controller 410 to target device 420. The field column of the CREATE DESCRIPTOR command shows the fields that are used such as the result, subfunction_one, descriptor_specifier_where, and descriptor_specifier_what. Descriptor_specifier_where indicates where in the descriptor hierarchy that a descriptor should be created and descriptor_specifier_what indicates the type of descriptor (e.g. entry descriptor) to be created. FF₁₆ may be applied to descriptor_specifier_where and descriptor_specifier_what when appending an entry descriptor to the end of a list. Additionally, a descriptor specifier of the location

of the SID for creating a root list or of an entry for creating a child list may be provided.

Data such as the position of a newly created entry descriptor, or list_ID of the new root list, or child list is added to the response frame in the row specified by the superscripts 1 and 2 of Table I and is returned from target device 420 to controller 410. One difference between the CREATE DESCRIPTOR command that implements the techniques of the invention and conventional CREATE DESCRIPTOR command relates to the response frame which the conventional CREATE DESCRIPTOR command lacks. In the ACCEPTED response frame of the new CREATE DESCRIPTOR command, the descriptor_specifier_where and descriptor_specifier_what fields are updated to contain the new information. The accepted column indicates that the list or the entry descriptor that was created has been accepted by target device 420. The creation of the entry descriptor, for example, is accepted by target device 420 when the entry descriptor is placed in a certain location of the descriptor hierarchy of target device 420 and the position of the entry descriptor is returned in the response frame to controller 410.

Rejected, on the other hand, means that target device 420 has rejected creating an entry descriptor, a root list, or a child list. One reason that a target device may reject a CREATE DESCRIPTOR command is that the target device may be unable to support the CREATE DESCRIPTOR command.

In contrast, the interim column for the CREATE DESCRIPTOR command has been established to indicate to a second device such as a controller that after a time period such as 100 milliseconds (ms), the target device is still trying to execute the CREATE DESCRIPTOR command. The interim column, therefore, indicates that the CREATE DESCRIPTOR command has neither been accepted nor rejected.

Given this explanation of the CREATE DESCRIPTOR command, presented below are methods of creating root and child list descriptors. In one embodiment, a root list is created from the SID by a controller 410 sending a CREATE DESCRIPTOR command to target device 420. The CREATE DESCRIPTOR command includes a response frame that identifies the root list ID

for the root list and indicates that the CREATE DESCRIPTOR has been accepted. The response frame is automatically returned to controller 410. By including the root list ID in the CREATE DESCRIPTOR command frame, controller 410 has the identifier of the root list and is able to directly access the root list.

In yet another embodiment, another CREATE DESCRIPTOR command frame may be issued by controller 410 to target device 420 in which the CREATE DESCRIPTOR command requires a child list be created from an entry in root list 450. The CREATE DESCRIPTOR command includes a response frame. The response frame is updated with the child list ID and the response frame is automatically returned to controller 410. By having access to the child list ID, controller 410 is then able to directly access the child list.

Figure 3 is a flow diagram of one embodiment for creating an entry descriptor. At block 510, a second device such as a controller issues a CREATE DESCRIPTOR command shown in Table I to a first device such as a target device. At block 520, this CREATE DESCRIPTOR command creates an entry descriptor at, for example, the end of a list descriptor in the target device. A response frame is part of the CREATE DESCRIPTOR command. By including the entry's position in the response frame of the CREATE DESCRIPTOR command, the controller is automatically updated as to the position of the new entry descriptor.

Figure 4 illustrates a flow diagram of one embodiment for creating a root list within a descriptor such as a list descriptor. At block 610, a CREATE DESCRIPTOR command is sent by a second device such as a controller to a first device such as a target device. The CREATE DESCRIPTOR command includes a root list type (*i.e.*, a root_list_type) specified in the CREATE DESCRIPTOR command. At block 620, a response frame of the CREATE DESCRIPTOR command returns the list ID (*i.e.*, list_ID) of the new root list from the target device to the controller. This makes it more convenient for the controller to directly access the root list after the command has been issued. At block 630, the controller may directly access the root list.

Figure 5 is a flow diagram of one embodiment for creating a child list. At block 710, a CREATE DESCRIPTOR command includes a child list type (*e.g.*, child

list_type) specified in the CREATE DESCRIPTOR command. The target device creates the child list. The parent's entry is updated with the child list ID by having the new child list ID (*i.e.*, child_list_ID) placed hierarchically below the parent entry. At block 720, the response frame of the CREATE DESCRIPTOR command returns the list ID (*i.e.*, list_ID) of the new child list to the controller. At block 730, the second device such as the controller has access to the position of the child list.

In the preceding detailed description, the invention is described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method comprising:
creating a list descriptor having a first end and a second end;
creating an entry descriptor at the second end of the list descriptor; and
returning a response frame to a device.
2. The method of claim 1, further comprising:
rejecting a first command in a response frame to a controller.
3. The method of claim 1, wherein the response frame includes a position of the list descriptor.
4. The method of claim 3, wherein the position is FF_{16} .
5. The method of claim 3, further comprising:
accessing the entry descriptor by the device.
6. The method of claim 3, wherein the response frame uses a descriptor_specifier 10_{16} .
7. A method comprising:
creating a descriptor;
creating a root list in the descriptor; and
returning a response frame having a list ID of the root list to a device.
8. The method of claim 7, wherein the list ID is returned using a descriptor_specifier 10_{16} .
9. The method of claim 7, further comprising:
accessing the list ID by a controller.
10. A method comprising:
creating a descriptor;

- creating a child list in the descriptor; and
returning a response frame having a list ID of the child list.
11. The method of claim 10, further comprising:
accessing the list ID by a controller.
 12. The method of claim 10, wherein the response frame returns a list ID descriptor to a controller.
 13. A system comprising:
a first device (420);
a second device (410) coupled to the first device (420), said second device (410) to send a first command to the first device (420); and
in response to the first command, said first device (420) creating an entry descriptor comprising at least one list descriptor having a first end and second end, an entry descriptor located at the second end, and sending a response including the entry descriptor to the second device (410).
 14. The system as set forth in claim 13, further comprising a bus, the first and second devices (420, 410) coupled via the bus.
 15. The system as set forth in claim 13, wherein the first command comprises a CREATE DESCRIPTOR command.
 16. The system of claim 15, wherein the position is FF_{16} .
 17. The system of claim 15, wherein the second device accesses the entry descriptor.
 18. The system of claim 15, wherein the response frame uses a descriptor_specifier 10_{16} .
 19. A system comprising:
a first device (420);

a second device (410) coupled to the first device (420), said second device (410) to send a first command to the first device (420); and

in response to the first command, said first device (420) creating a root descriptor comprising at least one root list and sending a response including the list ID to the second device (410).

20. The system as set forth in claim 19, further comprising a bus, the first and second devices (420, 410) coupled via the bus.

21. The system as set forth in claim 19, wherein the first command comprises a CREATE DESCRIPTOR command.

22. A system comprising:

a first device (420);

a second device (410) coupled to the first device (420), said second device (410) to send a first command to the first device (420); and

in response to the first command, said first device (420) creating a child descriptor comprising at least one child list and sending a response including the list ID to the second device (410).

23. The system as set forth in claim 22, further comprising a bus, the first and second devices (420, 410) coupled via the bus.

24. The system as set forth in claim 22, wherein the first command comprises a CREATE DESCRIPTOR command.

25. A machine readable storage media containing executable program instructions which when executed cause a digital processing system to perform a method comprising:

creating a list descriptor having a first end and a second end;

creating an entry descriptor at the second end of the list descriptor; and

returning a response frame to a device.

26. The machine readable media of claim 25, wherein the method further comprises:

rejecting a first command in a response frame to a controller.

27. The machine readable media of claim 25, wherein the method further comprises:

accessing the entry descriptor by the device.

28. A machine readable storage media containing executable program instructions which when executed cause a digital processing system to perform a method comprising:

creating a descriptor;

creating a root list in the descriptor; and

returning a response frame having a list ID of the root list to a device.

29. The machine readable media of claim 25, wherein the method further comprises:

accessing the list ID by a controller.

30. A machine readable storage media containing executable program instructions which when executed cause a digital processing system to perform a method comprising:

creating a descriptor;

creating a child list in the descriptor; and

returning a response frame having a list ID of the child list.

31. The machine readable media of claim 25, wherein the method further comprises:

accessing the list ID by a controller.

1/5

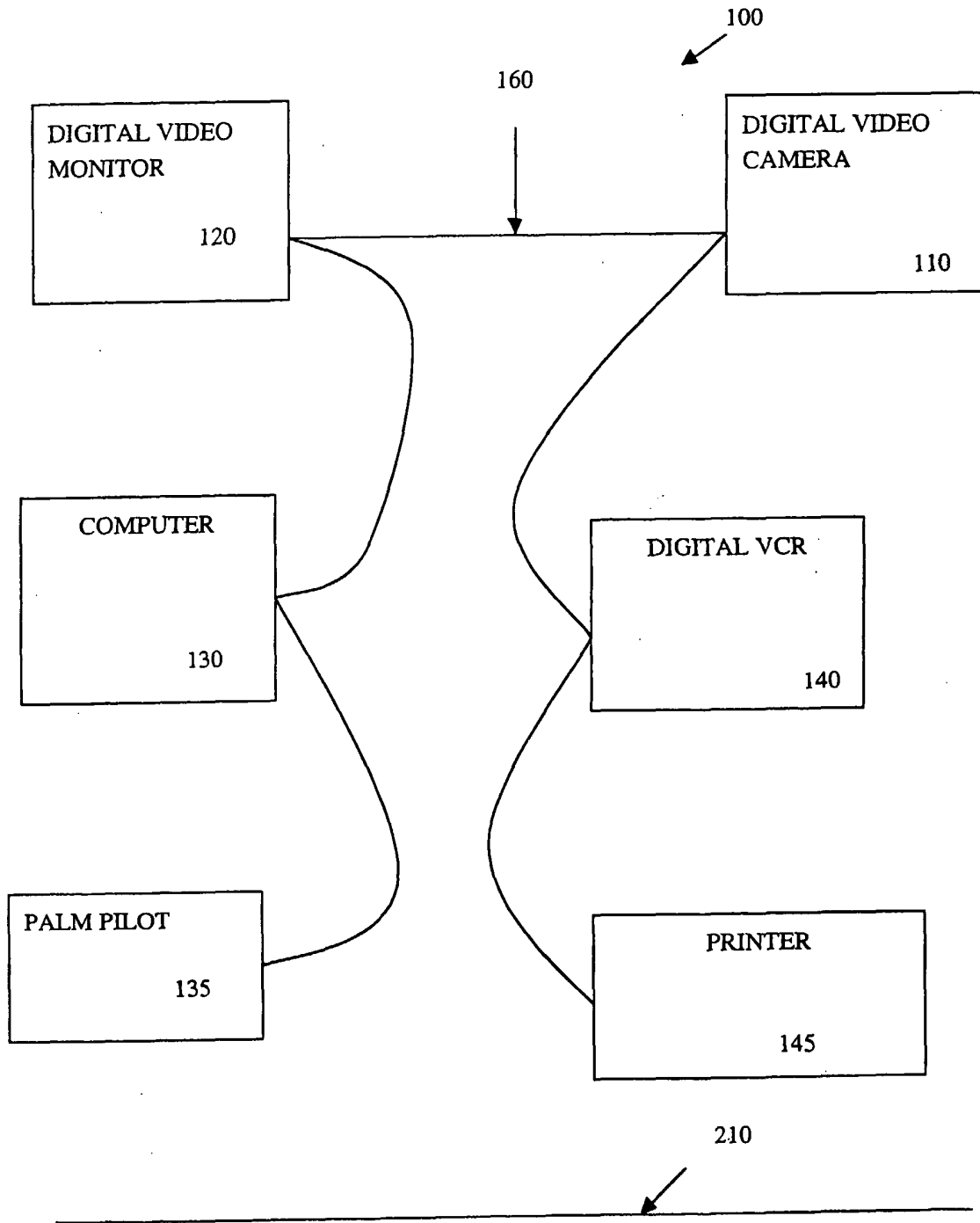


FIGURE 1

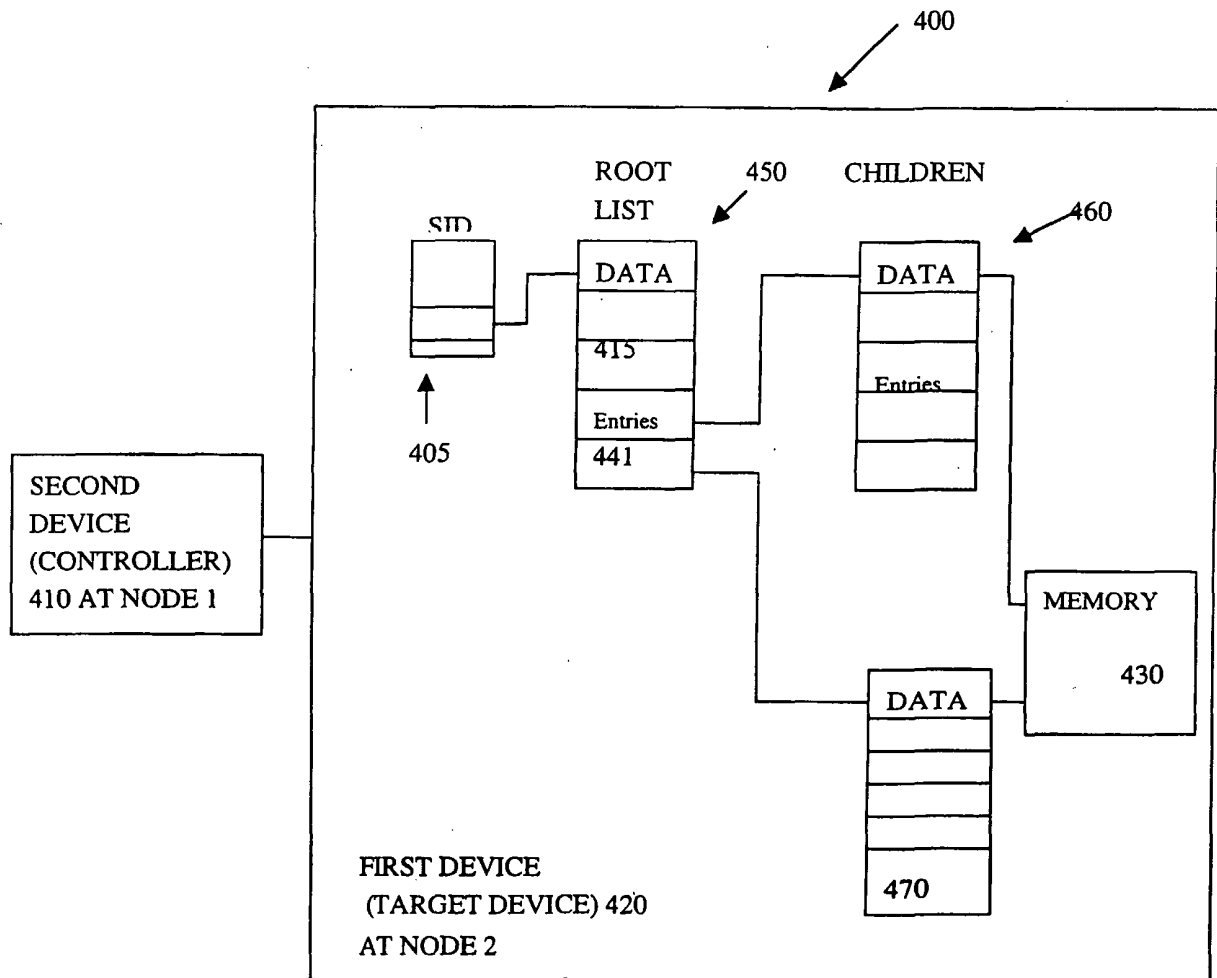
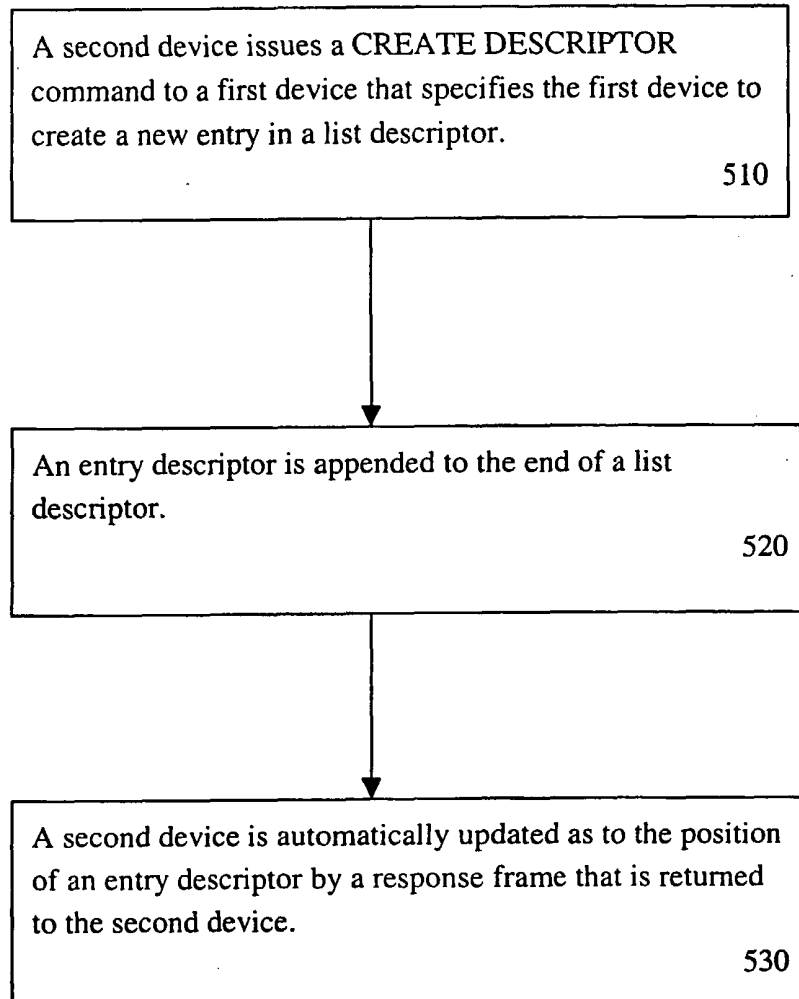
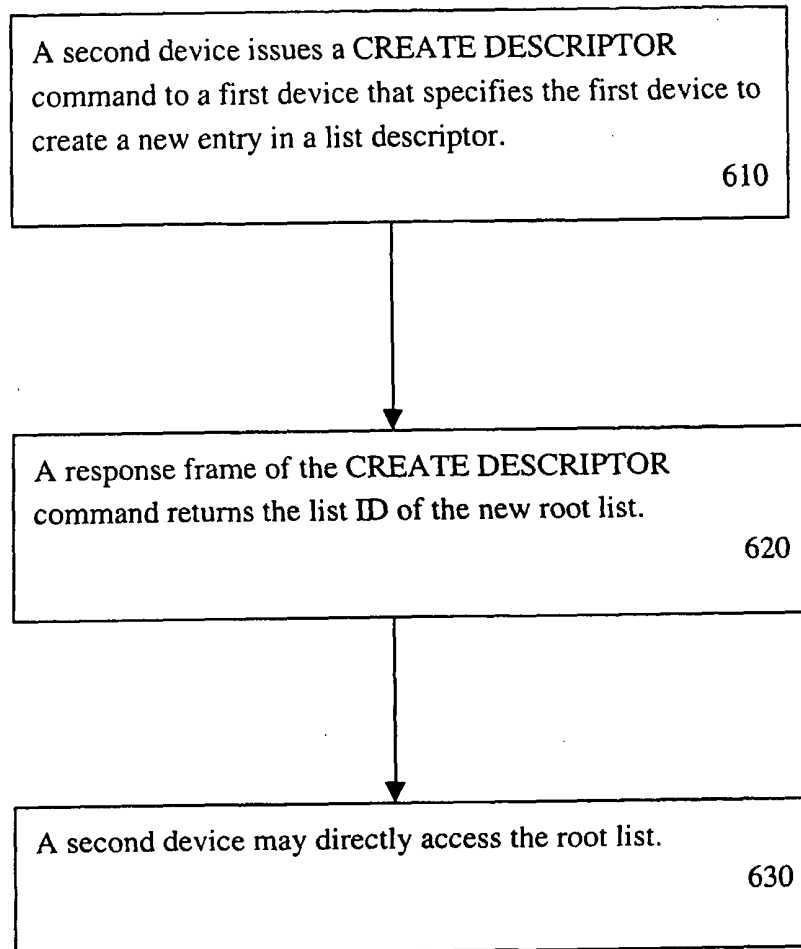


Fig. 2

3/5

**FIG. 3**

4/5

**Fig. 4**

5/5

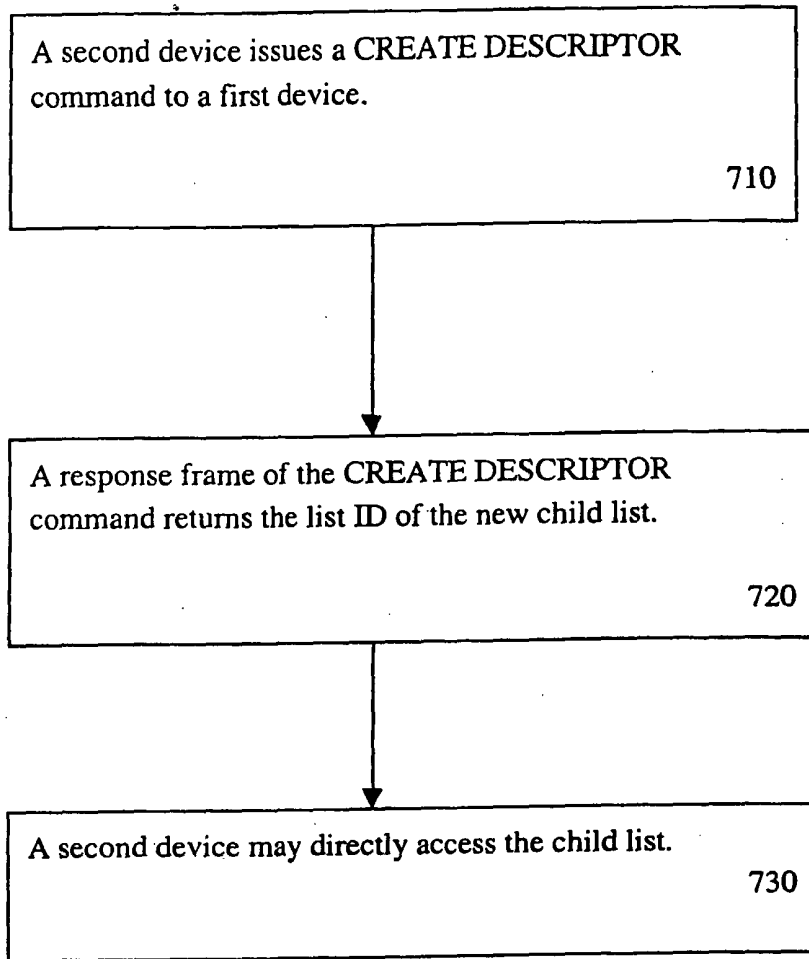


Fig. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/10792

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 13/00

US CL : 710/107, 5, 33; 345/352, 509; 707/4

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 710/107, 5, 33; 345/352, 509; 707/4

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 6,131,129 (LUDTKE et al.) 10 October 2000, figs. 7 and 19, col. 11, lines 11-54, col.14, lines 56-63, col. 15, 19-34.	1-31
Y, P	US 6,163,820 (GARRETT et al.) 19 December 2000, col. 3, lines 18-35, col. 4, lines 54-64.	1-31
Y, P	US 6,073,129 (LEVINE et al.) 6 June 2000, col. 9, lines 1-60.	7-12, 28-31
Y, P	US 6,182,166 B1 (SHKLARSKY et al.) 30 January 2001, col. 23, lines 30-43).	13-14, 17-20, 22-23
Y	US 5,966,127 (YAJIMA) 12 October 1999, col. 11, lines 47-56.	15, 21, 24

 Further documents are listed in the continuation of Box C.
 See patent family annex.

* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
B earlier document published on or after the international filing date	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	* & * document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means	
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

30 MAY 2001

Date of mailing of the international search report

19 JUL 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer


ETIENNE ARIO

Telephone No. (703)308-7562

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/10792

B. FIELDS SEARCHED

Electronic data bases consulted (Name of data base and where practicable terms used):

EAST, DIRECT PROQUEST

search terms: descriptor, list, table, file, queue, response, frame, packet, id , identifier, bus, controller, I/O device, node, controller, entry, value, append, data, message, access, create, item, etc.