

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
22 February 2001 (22.02.2001)

PCT

(10) International Publication Number
WO 01/13644 A1

(51) International Patent Classification⁷: H04N 7/24, 5/00

(21) International Application Number: PCT/US00/22404

(22) International Filing Date: 15 August 2000 (15.08.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/148,691 16 August 1999 (16.08.1999) US
09/618,074 17 July 2000 (17.07.2000) US

(71) Applicant: GENERAL INSTRUMENT CORPORATION [US/US]; 101 Tournament Drive, Horsham, PA 19044 (US).

(72) Inventors: POLI, Christopher; 2607 Red Gate Drive, Doylestown, PA 18901 (US). DEL SORDO, Christopher,

S.; 229 Heatherfield Drive, Soudertown, PA 18964 (US). BATES, Thomas, F., IV; 115 Tanyard Road, Richboro, PA 18954 (US). MAKOFKA, Douglas, S.; 516 Fairhill Street, Willow Grove, PA 19090 (US). LEHRMAN, Ira, S.; 10976 Puma Run, Littleton, CO 80124 (US).

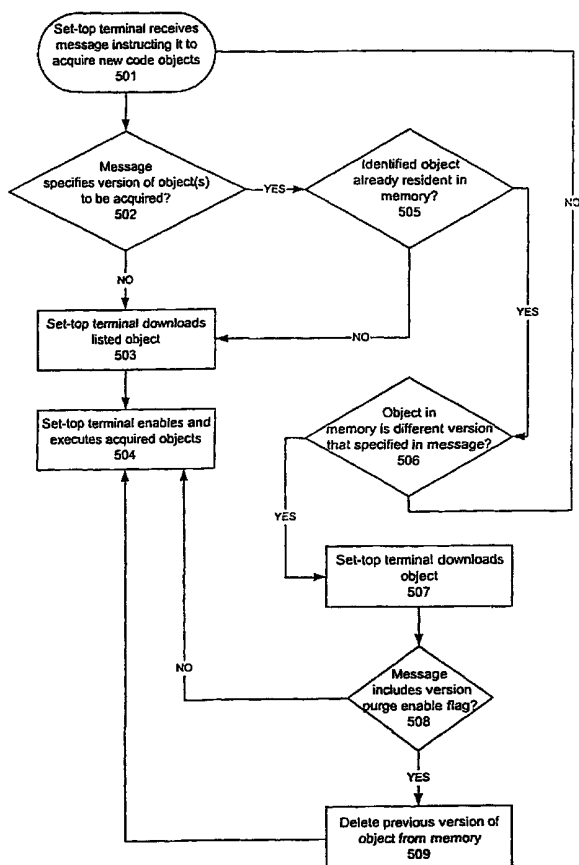
(74) Agent: SHIH, Anna, M.; Rader, Fishman & Grauer PLLC, Suite 140, 39533 Woodward Avenue, Bloomfield Hills, MI 48304 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, 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.

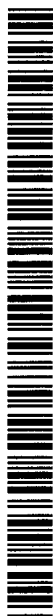
(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR AUTOMATICALLY PURGING CODE OBJECTS BEING UPDATED BY DOWNLOAD



(57) Abstract: A method and system allow a system operator of a cable television system to control the management of multiple versions of a code object in set-top terminals connected to the television system. When the system operator signals the set-top terminals to download code objects from the signal provided over the cable system, the download instructions specify a version of the object to be acquired. If an object with the same identifier, but a different version thereof, is already in the set-top terminal's memory unit, the terminal can automatically purge that other version of the object from memory when downloading the object specified by the system operator. A dedicated flag in the download instruction message can be used to selectively activate and deactivate the automatic purge from memory of different versions of an object being downloaded.



WO 01/13644 A1



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), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

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.

Published:

— *With international search report.*

TITLE OF THE INVENTION

METHOD AND SYSTEM FOR AUTOMATICALLY PURGING CODE OBJECTS BEING UPDATED BY
DOWNLOAD

FIELD OF THE INVENTION

5 The present invention relates to the field of programming or re-programming a set-top terminal in a cable television system. More particularly, the present invention relates to the field of controlling the download of programming, such as an operating system or native suite objects, for a set-top terminal over the cable network for initial or upgrade
10 programming of the set-top terminal.

BACKGROUND OF THE INVENTION

In a typical cable television system, subscribers are provided with a set-top box or terminal. The set-top terminal is a box of electronic equipment that is used to connect the
15 subscriber's television, and potentially other electronic equipment, with the cable network. The set-top box is usually connected to the cable network through a co-axial wall outlet.

The set-top box is essentially a computer that is programmed to process the signals from the cable network so as to provide the subscriber with the cable services. These services from the cable television company typically include access to a number of
20 television channels and, perhaps, an electronic programming guide (EPG). Additional premium channels may also be provided to subscribers for an additional fee. Pay-per-view events and video-on-demand service may also be provided over the cable network. The set-top box is programmed to provide these services to the subscriber.

However, the services of the cable company need not be limited to providing
25 television programming. Some cable companies are now offering internet access and e-mail service over their cable networks at speeds much faster than are available over conventional telephone lines. It is anticipated in the future that more and more services will be provided over the cable network, including perhaps even basic telephone service. Eventually, each home or office may have a single connection, via the cable network, to all
30 electronic data services.

As the cable network and the services provided evolve, the set-top terminal must also evolve so as to be able to provide subscribers with all the services available from the

cable network that are of interest to those subscribers. This set-top box evolution will primarily involve changes to the programming of the set-top box. By upgrading the software or firmware of the set-top box, the box can be made to perform more efficiently or offer new services as the cable network evolves.

5 In order to upgrade the population of set-top boxes on a cable network, it is preferable to transmit the new programming to the set-top boxes via the cable network itself. Otherwise, a technician must visit each subscriber to upgrade or re-program that subscriber's set-top box. Such field upgrades would obviously be at significant expense.

10 The headend is the facility from which the cable network operator broadcasts television signals and provides other services over the cable network. Updated software that is provided to the population of set-top terminals is broadcast from the headend over the cable network. The headend may also broadcast messages or instructions over the cable network to the set-top terminals.

15 As used herein, a programming code object, or code object, refers to an individual program, i.e. a piece of software or firmware, that can be downloaded to and executed by a set-top terminal. More broadly, a data object refers to any data structure, including a code object, message or instruction, that is transmitted by the headend over the cable network to one or more set-top terminals.

20 Frequently, in order to provide new services or to upgrade the programming in a set-top terminal, the set-top terminal must acquire several new code objects, i.e., pieces of software or firmware. These various pieces of programming may frequently interact and require each other in order to function properly. Consequently, problems arise if the set-top terminal acquires some, but not all, of the code objects that are to be downloaded as part of the programming upgrade. When the set-top terminal begins running the newly-

25 acquired code objects, those code objects may likely function improperly or not at all due to the absence of other interdependent code objects that the set-top terminal was to have acquired and executed.

30 Alternatively, implementing a new service or enhancing the operation of the set-top terminal may involve replacing a older version of a piece of programming, i.e. a code object, on the terminal with a newer, improved version. Conversely, it may be the case that a newly implemented version of a code object is found to operate improperly or cause

problems with the system thereby causing the system operator to want to revert to the earlier version of the object until the problems with the new version can be resolved.

In any of these instances, a set-top terminal may have two versions of the same code object to manage after acquiring the newer (or perhaps older) version in response to an instruction to do so from the system operator. Consequently, there is a need in the art for a method and system of controlling the download of code objects by a set-top terminal in a cable television system so as to prevent potential conflicts and the unnecessary consumption of memory associated with managing two different versions of the same code object.

SUMMARY OF THE INVENTION

It is an object of the present invention to meet the above-described needs and others. Specifically, it is an object of the present invention to provide a method and system of controlling the download of code objects by a set-top terminal in a cable television system so as to prevent potential conflicts and the unnecessary consumption of memory associated with managing two different versions of the same code object.

Additional objects, advantages and novel features of the invention will be set forth in the description which follows or may be learned by those skilled in the art through reading these materials or practicing the invention. The objects and advantages of the invention may be achieved through the means recited in the attached claims.

To achieve these stated and other objects, the present invention may be embodied and described as a method of managing multiple versions of a code object being downloaded by a set-top terminal in a cable television system where the set-top terminal receives a download instruction message over the cable television system that specifies at least one code object to be downloaded by the terminal and further specifies a version of the code object to be downloaded. The method of the present invention performs this function by selectively deleting from the memory unit of the set-top terminal a resident code object corresponding to the code object specified for download in the download instruction message but which resident code object in the memory unit is a version different from that specified in the download instruction message for the code object to be downloaded.

Described in more detail, the method of the present invention include identifying any resident code object stored in the memory unit of the set-top terminal that corresponds to the code object specified for download in the download instruction message; and comparing a version of any such resident code object with a version of the code object to be downloaded specified in the download instruction message. If the version indicators don't match, the resident code object may be deleted.

Alternatively, the method may include ignoring a download instruction of the download instruction message if the resident code object identified as corresponding to the code object specified for download in the download instruction message and is of the same version specified for the code object to be downloaded in the download instruction message.

Preferably, the process of selectively deleting a resident code object is performed in accordance with a version purge enable flag contained in the download instruction message. More specifically, the method preferably includes reading the version purge enable flag contained in the download instruction message; and performing the step of deleting a resident code object corresponding to the code object specified for download in the download instruction message, but which resident code object in the memory unit is a version different from that specified in the download instruction message for the code object to be downloaded, only if the version purge enable flag is affirmative.

The method of the present invention also encompasses the system or hardware necessary to perform the above-described method. More specifically, the present invention encompasses a system for managing multiple versions of a code object being downloaded by a set-top terminal in a cable television system. In this system, the set-top terminal receives a download instruction message over the cable television system that specifies at least one code object to be downloaded by the terminal and further specifies a version of the code object to be downloaded. The set-top terminal comprises a processor for downloading and executing code objects and a memory unit for storing code objects. The set-top terminal selectively deletes from the memory unit a resident code object corresponding to the code object specified for download in the download instruction message but which resident code object in the memory unit is a version different from that specified in the download instruction message for the code object to be downloaded.

BRIEF DESCRIPTION OF THE DRAWINGS:

The accompanying drawings illustrate the present invention and are a part of the specification. Together with the following description, the drawings demonstrate and explain the principles of the present invention.

Fig. 1 is a flowchart showing the basic sequence in which programming objects are acquired by the set-top terminal according to a related invention.

Fig. 2 is a block diagram of the basic elements of a cable television system with which the present invention is practiced.

Fig. 3 is a flowchart showing a method of controlling and enabling a list of downloaded code objects.

Fig. 4 is a graphic illustration of a download control message.

Fig. 5 is a flowchart showing a method of managing multiple versions of a code object being downloaded by a set-top terminal in a cable television system according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION:

A related patent application, U.S. Serial No. 60/130,328 (incorporated herein by reference in its entirety) comprehensively describes a proprietary method and system for downloading programming objects to set-top terminals in a cable system. Fig. 1 is a flowchart that illustrates the basic sequence in which programming objects are acquired by the set-top terminal over the cable system according to that method. Fig. 2 is a block diagram showing the basic elements of a cable television system which operates using the present invention.

With references to Figs. 1 and 2, a set-top terminal (202; Fig. 2) executes an automatic boot process immediately following connection of power to the set-top terminal. As shown in Fig. 1, the boot process begins with the set-top terminal executing a piece of programming known as the boot code (100). The boot code is a programming object that is factory-loaded into the read-only memory (ROM) of the set-top terminal. As with all programming executed by the set-top terminal (202), the boot code is stored in the memory unit (212; Fig. 2) and executed by the processor (211; Fig. 2)

In another alternative, the set-top terminal (202) may not initiate the boot process shown in Fig.1 until both power and a data transport stream from the headend (201; Fig. 2) have been connected to the terminal (202). The set-top terminal may also execute the boot code (100) in response to a reset signal received, for example, from the headend (201), i.e., the system operator.

When running, the boot code provides minimal programming for the set-top terminal (202) to allow the terminal to bring itself up to full functionality. The boot code does not typically allow the set-top terminal (202) to provide any services, but is designed only to acquire, if necessary, and load higher-level programming. The term "boot code" as used herein comprises the minimal code needed to accomplish this functionality.

As shown in Fig. 1, the boot code will first acquire the base platform code (101-105). The boot code will first check the memory (212; Fig. 2) of the set-top terminal (202) to determine if the base platform code is stored there (101, 102). If the base platform code is in memory, the processor (211; Fig. 2) of the set-top terminal (202) running the boot code will authenticate the base platform code (104) and, assuming the code passes the authentication process, will begin executing the base platform code (105).

If the base platform code is not in memory (102), the set-top terminal (202) will access the data transport stream from the headend (201; Fig. 2), locate a transmission of the base platform code within the headend signal using the tuner (210; Fig. 2) and download the base platform code (103). The base platform code is preferably transmitted to the set-top terminal on an out-of-band transport stream for storage in a Flash memory portion of the memory unit (212). However, the base platform code may be transmitted on an in-band control channel.

The downloaded platform code is then authenticated (104), as before, and executed. The boot code will preferably re-authenticate the base platform code every time it launches the base platform object regardless of whether the base platform code has been found in memory (212) or downloaded.

Once the base platform code is launched, control of the terminal (202) passes to the base platform code from the boot code. When executing the base platform code, the set-top terminal (202) can provide minimal functionality to subscribers, for example, allowing the subscriber to receive television programming from the cable system. However, in order for the set-top terminal to attain full functionality, the base platform code must acquire an

additional programming known as the native suite. The native suite comprises an operating system object and, usually, additional application programs. These additional or "resident" applications are computer programs that run on the set-top terminal under the operating system. The resident applications provide the capabilities of the set-top terminal that are in addition to watching television.

As shown in Fig.1, the base platform code, when first executed or reset by a signal from the headend, checks (106, 107) the memory unit (212) to determine if an operating system code object is already resident in the terminal (202). The operating system (O/S) is typically code from a third party (such as Microsoft's WinCE™) that allows the set-top terminal (202) to run the various resident applications of the native suite. The operating system code typically uses an additional embedded code module provided by the manufacturer of the set-top terminal which interfaces the operating system with the particular hardware of that set-top terminal to enable the operating system to function with that specific set-top terminal.

If the operating system code object is found in memory, the base platform code authorizes and authenticates the operating system code (109) from memory. If the operating system code is not in memory, the set-top terminal (202) running the base platform code, will access the data transport stream from the headend (201), and locate and download an appropriate operating system code object (108). The downloaded object is then authorized and authenticated (109).

Once authorized and authenticated, the operating system code is executed (110). When running the operating system code, the set-top terminal can also execute the additional code objects or applications of the native suite that provide any variety of services to subscribers. The base platform code can be configured to acquire all the elements of the native suite along with the operating system. Alternatively, as illustrated in Fig. 1, the operating system, when running, may be configured to acquire other applications.

Preferably, the processor (211) running the operating system code first checks the memory unit (212) for the expected elements of the native suite (111). If the native suite applications are in memory (112), the operating system can authorize and authenticate those applications (114) and execute them (115). The processor (211) will then determine if it has been instructed to acquire any additional objects (112), including native suite

applications not found in memory. If not, the set-top terminal is at full functionality (116). If, however, the processor (211) needs to acquire additional objects, those applications are downloaded from the cable network (113). The downloaded application objects are authorized and authenticated (117), and executed (118).

5 Once the operating system and other native suite applications are running, the set-top terminal has achieved full functionality (116). As will be clear to those skilled in the art, the list of native suite applications may vary from set-top terminal to set-top terminal. A subscriber who subscribes to fewer services from the cable system may need fewer native suite applications than a subscriber who pays for all the services available.
10 Additionally, as new services are added, applications may be added to the native suite to support the new services. Thus, the list of applications, which constitute the native suite, can vary with both time and subscriber needs and preferences.

 As noted above, the base platform code may acquire the native suite consisting of the operating system and resident applications during initialization of the set-top terminal
15 or upon receipt of an instruction to do so from the system operator. Consequently, when the system operator desires to alter or upgrade the operating system or resident applications in the set-top terminal, the system operator may send an initialization message to the set-top terminal via the cable network which instructs the base platform code to acquire or reacquire the operating system and resident applications then being broadcast over the
20 cable network. Alternatively, the system operator can signal the operating system to acquire or re-acquire elements of the native suite. These messages from the headend (201) may also be, at the same time, altering the list of elements that define the terminal's native suite.

 While this arrangement allows the capabilities of the set-top terminal to be readily
25 upgraded, altered or enhanced, interruptions to service for the set-top user may also result. For example, when the base platform code or operating system code receives an instruction to acquire new code objects over the cable network, that instruction will specify the code object or objects which are to be acquired. If more than one object is to be acquired, a problem may result if only one of the objects is actually received by the set-top terminal
30 (202).

 Specifically, the terminal (202) may receive and enable the first code object which it has been instructed to acquire. If then, for whatever reason, the set-top terminal is unable

to acquire the other code objects that it has been instructed to download, the functions performed by those unreceived code objects will be unavailable to the set-top terminal user.

Moreover, code objects are frequently inter-related. Consequently, the object that has been downloaded may not function properly without the other objects that were to have been downloaded, but were not received. Similarly, enablement and execution of the new code object which has been received may interfere with the execution of the software or firmware previously resident on the set-top terminal if the other objects specified for download are not acquired and executed as well.

Therefore, as noted above, there is a need in the art for a method of controlling the download of code objects by a set-top terminal in a cable television system so as to prevent the execution of newly-acquired code objects that require the presence of other code objects which the set-top terminal was supposed to have acquired but was unable, for whatever reason, to download. The present invention overcomes this possible problem, in the following manner.

The set-top terminal (202) running the base platform or operating system code will identify the list of objects it has been instructed to acquire. Typically, this list will be a complete list of the objects in the native suite. However, it is possible that the list of objects to be acquired may represent only a portion of the native suite.

The list of objects to be acquired and the instruction to acquire them may come from either a message received from the system operator via the headend (201) or may be part of the terminal's programming, either base platform or operating system code. In either case, the set-top terminal (202) will begin acquiring the listed objects from the transport stream received over the cable system (203) from the headend (201). The terminal (202) will not, however, enable or execute any of the received objects until all the objects listed to be acquired have been successfully received. The programming managing the download, e.g., the base platform code or the operating system code, is modified or reconfigured to require successful acquisition of all listed code objects before any of the objects are enabled and executed. In this way, the set-top terminal (202) will not end up with only a fraction of the objects it has been instructed to acquire and execute, or with an object that conflicts with programming already resident.

A graphic illustration of a message instructing a set-top terminal to download new or additional code objects is presented in Fig. 4. As shown in Fig. 4, the message (401)

may also contain a list-only enable flag (402) that will instruct the set-top terminal (202) whether it must enable only the entire list of objects being acquired or can accept, enable and execute object representing only a fraction of those listed as to be acquired. This list-only enable flag (402) need only be a single bit within the message (401) and is, of course, either affirmative (i.e., "present" or "on") or negative (i.e., "not present" or "off"). In other words, if the list-only enable flag is present, all of the objects received are enabled simultaneously or nearly simultaneously. Therefore, no objects will be enabled until all the objects in the list have been properly received. The simultaneous enablement of all the received objects will create the least possible disruption in service to the user of the set-top terminal.

In addition to enabling and executing the received code objects only when all listed objects have been received, the set-top terminal (202) may be programmed to purge listed objects from memory that have been received if all program objects in the object group are not received. In other words, if the set-top terminal (202) begins acquiring a list of objects that it has been instructed to acquire, and not all of the listed objects are eventually received, it may be desirable to remove from memory any of the objects that were acquired but which will not be used or enabled due to the failure to acquire all objects in the object group. Consequently, as an option of the present invention, the set-top terminal (202) may be programmed to delete from memory any code objects which have been acquired as a list of objects to be received in the event that not all listed objects are eventually received, enabled and used.

The set-top uses the list identifier and the list version to identify those objects that comprise the defined object suite (most recent list of objects from the download message). If an object is added or removed from a list in a download message, the set-top will update its suite of objects in memory associated with that list by adding or removing those object(s). In addition, the list version can be used to force the upgrade of an object where the object version identified in the list of the download message is different than the object version that the set-top has in memory.

This function of purging different versions of objects in a list may also be governed by a flag (403), i.e. an auto purge flag, within the message (401) from the headend. Again, the flag (403) need only be a single bit that represents that the flag is either present or not present within the message. In this way, the system operator has the option of clearing the

set-top terminal's memory (212) of all the listed objects of a different version than that specified in the download message.

This process is outlined in Fig. 3. As shown in Fig. 3, the method of the present invention begins with the set-top terminal receiving a message instructing it to acquire new
5 or additional code objects from the cable system (301). The set-top terminal first checks to see if the auto purge flag is present or positive in the message (306). If so, the terminal purges objects appropriately that are different versions of an object specified for download in the incoming message (309). After the purging, or if the auto purge flag is negative (306), the set-top attempts to download the code objects listed in the message (302).

10 When the downloading is complete or has otherwise been terminated, perhaps by a time-out counter, the terminal will determine if all the objects it was instructed to acquire were successfully downloaded. (303). If the terminal failed to acquire all the specified objects, the terminal may delete all the objects that were acquired from memory (307).

If, on the other hand, all the objects have been acquired, the set-top terminal can
15 proceed to enable and execute the objects. The terminal first check for the list-only enable flag (305). If the list enable flag is present or positive, the enablement of the downloaded objects is preferably performed simultaneously or nearly simultaneously, thereby treating the objects on the list as though they were a single object (304).

If the list enable flag is negative, but all objects have been received successfully, the
20 set-top will leave the acquired objects in memory, but without enabling or executing them (308). If only some of the objects that were to be acquired as successfully received (303), the set-top terminal will delete all objects on the list (307).

Under the principles of the present invention, then, a native or operating suite which consists of a number of code objects can be transmitted and used by the set-top terminals in
25 a cable television system for all intents and purposes as though the suite were a single code object. This feature of the present invention allows system operators to minimize disruptions to service and confusion which might otherwise result when the set of code objects are only partially downloaded successfully.

Another aspect of the present invention involves the management within the set-top
30 terminal of multiple versions of an object that result from a download to the terminal in response to an instruction from the headend. Objects may be identified uniquely within a system environment by an object identifier and object version (i.e., object_name and

object_version). This object_name is unique within the download transport stream(s) available to an individual set-top terminal (202). The download instruction message, which instructs the set-top terminal (202) to acquire a list of objects, may also identify both the object name/unique identifier and the version of the object desired. If the set-top
5 already has that particular object, but a different version, it may be desirable to remove the object currently resident in the set-top terminal, and acquire the version identified in the instruction message. Preferably, the instruction message from the system operator which initiates the acquisition of new code objects also includes a version purge enable flag (404) that instructs the set-top terminal whether to automatically purge code objects already
10 resident in the set-top terminal's memory which are identified in the message's object list by a unique object identifier (i.e., object_name), but which are a version other than that listed in the message's object list (i.e., object_version).

This aspect of the present invention is detailed in Fig. 5. As shown in Fig. 5, the set-top terminal (202) receives the download instruction message directing the terminal to
15 download one or more code objects from the cable system (501). The set-top terminal will first determine if, in addition to identifying objects for download, the download instruction message also specifies a version of the object to be downloaded. (502). If the download instruction message does not specify object version, the set-top terminal proceeds to download, enable and execute the code objects as instructed (503, 504).

20 If the download instruction message does specify the version of a code object or objects to be acquired, the set-top terminal (202) will check its memory (212) to determine if the object to be download is already resident in memory (212). If the object is not in memory (212), the set-top terminal (202) proceeds to execute the download instructions and acquires the object (503). Conversely, if the identified object is in memory (212), the
25 set-top terminal (202) checks the version of the object in memory against the version specified by the download instruction message for the object to be acquired (506).

If the version of the object in memory matches the version of the object the terminal has been directed to acquire, the set-top terminal (202) can ignore the direction to
download that code object because the desired version of the object is already in memory
30 (212). This assists the system operator when, for example, new set-top terminals are added to the cable system with newer versions of code objects already installed. The system operator can instruct the older terminals to acquire the newer version of the code object,

with the instruction being appropriately ignored by newer terminals already having the object.

If the object in memory is a different version, either newer or older, than the version specified for download in the download instruction message, the set-top terminal acquires the specified object (507). The terminal then checks the download instruction message for a version purge enable flag (508). This flag (404) operates similarly to the flags discussed earlier and is illustrated in Fig. 4. As with the other flags discussed herein, the version purge enable flag (404) need only be one binary bit to indicate whether the set-top terminal should automatically delete from memory code object which bears the same identifier, but is a different version, that the code object that has been downloaded as directed by the download instruction message.

If the version purge enable flag (404) is present or affirmative, the set-top terminal (202) automatically deletes from memory any version of the acquired object that does not match the version of that object specified in the download instruction message (509). The newly-acquired version of the object is then enabled and executed (504).

In this way, potential conflicts and wasted memory that may result from the set-top terminal having two different versions of the same code object are eliminated.

The preceding description has been presented only to illustrate and describe the invention. It is not intended to be exhaustive or to limit the invention to any precise form disclosed. Many modifications and variations are possible in light of the above teaching.

The preferred embodiment was chosen and described in order to best explain the principles of the invention and its practical application. The preceding description is intended to enable others skilled in the art to best utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A method of managing multiple versions of a code object being downloaded by a set-top terminal in a cable television system where the set-top terminal receives a download instruction message over the cable television system that specifies at least one code object to be downloaded by the terminal and further specifies a version of said code object to be downloaded, the method comprising selectively deleting from a memory unit of said set-top terminal a resident code object corresponding to said code object specified for download in said download instruction message but which resident code object in said memory unit is a version different from that specified in said download instruction message for said code object to be downloaded.

2. The method of claim 1, further comprising downloading said code object specified in said download instruction message to said set-top terminal.

3. The method of claim 1, wherein said selectively deleting a resident code object is performed in accordance with a version purge enable flag contained in said download instruction message.

4. The method of claim 3, further comprising:
reading said version purge enable flag contained in said download instruction message; and
performing said step of deleting a resident code object corresponding to said code object specified for download in said download instruction message, but which resident code object in said memory unit is a version different from that specified in said download instruction message for said code object to be downloaded, only if said version purge enable flag is affirmative.

5. The method of claim 1, wherein said method further comprises:
identifying any resident code object stored in said memory unit of said set-top terminal that corresponds to said code object specified for download in said download instruction message; and

comparing a version of any such resident code object with a version of said code object to be downloaded specified in said download instruction method.

5 6. The method of claim 5, further comprising ignoring a download instruction of said download instruction message if a resident code object is identified that corresponds to said code object specified for download in said download instruction message and which resident code object is a version matching that specified for said code object to be downloaded in said download instruction message.

10 7. A system for managing multiple versions of a code object being downloaded by a set-top terminal in a cable television system comprising:

 means with said set-top terminal for receiving a download instruction message over the cable television system, wherein said message specifies at least one code object to be downloaded by the terminal and further specifies a version of said code object to be
15 downloaded; and

 means for selectively deleting from a memory unit of said set-top terminal a resident code object corresponding to said code object specified for download in said download instruction message but which resident code object in said memory unit is a version different from that specified in said download instruction message for said code
20 object to be downloaded.

 8. The system of claim 7, further comprising means for downloading said code object specified in said download instruction message to said set-top terminal.

25 9. The system of claim 7, wherein means for said selectively deleting a resident code object operate in accordance with an version purge enable flag contained in said download instruction message.

 10. The system of claim 9, further comprising:

30 means for reading said version purge enable flag contained in said download instruction message;

wherein said means for deleting a resident code object corresponding to said code object specified for download in said download instruction message, but which resident code object in said memory unit is a version different from that specified in said download instruction message for said code object to be downloaded, are activated only if said version purge enable flag is affirmative.

11. The system of claim 7, further comprising:

means for identifying any resident code object stored in said memory unit of said set-top terminal that corresponds to said code object specified for download in said download instruction message; and

means for comparing a version of any such resident code object with a version of said code object to be downloaded specified in said download instruction method.

12. The system of claim 11, further comprising means for ignoring a download instruction of said download instruction message if a resident code object is identified that corresponds to said code object specified for download in said download instruction message and which resident code object is a version matching that specified for said code object to be downloaded in said download instruction message.

13. A system for managing multiple versions of a code object being downloaded by a set-top terminal in a cable television system comprising:

at least one set-top terminal connected to a cable television system, wherein said set-top terminal receives a download instruction message over said cable television system that specifies at least one code object to be downloaded by the terminal and further specifies a version of said code object to be downloaded;

wherein said set-top terminal comprises a processor for downloading and executing code objects and a memory unit for storing code objects; and

wherein said set-top terminal selectively deletes from said memory unit a resident code object corresponding to said code object specified for download in said download instruction message but which resident code object in said memory unit is a version different from that specified in said download instruction message for said code object to be downloaded.

14. The system of claim 13, wherein said set-top terminal further comprises a tuner controlled by said processor for accessing a data stream transmitted over said cable television system from which said set-top terminal can download code objects.

5

15. The system of claim 13, wherein said set-top terminal selectively deletes said resident code object in accordance with an version purge enable flag contained in said download instruction message.

10

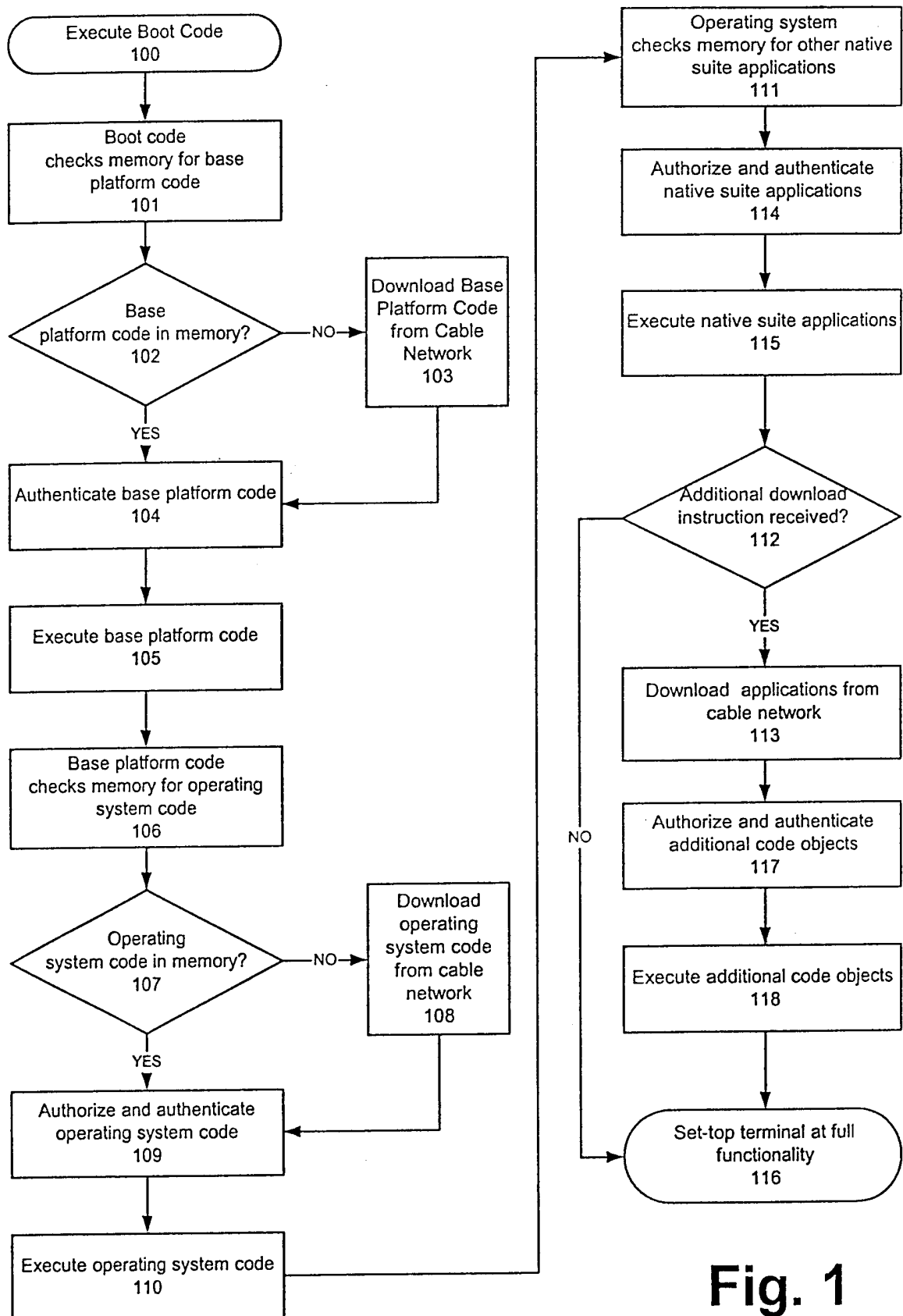
16. The system of claim 15, wherein said processor of said set-top terminal reads said version purge enable flag contained in said download instruction message and deletes said resident code object only if said version purge enable flag is affirmative.

15

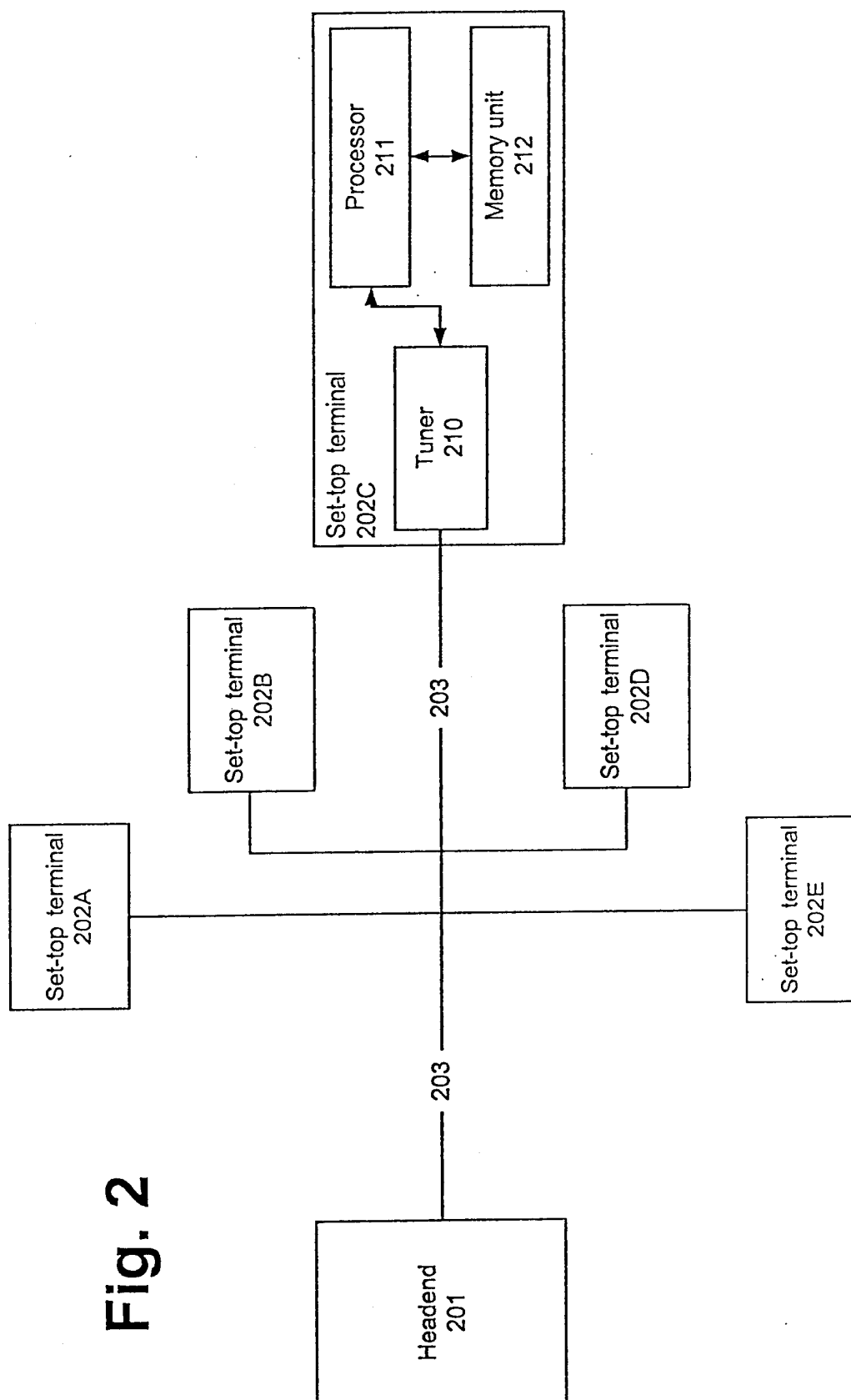
17. The system of claim 13, wherein said processor identifies any resident code object stored in said memory unit of said set-top terminal that corresponds to said code object specified for download in said download instruction message; and compares a version of any such resident code object with a version of said code object to be downloaded specified in said download instruction method.

20

18. The system of claim 17, wherein said processor ignores a download instruction of said download instruction message if a resident code object is identified that corresponds to said code object specified for download in said download instruction message and which resident code object is a version matching that specified for said code object to be downloaded in said download instruction message.

**Fig. 1**

2/5



3/5

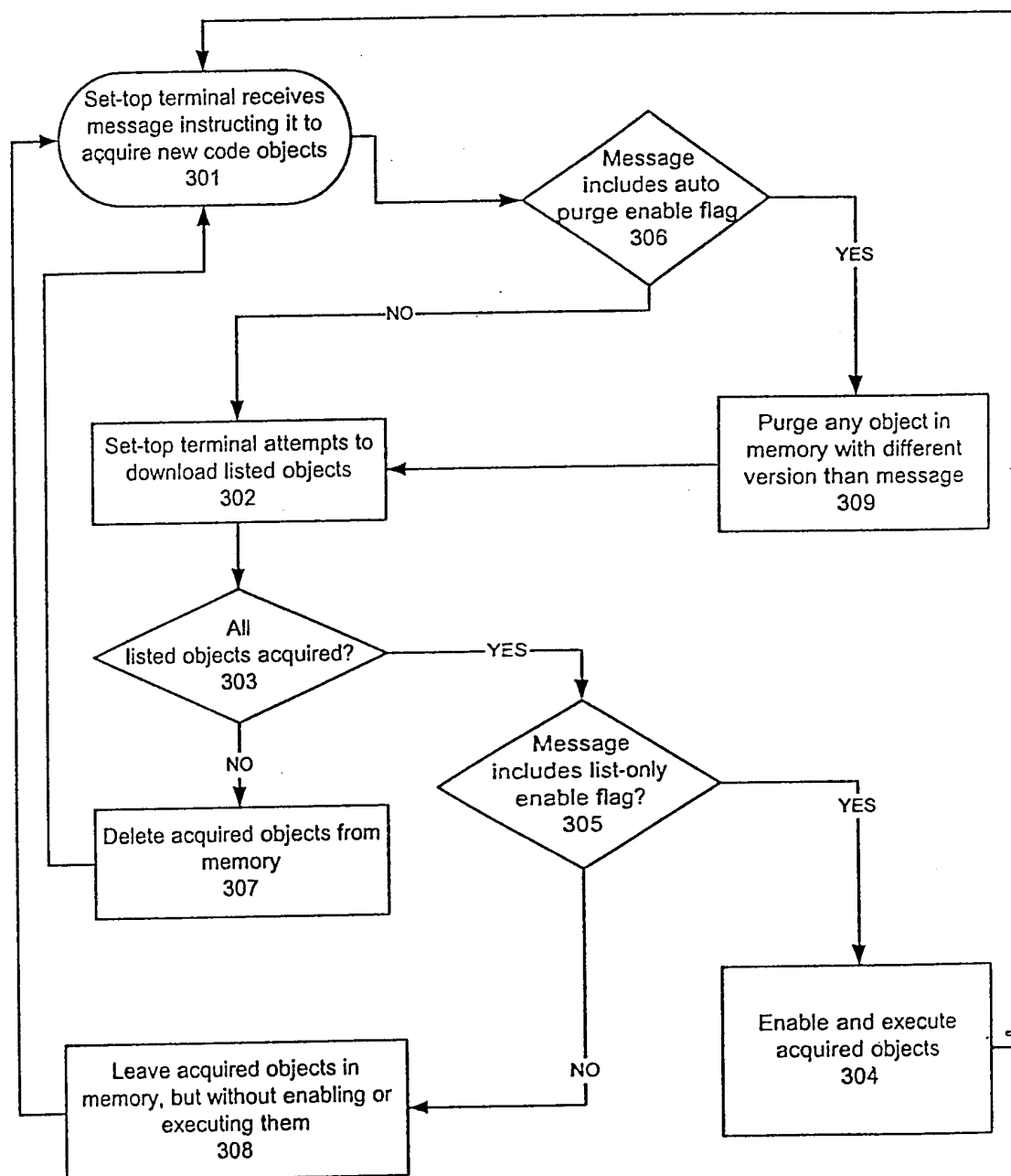


Fig. 3

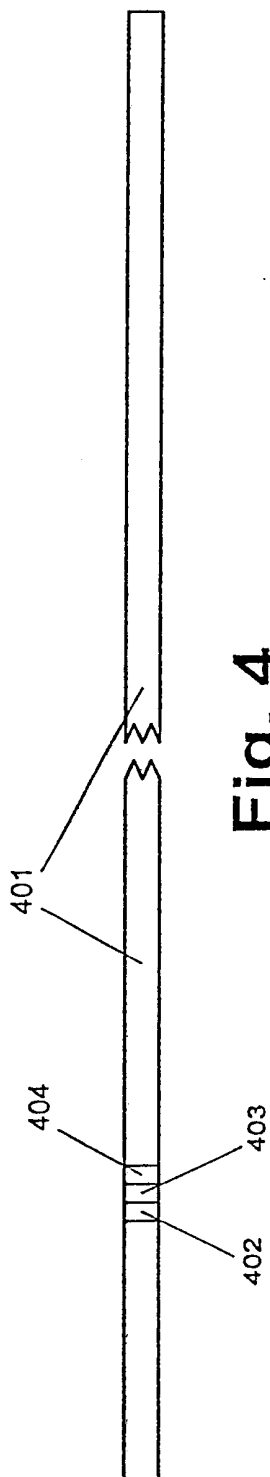
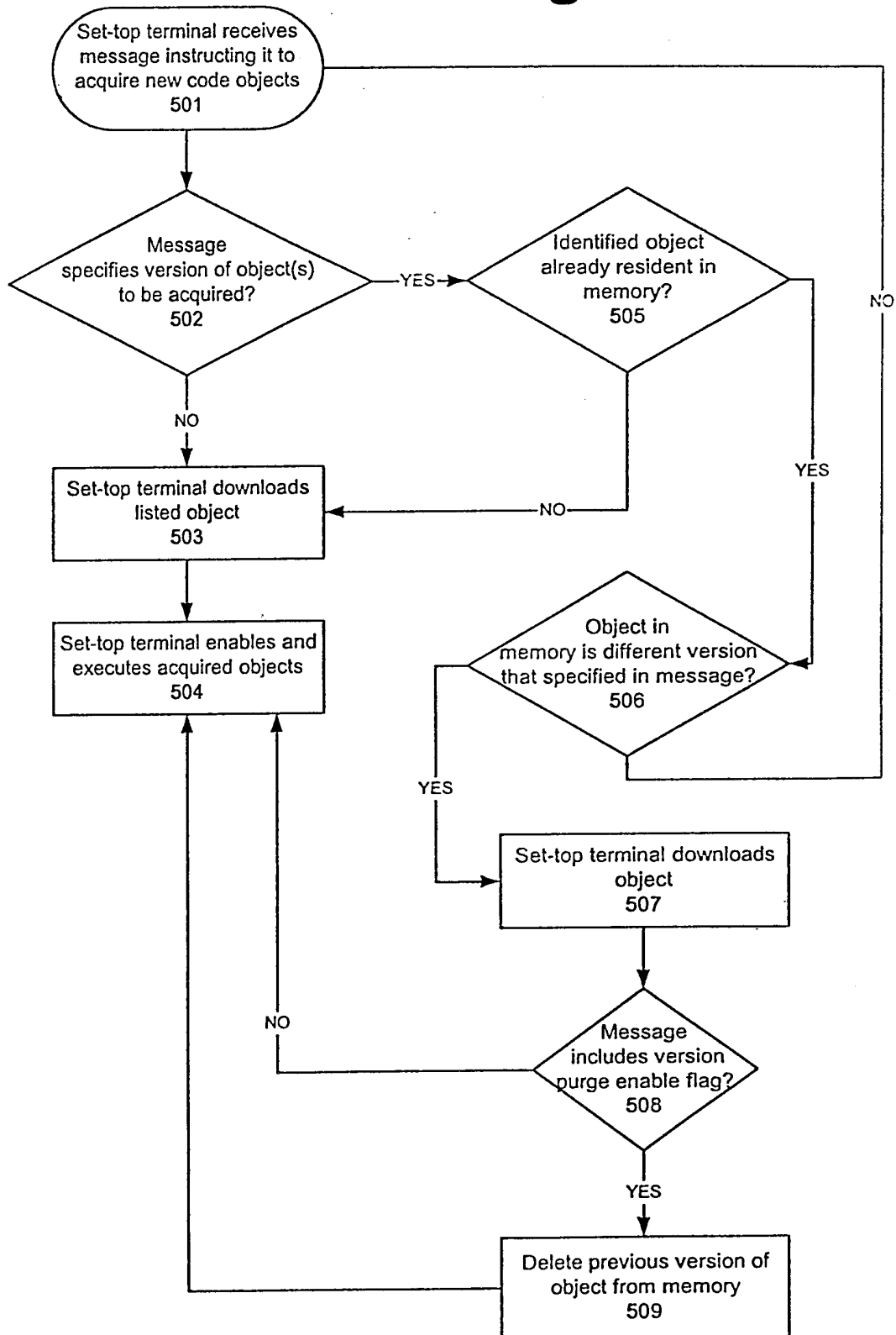


Fig. 4

5/5

Fig. 5



INTERNATIONAL SEARCH REPORT

Internat I Application No

PCT/US 00/22404

A. CLASSIFICATION OF SUBJECT MATTER
 IPC 7 H04N7/24 H04N5/00

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)

IPC 7 H04N G06F

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 practical, search terms used)

EPO-Internal, WPI Data, PAJ

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 907 285 A (CANAL PLUS SA) 7 April 1999 (1999-04-07) column 1, line 1 -column 6, line 49 column 14, line 52 -column 21, line 48 abstract; figures 12,13	1,2,5-8, 11-14, 17,18
A	---	3,4,9, 10,15,16
X	US 5 798 785 A (BONNER ALFRED E ET AL) 25 August 1998 (1998-08-25) column 2, line 38 - line 61 column 23, line 25 - line 35 column 25, line 26 -column 28, line 51 abstract; figures 10A,10B --- -/--	1-18

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

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)

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

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

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

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.

Z document member of the same patent family

Date of the actual completion of the international search

10 November 2000

Date of mailing of the international search report

30/11/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
 Fax: (+31-70) 340-3016

Authorized officer

La, V

INTERNATIONAL SEARCH REPORT

Internat' Application No

PCT/US 00/22404

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 666 293 A (HUDSON JR HENRY G ET AL) 9 September 1997 (1997-09-09) column 4, line 39 -column 5, line 67 column 8, line 9 -column 10, line 12 column 22, line 46 -column 23, line 39 column 35, line 54 -column 38, line 38 abstract; figure 9 ---	1-18
A	US 5 440 632 A (BACON KINNEY C ET AL) 8 August 1995 (1995-08-08) column 2, line 1 -column 3, line 64 column 9, line 25 -column 11, line 42 column 13, line 54 -column 16, line 42 abstract; figure 9 ---	1-18
A	WO 98 43415 A (DECLERCK CHRISTOPHE ; SARFATI JEAN CLAUDE (FR); CANAL PLUS SA (FR);) 1 October 1998 (1998-10-01) claims 12-15 ---	1-18
A	US 5 835 911 A (NAKAGAWA TORU ET AL) 10 November 1998 (1998-11-10) column 30, line 14 -column 31, line 34 abstract ---	1-18
A	US 5 859 977 A (KOBAYASHI HIROSHI ET AL) 12 January 1999 (1999-01-12) column 12, line 16 -column 14, line 16 -----	1-18

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/22404

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 0907285	A	07-04-1999	AU 9278398 A	27-04-1999
			BR 9813235 A	22-08-2000
			EP 1020078 A	19-07-2000
			WO 9918724 A	15-04-1999
			NO 20001651 A	05-06-2000
			ZA 9808996 A	24-05-1999
<hr/>				
US 5798785	A	25-08-1998	AU 1333795 A	19-06-1995
			AU 693148 B	25-06-1998
			AU 1430695 A	19-06-1995
			BR 9408211 A	26-08-1997
			CA 2177153 A	08-06-1995
			CA 2177154 A	08-06-1995
			EP 0732027 A	18-09-1996
			EP 0732031 A	18-09-1996
			EP 0963116 A	08-12-1999
			JP 9506225 T	17-06-1997
			JP 9510327 T	14-10-1997
			NZ 278185 A	27-04-1998
			WO 9515649 A	08-06-1995
			WO 9515658 A	08-06-1995
			AT 177277 T	15-03-1999
			AT 176840 T	15-03-1999
			AT 192005 T	15-05-2000
			AT 190180 T	15-03-2000
			AT 183352 T	15-08-1999
			AT 176841 T	15-03-1999
			AU 715683 B	10-02-2000
			AU 4440797 A	29-01-1998
			AU 712157 B	28-10-1999
			AU 4532597 A	05-02-1998
			AU 693775 B	09-07-1998
			AU 5732994 A	04-07-1994
			AU 692427 B	11-06-1998
			AU 5733094 A	04-07-1994
			AU 691479 B	21-05-1998
			AU 5733194 A	04-07-1994
			AU 692428 B	11-06-1998
			AU 5733294 A	04-07-1994
			AU 5736394 A	04-07-1994
			AU 5845894 A	22-06-1994
			AU 5869894 A	04-07-1994
			AU 716184 B	24-02-2000
			AU 6066798 A	04-06-1998
			AU 716182 B	24-02-2000
			AU 6066898 A	04-06-1998
			BR 9307619 A	15-06-1999
			BR 9307620 A	10-08-1999
			BR 9307621 A	15-06-1999
			BR 9307622 A	15-06-1999
			BR 9307623 A	16-05-2000
			BR 9307624 A	15-06-1999
			BR 9307625 A	31-08-1999
			CA 2151456 A	23-06-1994
			CA 2151457 A	23-06-1994
<hr/>				
US 5666293	A	09-09-1997	US 5734589 A	31-03-1998
			US 5635979 A	03-06-1997

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/22404

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5666293 A		US 5768539 A US 5978855 A AU 2657995 A WO 9533338 A	16-06-1998 02-11-1999 21-12-1995 07-12-1995
US 5440632 A	08-08-1995	US 5715515 A	03-02-1998
WO 9843415 A	01-10-1998	AU 2770897 A EP 0968602 A NO 994538 A PL 335776 A AU 2770697 A AU 7038198 A BR 9714603 A BR 9808283 A BR 9808288 A CN 1254472 A CN 1260056 A CN 1254477 A CN 1254478 A CN 1254469 A CN 1254423 A CN 1262754 A CN 1254473 A CN 1254422 A CN 1254475 A CN 1254476 A CN 1254474 A CN 1255266 T CN 1255212 T CN 1255268 T CN 1257630 T WO 9843425 A WO 9843426 A WO 9843162 A WO 9843431 A WO 9843248 A WO 9843165 A WO 9843172 A WO 9843433 A WO 9843427 A WO 9843437 A WO 9843167 A WO 9843428 A WO 9843421 A EP 0872798 A EP 0866611 A EP 0866616 A EP 0866613 A EP 0968610 A EP 0968609 A EP 0968607 A EP 0974229 A EP 0974230 A EP 0968468 A EP 0968465 A	20-10-1998 05-01-2000 22-11-1999 22-05-2000 20-10-1998 20-10-1998 16-05-2000 16-05-2000 16-05-2000 24-05-2000 12-07-2000 24-05-2000 24-05-2000 24-05-2000 24-05-2000 09-08-2000 24-05-2000 24-05-2000 24-05-2000 24-05-2000 24-05-2000 31-05-2000 31-05-2000 31-05-2000 21-06-2000 01-10-1998 01-10-1998 01-10-1998 01-10-1998 01-10-1998 01-10-1998 01-10-1998 01-10-1998 01-10-1998 01-10-1998 01-10-1998 21-10-1998 23-09-1998 23-09-1998 23-09-1998 05-01-2000 05-01-2000 05-01-2000 26-01-2000 26-01-2000 05-01-2000 05-01-2000
US 5835911 A	10-11-1998	JP 7225723 A	22-08-1995

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/22404

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5835911 A		JP 7225724 A	22-08-1995
		JP 9054734 A	25-02-1997
US 5859977 A	12-01-1999	JP 5108317 A	30-04-1993
		AU 656813 B	16-02-1995
		AU 2633692 A	22-04-1993
		AU 674261 B	12-12-1996
		AU 7916894 A	09-02-1995
		CA 2080401 A	19-04-1993
		DE 4235193 A	13-05-1993