



US006467088B1

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

(10) **Patent No.:** US 6,467,088 B1
(45) **Date of Patent:** Oct. 15, 2002

(54) **RECONFIGURATION MANAGER FOR CONTROLLING UPGRADES OF ELECTRONIC DEVICES**

WO WO9015394 6/1990 G06F/15/46
WO WO9425923 11/1994 G06F/15/21
WO WO9632679 10/1996 G06F/13/00

(75) Inventors: **Yasser alSafadi**, Yorktown Heights, NY (US); **J. David Schaffer**, Wappingers Falls, NY (US)

(73) Assignee: **Koninklijke Philips Electronics N.V.**, Lindhoven (NL)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/343,607**

(22) Filed: **Jun. 30, 1999**

(51) **Int. Cl.**⁷ **G06F 9/45**

(52) **U.S. Cl.** **717/173; 713/100**

(58) **Field of Search** **717/173, 178, 717/177; 710/10; 713/100**

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,155,847	A	10/1992	Kirouac et al.	395/600
5,253,344	A	* 10/1993	Bostick et al.	710/8
5,327,560	A	* 7/1994	Hirata et al.	709/221
5,497,490	A	* 3/1996	Harada et al.	713/100
5,634,075	A	* 5/1997	Smith et al.	710/9
5,822,531	A	* 10/1998	Gorczyca et al.	707/202
5,898,872	A	* 4/1999	Richley	713/100
5,918,194	A	* 6/1999	Banaska et al.	702/91
5,933,026	A	* 8/1999	Larsen et al.	326/81
6,058,455	A	* 5/2000	Islam et al.	710/10
6,065,068	A	* 5/2000	Foote	710/10
6,167,408	A	* 12/2000	Cannon et al.	707/200
6,301,707	B1	* 10/2001	Carroll et al.	717/177
6,385,668	B1	* 5/2002	Gadness et al.	370/254

FOREIGN PATENT DOCUMENTS

EP	0308056	* 3/1989	G06F/11/00
GB	2325766	12/1998	G06F/9/44

OTHER PUBLICATIONS

Mitchell et al., Dynamically Reconfiguring Multimedia Components: A Model—Based Approach, Sep. 1998, ACM, p. 40–46.*

* cited by examiner

Primary Examiner—Gregory Morsc

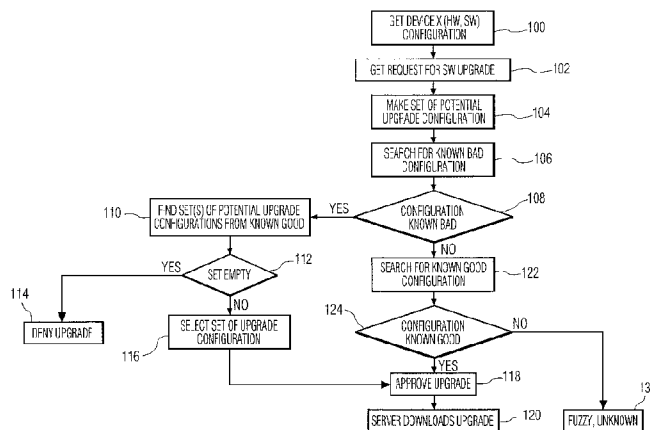
Assistant Examiner—John O. Chavis

(74) *Attorney, Agent, or Firm*—Daniel J. Piotrowski

(57) **ABSTRACT**

A reconfiguration manager implemented on a computer or other data processing device controls the reconfiguration of software or other components of an electronic device such as a computer, personal digital assistant (PDA), set-top box, television, etc. The reconfiguration manager receives a reconfiguration request, e.g., a software upgrade request from the electronic device, and determines one or more device components that are required to implement the reconfiguration request. The reconfiguration manager also determines, e.g., from information in the request, identifiers of one or more additional components currently implemented in the electronic device. The reconfiguration manager then compares the needed and currently implemented components with previously-stored lists of known acceptable and unacceptable configurations for the electronic device. If the needed and currently implemented components correspond to a configuration on the list of acceptable configurations, the request is approved and the needed components are downloaded to the electronic device. If the needed and currently implemented components correspond to a configuration on the list of unacceptable configurations, the request is denied. Otherwise, the reconfiguration manager may indicate that the requested reconfiguration is unknown, or may take another action such as responding to the electronic device with a list of other components that would be required to implement the request.

21 Claims, 3 Drawing Sheets



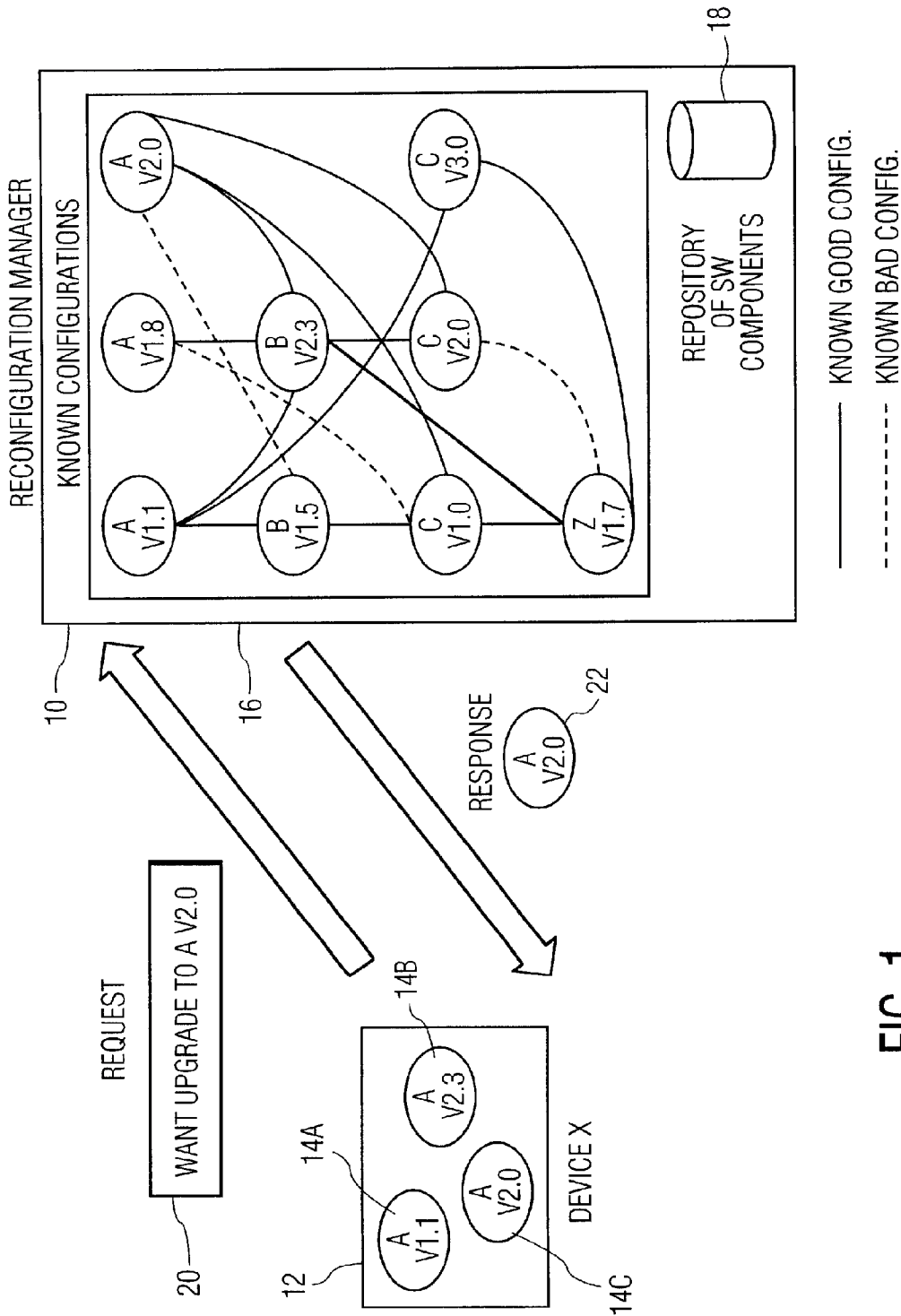


FIG. 1

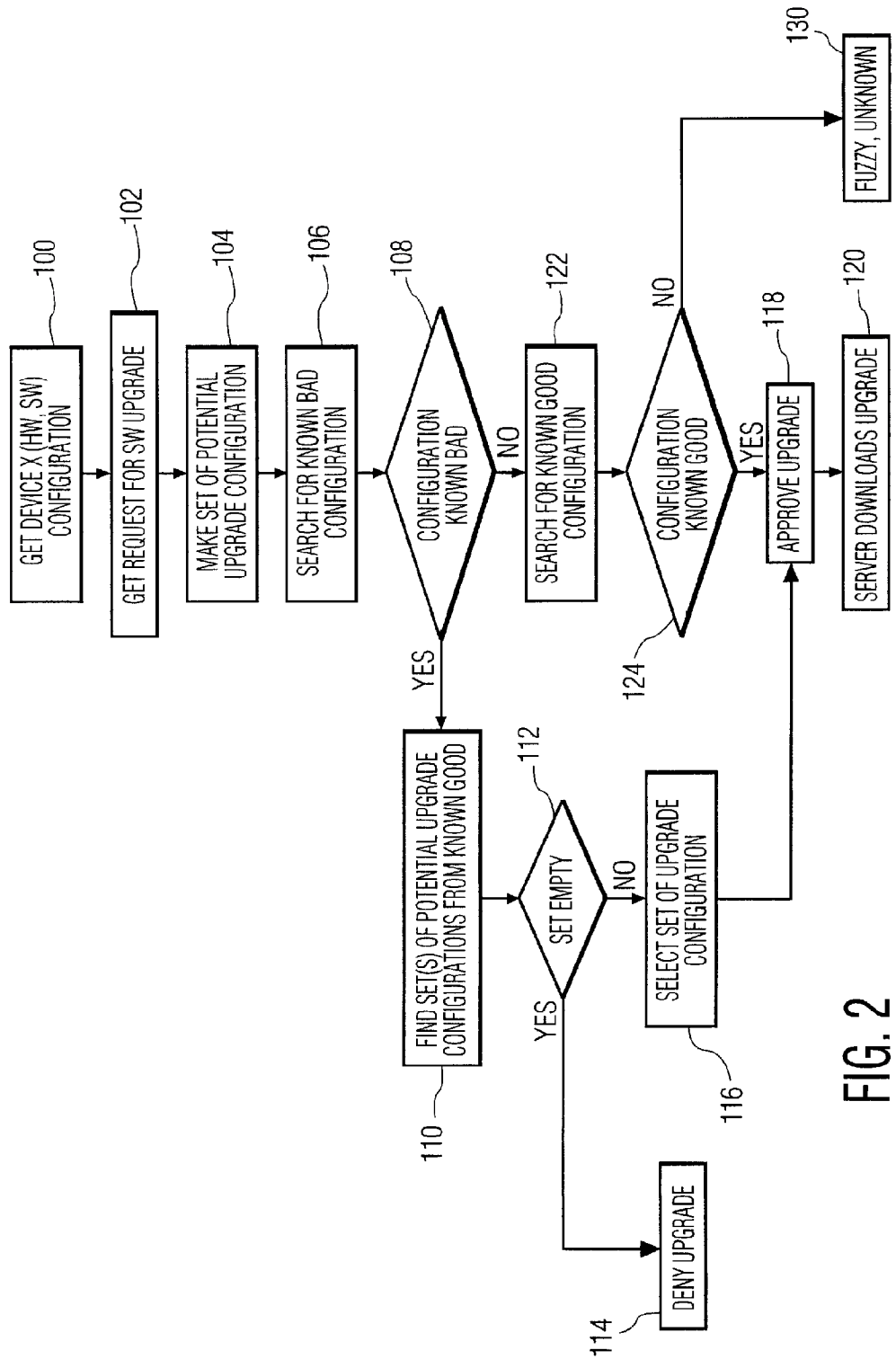


FIG. 2

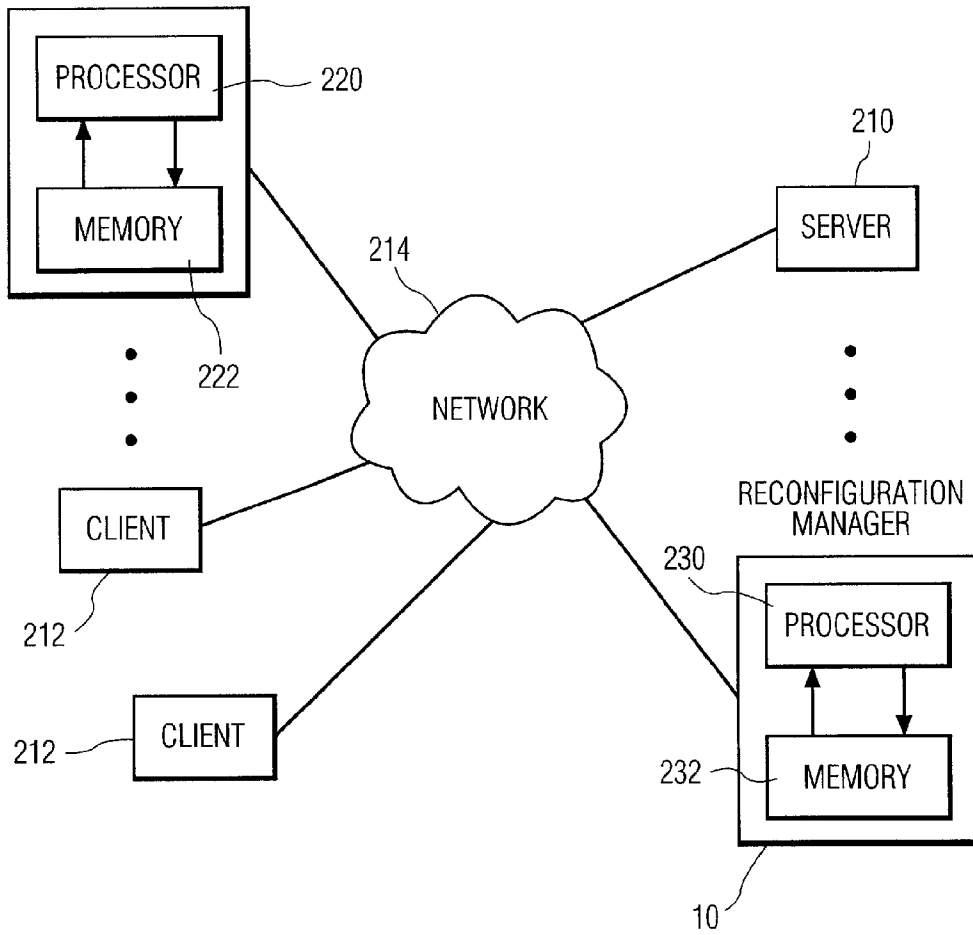


FIG. 3

RECONFIGURATION MANAGER FOR CONTROLLING UPGRADES OF ELECTRONIC DEVICES

FIELD OF THE INVENTION

The present invention relates generally to the field of electronic devices, and more particularly to techniques for upgrading or otherwise reconfiguring software and/or hardware components in such devices.

BACKGROUND OF THE INVENTION

For many different electronic devices, such as desktop, laptop and palmtop computers, personal digital assistants (PDAs), telephones, televisions, set-top boxes and other consumer electronic processing devices, it is common for ongoing development efforts to continue to produce improvements to existing device software or hardware components, as well as new components that add to or otherwise improve device functionality. Users of such devices often prefer to upgrade their devices, incrementally, rather than discard their current devices and purchase new ones. However, for most contemplated upgrades, it is generally necessary to determine if the new or improved component is compatible with the rest of the device, and if not, what other components would need simultaneous upgrading in order to provide the desired compatibility. This compatibility determination can be particularly difficult if the range of possible device configurations is large and the interaction among device components is complex.

A number of different techniques have been developed for updating components of electronic devices. For example, U.S. Pat. No. 5,155,847 discloses a technique for updating software at remote locations. A central computer system stores the original software, and keeps track of all the software configurations for a number of remote systems. The remote system software is upgraded or otherwise changed based on patches transmitted by the central computer system. However, this technique generally requires the central computer system to keep track of the particular software configurations at each of the remote systems. Furthermore, the technique is not directly applicable to electronic devices other than computers, and cannot efficiently handle reconfiguration of hardware components, or hardware and software interdependencies.

Another conventional technique, described in PCT Application No. WO 94/25923, manages the configuration of an enterprise-wide network which includes at least one centralized computer and a plurality of desktop computers. The technique attempts to ensure that each of the desktop computers has an appropriate set of resources as determined in accordance with a set of enterprise policies. However, the technique generally assumes that the resources required by each desktop computer are independent, and fails to adequately address situations in which the required resources are highly interdependent. Furthermore, this technique generally assumes that the information regarding component interactions is fully specified and built in to the system.

UK Patent Application No. GB 2,325,766 discloses a version management system for keeping files on remote devices updated to latest versions as determined by a master list maintained on a central server. The updating process in this approach generally involves adding, amending and deleting files in their entirety. A significant problem with this approach is that it apparently assumes either that the files are independent or that any potential conflicting requirements

have already been resolved using other techniques. It fails to provide generalized techniques for ensuring compatibility among requested components.

A convention technique disclosed in PCT Application No. WO 96/32679 describes the remote patching of operating code in a mobile unit of a distributed system. A manager host device in the system transmits patches to the mobile unit, and the mobile unit creates patched operating code by merging the patches with current operating code and switching execution to the patched operating code. However, like the other conventional techniques described previously, this technique also fails to adequately ensure compatibility among software and hardware components for a variety of different electronic devices.

As is apparent from the above, a need exists for improved techniques for managing reconfiguration of electronic devices, such that compatibility determinations can be facilitated, particularly for large and complex device configurations.

SUMMARY OF THE INVENTION

The invention provides a reconfiguration manager that may be implemented on a computer or other data processing device to control the reconfiguration of software or other components of an electronic device such as a computer, personal digital assistant (PDA), set-top box, television, etc. In accordance with the invention, a reconfiguration manager receives a reconfiguration request, e.g., a software upgrade request from the electronic device, and determines one or more device components that are required to implement the reconfiguration request. The reconfiguration request can be received directly from the electronic device itself, or otherwise supplied to the reconfiguration manager.

The reconfiguration manager also determines, e.g., from information supplied by the electronic device as part of the request, identifiers of one or more additional components currently implemented in the electronic device. The reconfiguration manager then compares the needed and currently implemented components with previously-stored lists of known acceptable and unacceptable configurations for the electronic device. If the needed and currently implemented components correspond to a configuration on the list of acceptable configurations, the request is approved and the needed components are downloaded or otherwise supplied to the electronic device. If the needed and currently implemented components correspond to a configuration on the list of unacceptable configurations, the request is denied. Otherwise, the reconfiguration manager may indicate that the requested reconfiguration is unknown, or may take another action such as responding to the electronic device with a list of other components that would be required to implement the reconfiguration request.

Advantageously, the invention provides efficient techniques for incrementally upgrading or otherwise reconfiguring electronic devices. The invention ensures that upgrades are compatible with the configuration of a given device before they are implemented in that device, thereby avoiding problems associated with inconsistent upgrades. Although particularly well suited for use with software upgrades delivered over a network, the invention is applicable to reconfiguration of other types of device components, e.g., hardware components or combinations of hardware and software components, and to numerous other applications. These and other features and advantages of the present invention will become more apparent from the accompanying drawings and the following detailed description.

Explore Litigation Insights

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

Real-Time Litigation Alerts



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

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

Advanced Docket Research



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

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

Analytics At Your Fingertips



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

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

API

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

LAW FIRMS

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

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

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.