Contact Information

Publications@ipcg.com

ipCapital Group, Inc.
400 Cornerstone Drive, Suite 325
Williston, VT 05495
United States of America
(802) 872-3200 x214

Title of the Invention

Method and System for Updating Software from a Remote Service Computer

Description of the Invention

This invention describes a system where a service machine installs the most recent version of a given piece of software, regardless of what version is currently installed. This provides for an automatic update system where no information aside from the name of the program needing the update must be transmitted to the service machine.

Problem/Opportunity

Currently in high-tech devices, software updates are still a process initiated by the user of the device, that is, software on the user machine contacts a remote maintenance machine that inventories what software is installed and what version is installed on the user machine. The service machine then in turn identifies software in need of updates or maintenance, and finally downloads and installs the new updates or software on the user machine. However, such a process often leads to user errors or mistakes, potentially leading to data loss, user frustration, or even a lack of software usability.

What is needed is a system and method for updating software on user machines automatically, without the need for intervention by the user machine or the user, and without the need to check whether the version is up-to-date or not. Ideally, when new updates are available, the service machine would initiate the update process, contact the user machine, and apply the updates remotely. If this system is implemented by constantly overwriting old software with the newest version, it is also imperative that updates do not cause user preferences or data to be lost in the process.

Detailed Description of the Invention

This invention describes a system where, rather than having, or even needing, an "inventory" of currently installed versions on the user machine, a service machine instead just installs the most recent version of a given piece of software, regardless of what version is currently installed. More specifically, this provides for an automatic update system where no information aside from the name of the program needing the update must be transmitted to the service machine. This system is based on updates for any given User Machine (1.1), consisting of a processing device with an internet or network connection (such as a PC, laptop, cellular phone etc). An operating system such as Linux (not shown) is installed on the user machine. This user machine must also have an IP Address (1.2) as a means for the Service Machine to connect to it. Updates are provided by a Service Machine (1.3), which can be classified as a maintenance server located on the Internet or network.

Firstly, the user machine must connect to the service machine, which may be initiated by either the user machine (in the case the service machine has no means of locating the user machine on its own) or by the service machine contacting the user machine (assuming the service machine has an address to communicate with the user machine). If automatic updates are turned on, the user machine, on connection to the internet, would transmit its IP address to the service machine, and then return to whatever job is being done on the user machine. The User may also choose the frequency with which software may be overwritten. For example, the User could specify to search for and install updates every night at midnight, or every weekend. This gives the user control over when the overwrite process will occur. When the Service Machine is ready to service the User Machine, it connects to the User machine using the provided IP address. This would also allow the service machine to have a means of contacting the user machine should a new update become available, eliminating the need for the user machine to initiate the update process. Next, the user machine will identify what piece of software it would like to update. The service machine, in response, installs the most current version of Program X on the user machine, regardless of what version of Program X is currently installed. This process can also be made iterative by allowing the user machine to go through programs one after another to request updates for them. When the user machine is completely updated, it disconnects from the service machine.

In operation, the system would work as follows

2.1 Establish a connection between the service machine and user machine.
2.2 The user machine transmits to the service machine a piece of software that is currently installed and would like to be updated.
2.3 The service machine proceeds to install the most current version of the software onto the user machine.
2.4 Steps 2.2 and 2.3 are repeated until all software is updated.

A particularly important aspect of the system is that it is developed in such a way that forced overwrites of the current program may cause user preferences and settings to be lost in the process. Currently, some program updates work by only injecting some code into the current application, or by simply changing a file or attribute of a program, which fixes the problem the update is meant to address. Programs updated in this way would not be affected by the problem mentioned. However, should a program be updated by completely replacing its code with new code, there is the possibility that some user preferences may be overwritten; for example, if user data is located in the same file as real world data parameters. It is critical that, in the updating of programs using this system, that the data portion of the program (e.g. program data files, preference files, etc) are not altered, but rather only the executable portion of code involved in the program. Naturally, some exceptions may be made to this rule should there be a need for a change to real world data parameters (for example, exchange rates in banking software). In the case of updating particular data fields, if historical data is not needed, then the data files may be replaced outright. However, should historical data be needed, the new data file should be added to the system without deleting or outright replacing the obsolete file. Rather, the program should be updated to recognize the new data file as the most current, preserving historical values while keeping fields current.

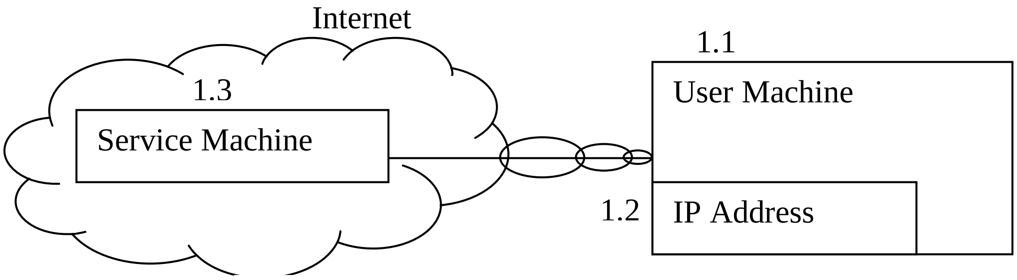

Figure 1: System Diagram

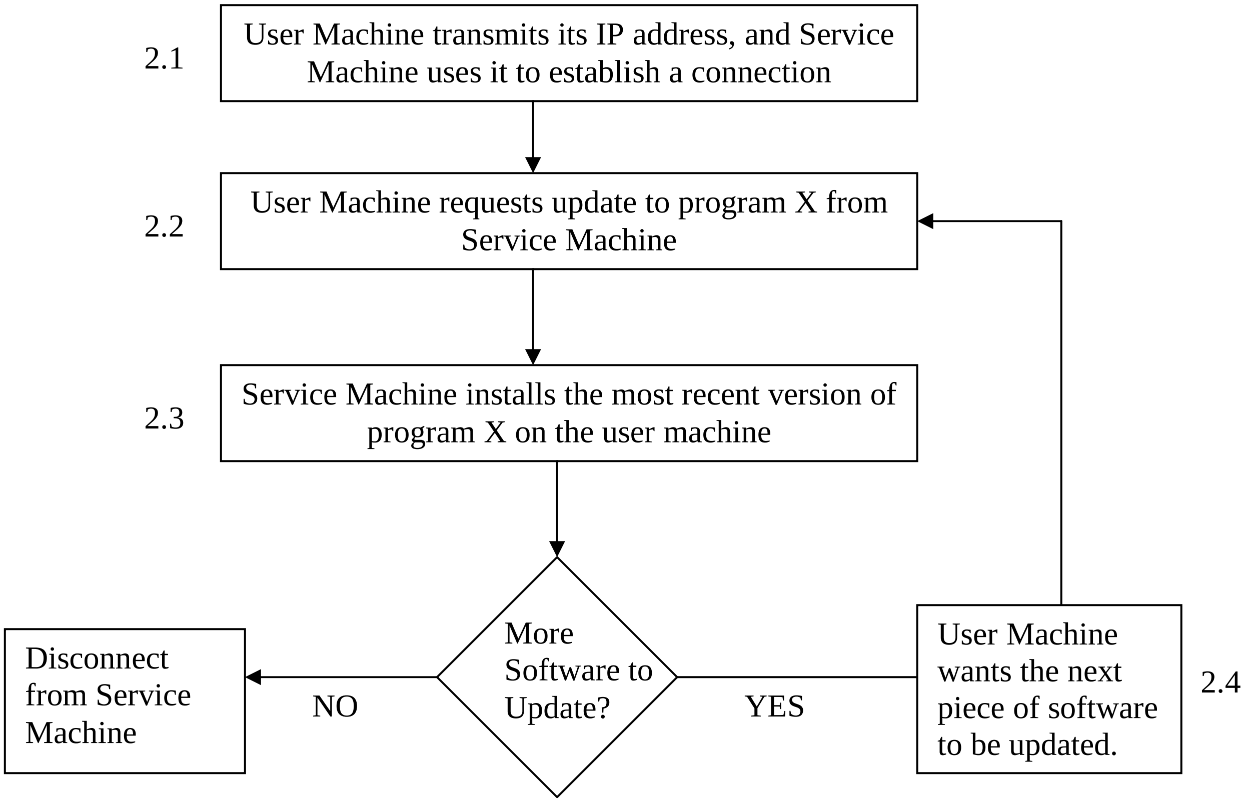
2.1
User Machine transmits its IP address, and Service Machine uses it to establish a connection
2.2
User Machine requests update to program X from Service Machine
2.3
Service Machine installs the most recent version of program X on the user machine
More Software to Update?
NO
Disconnect from Service Machine
YES
User Machine wants the next piece of software to be updated.
2.4

Figure 2: Operational Flowchart