Software Application Control with Flexible Activation and Deactivation of Application Features

Field of the Invention

[0001] The present invention relates to software application control. In particular, the invention relates to a system and method for controlling accessibilities of software applications of computer terminals connecting to a network, including Internet and intranet.

Background

[0002] Trial software products or shareware products are often limited in time or in feature until a full version is purchased. These products are designed with a time-based expiry mechanism or feature disabling mechanism or a combination of both to control those limitations. The time-based expiry mechanism checks against the computer system date (i.e. current date) with a preset expiry date to disable the software applications upon expiry. For example, when a 30-days trial software is installed in a computer, the day of installation is recorded and a preset date, i.e. 30 days from the installation date is determined. Upon expiry of the trial software, the software is rendered unusable/disable. The feature disabling mechanism disables the software product partially by disabling only certain features of the software product without expiry date. For example, a trail version graphic editing software can be used without expiry date, but the save and print features are disabled. Accordingly, a user may experience all the features offered by the software, but the user may not be able to save print the edited work.

[0003] The limitations can usually be removed by entering a unique serial key obtained through a registration of that software product. These mechanisms are used commonly in software marketing industry, where the developers do not require flexible enabling/disabling features for their products. These mechanisms are not applicable for

limiting the accessibilities of licensed software applications that are already installed in a local computer.

[0004] In a computer network, users of the computers may freely utilize the licensed software applications that are installed on the local computers. In some occasion, there may be a need to restrict the usage of these software applications. For example, when a group of students doing their exam in a computer lab where only some computer applications are allowed for the exam. That requires the lab assistants to configure all the computers to disable the prohibited applications or features in advance, and after the exam, the lab assistants required to restore functionalities of the applications or features.

[0005] There is accordingly a desire to provide a system and method for controlling software applications or features accessibilities for computer terminals in a computer network in real-time.

Summary

[0006] In accordance with one aspect of the present invention, there is provided a system for controlling accessibilities of features executable and remaining on a terminal over a network having a control terminal and a plurality of client terminals, the system comprises a control agent for residing in the control terminal operable to define instructions for controlling the accessibilities of features and broadcast the instructions across the network; and a client agent for residing in the client terminals operable to listen to the broadcast from the control agent and execute the instructions to intercept activities in the client terminals to control the accessibilities and processes of features.

[0007] In accordance with one embodiment, the instructions may include user/node IDs that correspond to the client terminals to be controlled. The instructions may also include information of features to be controlled. The instructions may further

include a time period for executing the instructions. These instructions may be broadcasted in a form of message.

[0008] In accordance with another embodiment, the system may further comprise a database accessible by the control agent for storing user/node IDs. It may also comprise a database accessible by the control agent for storing features characteristics/signatures for accompanying with the instructions.

[0009] In accordance with yet another embodiment, the control of accessibilities of features may include permitting features to be executed in the client terminal. The control accessibilities of features may include rejecting features to be executed in the client terminal in the client terminal.

[0010] In accordance with yet another embodiment, the control of accessibilities of features may include intercepting running processes and shuts down the processes.

[0011] In accordance with yet another embodiment, it is possible that the features are software applications. Further, it is also possible that the network includes Internet and intranet.

[0012] In accordance with another aspect of the present invention, a method for controlling accessibilities of features executable and remaining on a terminal over a network having a control terminal and a plurality of client terminals, said method comprises defining instructions for controlling the accessibilities of features at the control terminal; broadcasting the instructions across the network; listening to the broadcasted instructions at the client terminals; and executing the instructions to intercept activities in the client terminals to control accessibilities of features.

[0013] In accordance with one embodiment, the instructions may include user/node IDs that correspond to the client terminals to be controlled. The instructions may also include information of features to be controlled. The instructions may further

include a time period for executing the instructions. These instructions may be broadcasted in a form of message.

[0014] In accordance with another embodiment, the method may further comprise accessing a database by the control agent for obtaining user/node IDs. It may also comprise accessing a database accessible by the control agent for obtaining features characteristics/signatures for accompanying with the instructions.

[0015] In accordance with yet another embodiment, the method may further comprise permitting features to be executed in the client terminal. The method may further comprise rejecting features to be executed in the client terminal in the client terminal.

[0016] In accordance with yet another embodiment, it is possible that the method comprises intercepting running processes and shuts down the processes. It is also possible that the method comprises receiving broadcasted user/node IDs from the control agent. Yet, it is possible that the method comprises matching the broadcasted user/node IDs with a user/node ID of the client terminal. The method in the aforesaid embodiment may further comprise acquiring information of characteristics/signatures of the features to be controlled from the control agent; and acquiring time period for executing the instructions.

[0017] In accordance with the aforesaid embodiment, the features may be software applications. It is also possible that the network includes Internet and intranet.

Brief Description of the Drawings

[0018] This invention will be described by way of non-limiting embodiments of the present invention, with reference to the accompanying drawings, in which:

[0019] FIG. 1 illustrates network of a plurality of terminals having an access control system in accordance with one embodiment of the present invention;

[0020] FIG. 2 illustrates a block diagram of an access control system in accordance with one embodiment of the present invention;

[0021] FIG. 3 illustrates a flow chart for performing an access control process in accordance with one embodiment of the present invention; and

[0022] FIG. 4 exemplifies a schematic block diagram of access control perform by the access control system in accordance with one embodiment of the present invention.

Detailed Description

[0023] In line with the above summary, the following description of a number of specific and alternative embodiments are provided to understand the inventive features of the present invention. It shall be apparent to one skilled in the art, however that this invention may be practiced without such specific details. Some of the details may not be described at length so as not to obscure the invention. For ease of reference, common reference numerals will be used throughout the figures when referring to the same or similar features common to the figures.

[0024] According to one embodiment, the invention provides a system and method to fully or partially control accessibilities of features for software applications for computer terminals over a computer network. The system can be used to schedule a period for implementing the access control to a specific group of users or nodes. The access control can be performed in real-time within the Internet or intranet environment. The system provides a control agent for defining and broadcasting instructions for the access control, and a client agent residing in the computer terminals for listening to the broadcasted message. The client agent executes the instructions to control accessibilities of software applications in the computer terminals.

[0025] FIG. 1 illustrates a network **100** of a plurality of terminals having an access control system in accordance with one embodiment of the present invention. The plurality of terminals connect to the network **100** via standard connections, such as LAN, wireless LAN, WAN and etc. The network may be an intranet or Internet. The access control system is adapted to control software accessibilities fully and partially at the terminals. The software includes any software applications such as word processing application, graphic processing application, and etc. The Access control system is also adapted to control users' accessibilities to all features/processes, commands/instructions and the like that are executable on the terminals. In full control, the access control system activate/deactivate the entire software, whilst in partial control, the access control system activate/deactivate only some features/processes of the software applications. For simplicity, the software

applications, the executable features/processes and commands/instructions that provide features to enable a computer device to perform specific tasks are herewith referred to as "the features". Similarly, the operations of controlling the features' accessibility in any terminals are herein referred to as "the access control".

[0026] Still referring to FIG. 1, the network **100** comprises a control terminal **110** and client terminals **120**. The access control system comprises an agent model (not shown in FIG. 1) for installing in each terminal. In an event where the access control is required, the agent model of the control terminal **110** broadcasts commands/instructions in a form of message to the client terminals **120** via the network **100**. Upon receiving the message, the agent models reside in the client terminal **120** executes the instructions/commands and screens all the activities/processes that are to be executed by the client terminals **120** based on the instructions/commands given in the message. Depending on modes of the access controls, the agent model of the client terminals **120** permits/prohibits the features to be executed based on the message received. For example, when the message contains instructions/commands that prohibits certain features, the agent model of the terminals **120** rejects the prohibited features upon detecting it while screening the activities/processes. The message comprises user/node IDs of the terminals **120** to be controlled, commands/instructions for the access control and a validity period of that commands/instructions, and possibly encryption key information.

[0027] FIG. 2 illustrates a block diagram of an access control system **200** for the network **100** of FIG. 1 in accordance with one embodiment of the present invention. The access control system **200** comprises a control agent **210**, a user/node ID data **212**, an application control data **214** and a client agent **220**. The control agent **210**, the user/node ID data **212** and the application control data **214** are resided in the control terminal **110**, and the client agent **220** is resided in each of the client terminals **120** that comprises a plurality of features **222** and a system registry/hidden file. The control agent **210** and the client agent **220** are agent models/programs that reside in the control terminals **110** and **120** for performing the access control. The control agent **210** provides functions for defining instructions/commands for the access control and

broadcasting the instructions/commands to the client terminal **120** in a form of message. The control agent **210** comprises a user/node IDs lister **211**, validity period scheduler **213** and a feature lister **214**. The user/node IDs lister **211** provides functions and platforms for entering and selecting the user/node IDs. The user/node IDs lister **211** is accessible to the user/node ID data **212** for selecting the listed user/node ID and inputting new user/node IDs. The validity period scheduler **213** provides functions and platforms for entering and selecting validity period for each individual access control. The feature lister **214** provides functions and platforms for entering and selecting the features to be controlled. The feature lister **214** is accessible to the application control data **214** for selecting the listed features and importing new features. The user/node ID data **212** stores user IDs or terminal's node IDs of the network **100** that are selectable by the control agent **210** for identifying the user/node IDs that correspond to the client terminals **120** to be controlled. The application control data **214** defines information and characteristics/signatures of the features for accompanying with the instructions/commands for identifying the features to be controlled. The client agent **220** listens to the broadcast from the control agent **210** and performs the access control based on the instructions/commands defined.

[0028] Still referring to FIG. 2, when the client terminals **120** are started up, the client agents **220** stay in the background of the client terminals **120** to monitor all features **222** that runs on the client terminals **120** and intercepts the features **222** that are to be controlled. When the access control is required within the network **100**, the control agent **210** is used to define user/node IDs of the designated client terminals **120** to be controlled, the instructions/commands for the access control, and a validity period for the access control. The user/node IDs may include user ID, IP address, and any any identifications of users or terminals are selectable from the user/node ID data **212**. Accompanying the instructions/commands, the features **222** to be controlled are also specified in a form of parameters of instructions/commands. These features **222** are available for selections from the application control data **214**. The instructions/commands include prohibiting-type instructions, permitting-type of instructions or a combination of both. The prohibiting-type instructions specify what are the features **222** that are not allowed. The permitting-type instructions specify what

are the features **222** that are allowed. Once the message is defined, the control agent **210** broadcasts the message to the client agent **220** across the network **100**. When the client agent **220** receives the message, it screens through the activities/processes of the client terminal **120** and controls the accessibility of the features **222** based on the instructions/commands and validity period defined in the message. During the validity period, the client agent **220** provides two tiers operations for controlling access. First, the client agent **220** sends messages to the relevant software applications to enable/disable the features **222** that are to be controlled. When the features **222** are disable, the relevant features **222** become unselectable by the user therefore achieving the access control. Second, the client agent **220** scans and intercepts all processes that are running by the client terminal **120**. This operation provides controls over the features **222** that cannot be done by the first tier operation. The scanning and intercepting allow the client agent **220** to control the features **222** that are already running on the client terminal **120** before the access control is activated. The client agent **220** further intercepts the client terminal's **120** processor to control the features **222** that have bypassed the scanning of the client agent **220**. When the user of the client terminal **120** attempts to activate any software applications or features **222**, the client agent **220** intercepts the process to see if the relevant features **222** are permitted/prohibited, based on the message. If it is a prohibiting feature, the client agent **220** denies the feature to be processed by the client terminal **120**. In operations, the client agent **220** further monitors status of active features **222** in the client terminal **120** and sends the status back to the control agent **210**. When any prohibited features **222** are detected active, the client agent **220** intercepts the features' process and shuts down those features **222** immediately.

[0029] Still referring to FIG. 2, the application control data **214** is pre-defined and stored in a database to form parameters for defining instructions/commands for the access control. The information and characteristics/signatures of the features **222** can be obtained directly from the features **222** providers via the relevant technical manuals or support websites, or by the any third party software, such as Spy++ from Microsoft, or the like that automatically scans and reveals all features' processes, threads and messages as they run on the terminals. The application control data **214** requires

updating for new software applications or features **222** are provided in the client terminals **120**. With the application control data **214** provided, the access control may be performs by using the appropriate commands with parameters that specifying the features **222** and period. The examples below exemplify some commands and parameters:

[0030] Example 1: <<command>> [tehah, winword, 09-02-2007, 1530, 60, 0]

[0031] Example 1 provides a syntax that include a command and parameters. The <<command>> represents the command to be executed for the access control. The command may be a prohibiting-type of instructions or permitting-type of instructions as described above. Following which, are parameters that described how the command should behave. The parameters start with a user/node ID "tehah" that tells which are the client terminals **120** that are to be controlled. This parameter is user-specific and it specifies that the access control should be performed on any client terminal **120** with the user ID "tehah" logged in. In the case where a specific client terminal **120** is to be controlled, a node ID such as IP address of the client terminal **120** may be used. Then, the parameter "winword" specifies the Microsoft Word (MSWORD) be controlled on the client terminal **120**. The subsequent three parameters specifies the time and period, in which, the above syntax is to be activated on 9 February 2007, 1530hr for a period of 60 minutes (1 hour). A zero right at the end of the parameters shows that the syntax is to apply on the entire Microsoft Word application.

[0032] Example 2: <<command>> [windows, tehah, winword, 09-02-07, 1530, 60, WM_COMMAND, 0200, 09114E]

[0033] Example 2 also provides a syntax that includes a command and parameters. Similarly, the <<command>> represents the command to be executed for the access control. The parameters start with "windows" that represent this instruction applies to Windows-Based Operating System. The subsequent five parameters have the same functions as that of Example 1. Thereafter, the three parameters specify a

specific feature of in the MSWORD is to be controlled rather than the entire software program. "WM_COMMAND, 0200, 09114E" indicates that the print feature in MSWORD is to be controlled. These parameters are provided from the application control data **214**.

[0034] Example 3: `[windows, tehah, winword, 09-02-07, 1530, 60, WM_COMMAND, 0400, 0120F76]`

[0035] Example 3 provides only one parameter syntax of which a commands may be provided for controlling a plurality of parameter syntaxes. The parameter syntax provides parameters that start with "windows" that represent this instruction applies to Windows-Based Operating System. The subsequent five parameters have the same functions as that of Example 1. Thereafter, the three parameters specify a specific feature of in the MSWORD is to be controlled rather than the entire software program. "WM_COMMAND, 0400, 0120F76" indicates that the save feature in MSWORD is to be controlled. It is possible that Boolean expressions may be used in the command to join the plurality of parameter syntaxes for performing access control in a more complex manner.

[0036] Most commonly, one command is to be applied to a plurality of user/note ID. Accordingly, the parameter specifying the user/node ID may be a link to a list rather than a specific user/node ID. Similarly, the other parameters may also contain a link to lists with a group of parameters of the same category.

[0037] The application control data **214** in accordance with the aforesaid embodiments is provided at the control terminals **110** where the control agent **120** is resided for easy references and selections. It is understood to the skilled person that the application control data **214** may be resided remotely from the control agent, such as a server that the control terminal **110** is connected to.

[0038] In the above embodiment, the control agent **210** is adapted to provide capabilities of defining the instructions/commands for the access control and

broadcasting the instructions/commands within the network **100**. It is possible that the control capabilities are also provided in the client agent **220** that make every client terminal **220** also a control terminal so that the access control can be accessed anywhere within the network **100**. In such case, there may require a security system for accessing the control capabilities. Further, it is also possible that the user/node ID data **212** and the application control data **214** are hosted by the control terminal **110**. It is understood to the skilled person that the data **212** and **214** may be stored in the client terminal **120** for accessing by the client agent **220**.

[0039] FIG. 3 illustrates a flow chart for performing an access control process in accordance with one embodiment of the present invention. When an access control is required within the network **100**, control agent hosted by the control terminal **110** is launched and the user/node IDs that are to be controlled are defined at step **310**. The control agent **210** generates a message containing the user/node IDs and broadcasts it among the network **100** in step **320**. In step **330**, the client agent **220** of the client terminals **120** checks their own user/node ID with the user/node IDs defined in the message upon receiving the broadcasted message. When the user/node ID of the client terminals **120** is matched, the client agents **220** of the client terminals **120** connect to the control agent **210** of the control terminal **110** at step **340**. For those user/node ID of the client terminal **120** that does not match with the user/node IDs defined in the message, the client terminal **120** discards the message and listens for other broadcasted messages. At step **350**, the matching client terminals **120** connect to the control agent **220** to retrieve corresponding instructions/commands for the access control and validity periods of the instructions/commands. When the instructions/commands is to enable the features fully or partially at step **360**, instructions/commands are kept in system registry or as a hidden file for storing in the terminal **120**. These instructions/commands are kept for monitoring logging based on the instructions provided along with the message in step **380**. Further, should the terminals are rebooted or interrupted, the client agent **220** may recall the instructions/individuals from the stored system registry or hidden file without the need for retrieving the same from the control agent **210** again. When the instructions/commands is to disable the features at step **370**, the client agent **210** prohibits and rejects all features that are

prohibited. In steps **360** and **370**, whether the instructions/commands are prohibited or permitted, the client agents **220** monitor status of the access control in step **390** and interrupt any processes that are omitted. The control agent **210** may further retrieve the status from each client terminal **220**, when necessary.

[0040] In the aforesaid embodiment, the message is broadcasted with only the user/node IDs only, and the client terminals **120** acquire the necessary instructions/commands upon receiving the broadcasted message. In accordance with an alternative embodiment, the message carries the user/node IDs and the necessary instructions/commands together and the client agents **220** execute the instructions/commands without the need for steps **340** and **350**. Further, when a different user accessing the client terminal **120** with the access control period, the client agent **220** may retrieve the instructions/commands from the system registry or hidden file, where applicable.

[0041] FIG. 4 exemplifies a schematic block diagram of the access control performed by the access control system **200** in accordance with one embodiment of the present invention. The client agent **220** resides on the client terminal **120** listening to any broadcast from the control agent **210**. The client terminal **120** comprises software A, software B, software C, and software D that can be executed by the Center-Processing-Unit (CPU) **400** of the client terminal **120**. In use, the control agent **210** accesses the user/node ID data **212** and the application control data **214** to define the user/node IDs and instructions/commands for the access control. The control agent generates a message **410** for broadcasting it to the client terminals **120** via the Internet **490**. Upon receiving the message **410**, the client agent **220** authenticates the message with security measures, such as any encryptions methods. The message **410** contains the user/node IDs of the client terminal **120** to be controlled. The client agent **220** of the client terminal **120** connects to the control agent **210** via the Internet **490** to retrieve corresponding instructions/commands from the control agent **210**. Based on the instructions/commands given by the control agent **210**, the client agent **220** screens all the activities/processes in the client terminal **120** and controls the accessibilities accordingly. For example, the control agent **210** has been configured to permit the

entire software A (full features) and feature B of the software B, and prohibit feature A of software B and the entire software C and D within 5 hours every of a particular week. During that period, the client agent **220** intercepts all the activities/processes that are to be executed by the CPU **400**. Accordingly, any attempts to execute feature A of software B and the entire software C and D on the CPU **400** will be rejected and only the software A and feature B of the software B are permitted to be executed.

[0042] Still referring to FIG. 4, the client agent **220** performs a real-time monitoring **420** for all the activities/processes that are running on the CPU **400**. When a prohibited feature is found active in the CPU **400**, the client agent **220** intercepts and shuts down the prohibited feature in real-time. For example, the software D is not screened by the client agent **220** and it is active in the CPU **400**. When the client agent **220** performs the real-time monitoring **420**, the client agent **220** intercepts and shuts down the software D upon detection. Therefore, during the periods of access control specified by the control agent **210**, an output **450** of the client terminal **120** allows only software A (full features) and feature B of software B to be executed.

[0043] While specific embodiments have been described and illustrated, it is understood that many changes, modifications, variations and combinations thereof could be made to the present invention without departing from the scope of the invention.

Claims

1. A system for controlling accessibilities of features executable and remaining on a terminal over a network having a control terminal and a plurality of client terminals, the system comprising:

a control agent for residing in the control terminal operable to define instructions for controlling the accessibilities of features and broadcast the instructions across the network; and

a client agent for residing in the client terminals operable to listen to the broadcast from the control agent and execute the instructions to intercept activities in the client terminals to control the accessibilities and processes of features.

2. The system according to claim 1, wherein the instructions include user/node IDs that correspond to the client terminals to be controlled.

3. The system according to claim 2, wherein the instructions include information of features to be controlled

4. The system according to claim 3, wherein the instructions include a time period for executing the instructions.

5. The system according to claim 4, wherein the instructions is broadcasted in a form of message.

6. The system according to claim 1, the system further comprises a database accessible by the control agent for storing user/node IDs.

7. The systems according to claim 1, the system further comprises a database accessible by the control agent for storing features characteristics/signatures for accompanying with the instructions.

8. The system according to claim 1, wherein the control of accessibilities of features includes permitting features to be executed in the client terminal.

9. The system according to claim 1, wherein the control accessibilities of features includes rejecting features to be executed in the client terminal in the client terminal.

10. The system according to claim 1, wherein the control of accessibilities of features includes intercepting running processes and shuts down the processes.

11. The system according to claim 1, wherein the features are software applications.

12. The system according to claim 1, wherein the network includes Internet and intranet.

13. A method for controlling accessibilities of features executable and remaining on a terminal over a network having a control terminal and a plurality of client terminals, said method comprising:

defining instructions for controlling the accessibilities of features at the control terminal;

broadcasting the instructions across the network;

listening to the broadcasted instructions at the client terminals; and

executing the instructions to intercept activities in the client terminals to control accessibilities of features.

14. The method according to claim 13, wherein the instructions include user/node IDs that correspond to the client terminals to be controlled.

15. The method according to claim 14, wherein the instructions include information of features to be controlled

16. The method according to claim 15, wherein the instructions include a time period for executing the instructions.

17. The method according to claim 16, wherein the instructions is broadcasted in a form of message.

18. The method according to claim 13, the method further comprises accessing a database by the control agent for obtaining user/node IDs.

19. The method according to claim 13, the method further comprises accessing a database accessible by the control agent for obtaining features characteristics/signatures for accompanying with the instructions.

20. The method according to claim 13, further comprises permitting features to be executed in the client terminal.

21. The method according to claim 13, further comprises rejecting features to be executed in the client terminal in the client terminal.

22. The method according to claim 13, further comprises intercepting running processes and shuts down the processes.

23. The method according to claim 13, further comprises receiving broadcasted user/node IDs from the control agent.

24. The method according to claim 23, further comprises matching the broadcasted user/node IDs with a user/node ID of the client terminal.

25. The method according to claim 24, further comprises:

acquiring information of characteristics/signatures of the features to be controlled from the control agent; and

acquiring time period for executing the instructions.

26. The method according to claim 13, wherein the features are software applications.

27. The method according to claim 13, wherein the network includes Internet and intranet.