

METHOD AND SYSTEM FOR PROVIDING REMOTE ACCESS TO A STATE OF AN APPLICATION PROGRAM

FIELD OF THE INVENTION

[001] The instant invention relates to computer networking and in particular to a method and system for providing remote access to a state of an application via a computer network.

BACKGROUND

[002] Healthcare institutions, such as hospitals and diagnostic clinics, utilize information systems such as Hospital Information Systems (HIS), Radiology Information Systems (RIS), Clinical Information Systems (CIS), Picture Archive and Communication Systems (PACS), Library Information Systems (LIS), Electronic Medical Records (EMR), to name a few. Information stored and processed in such systems includes, for example, patient medical histories, imaging data, test results, diagnosis information, management information, and scheduling information. The information is stored centrally or divided at a plurality of locations of a computer network – typically comprising a client-server architecture. Healthcare practitioners access the patient information or other information at various time instants and locations using sophisticated software application programs to gather, analyze, manipulate, and store data.

[003] For example, using a PACS workstation a radiologist performs an image reading for a diagnosis based on content of diagnostic images and reports the results electronically in a patient application file. Because of the large volume of data and intensive computing requirements, the software used to perform this task typically requires dedicated workstation hardware and high bandwidth network access to the diagnostic images. As a consequence, there is limited or no access to the diagnostic information and analysis capability off-site from the PACS workstation. This limits the ability of radiologists, technicians, and other specialists who are not on-site to be able to gain access to the necessary software and data to provide timely diagnosis, for example, in an urgent situation.

[004] Presently, this problem is typically overcome by providing remote network access to the screen or console of the workstation hardware, or by downloading sensitive data such as, for

example, diagnostic data, to a remote system. In the first case, remote access software does not have any special knowledge of the application programs that are executed and are not able to optimize the presentation of the display of the application program to the user based on the state of the application program –resulting in inefficient use of network bandwidth and poor performance for the remote user. In the second case, the transmission of patient related information to an uncontrolled remote site is a security risk for such sensitive information, and furthermore, in the case of diagnostic imaging data, requires transmission of large amounts of information which is beyond the capacity of many networks.

[005] It is desirable to provide a method and system for providing remote access to the state of an application such that the state of the application is presented to a remote user in a fashion adapted to hardware capabilities of the remote computer.

[006] It is also desirable to provide a method and system for providing remote access to the state of an application such that the state of the application is presented to a remote user absent replication of the application program's data processing.

[007] It is also desirable to provide a method and system for providing remote access to the state of an application wherein transmission of data is substantially reduced.

SUMMARY OF EMBODIMENTS OF THE INVENTION

[008] Accordingly, one object of the present invention is to provide a method and system for providing remote access to the state of an application such that the state of the application is presented to a remote user in a fashion adapted to hardware capabilities of the remote computer.

[009] Another object of the present invention is to provide a method and system for providing remote access to the state of an application such that the state of the application is presented to a remote user absent replication of the application program's data processing.

[0010] Another object of the present invention is to provide a method and system for providing remote access to the state of an application wherein transmission of data is substantially reduced.

[0011] According to one aspect of the present invention, there is provided a method for providing remote access to a state of an application. The method comprises executing an

application program at a server computer for performing an application associated therewith. At a client computer connected to the server computer via a communication network a remote access program is executed for providing remote access to a state of the performed application. At the client computer a client difference program having encoded data indicative of a change of a state of the application last received from the server computer is generated and transmitted to the server computer. At the server computer the client difference program is executed for determining an updated state of the application and a server difference program having encoded a difference between the updated state of the application and the state of the application last sent to the client computer is generated. The server difference program is then transmitted to the client computer and at the client computer executed for updating the state of the application last received from the server computer. Display data indicative of the updated state of the application last received from the server computer are generated and displayed in a human comprehensible fashion on a display.

[0012] According to another aspect of the present invention, there is further provided a storage medium having stored therein executable commands for execution on a processor of a client computer. The processor when executing the commands providing remote access to an application performed on a server computer connected thereto. The processor generates a client difference program having encoded data indicative of a change of a state of the application last received from a server computer and transmits it to the server computer. The processor receives from the server computer a server difference program and executes it for updating the state of the application last received from the server computer. The processor then generates display data indicative of the updated state of the application last received from the server computer.

[0013] According to another aspect of the present invention, there is further provided a storage medium having stored therein executable commands for execution on a processor of a server computer. The processor when executing the commands providing remote access to an application performed on the server computer to a client computer connected thereto. The processor receives from the client computer a client difference program having encoded data indicative of a change of a state of the application last received from a server computer and executes the client difference program for determining an updated state of the application. The processor generates a server difference program having encoded a difference between the

updated state of the application and the state of the application last sent to the client computer. The processor then transmits the server difference program to the client computer.

[0014] The advantage of the present invention is that it provides a method and system for providing remote access to the state of an application such that the state of the application is presented to a remote user in a fashion adapted to hardware capabilities of the remote computer.

[0015] A further advantage of the present invention is that it provides a method and system for providing remote access to the state of an application such that the state of the application is presented to a remote user absent replication of the application program's data processing.

[0016] A further advantage of the present invention is that it provides a method and system for providing remote access to the state of an application wherein transmission of data is substantially reduced.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] A preferred embodiment of the present invention is described below with reference to the accompanying drawings, in which:

[0018] Figure 1 is a simplified block diagram of a system for providing remote access to a state of an application according to a preferred embodiment of the invention;

[0019] Figures 2 to 4 are simplified flow diagrams of a method for providing remote access to a state of an application according to a preferred embodiment of the invention

[0020] Figure 5 is a simplified flow diagram of a preferred implementation of the method for providing remote access to a state of an application illustrated in Figure 2; and,

[0021] Figure 6 is a simplified flow diagram of another implementation of the method for providing remote access to a state of an application illustrated in Figure 2.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

[0022] Unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which the invention

belongs. Although any methods and materials similar or equivalent to those described herein can be used in the practice or testing of the present invention, the preferred methods and materials are now described.

[0023] While the description of the preferred embodiment herein below is with reference to patient application files in a clinical environment only for the sake of simplicity, it will become evident to those skilled in the art that the preferred embodiment is not limited thereto, but is applicable in numerous other fields where users remotely access a state of an application program via a computer network.

[0024] Referring to Figure 1, a system 100 for providing remote access to the state of an application according to a preferred embodiment of the invention is shown. The system comprises a plurality of client computers 110A, 110B - for example, workstations, personal computers, laptop computers, and/or wireless handheld devices – connected via a communication network 108 to a server computer 102 – for example, a central computer or a high performance workstation. For example, in a clinical environment numerous tasks executed on the system 100 are related to: viewing medical images of a patient and entering a diagnostic report into a respective patient application file; and for entering other information such as scheduling information of the patient's appointments with medical practitioners. The communication network 108 comprises, for example, a Local Area Network (LAN) connecting the client computers with one or more server computers within an organization such as, for example, a hospital. Alternatively, the communication network 108 comprises a Wide Area Network (WAN) or the Internet connecting client computers of various organizations such as, for example, hospitals, diagnostic clinics, and family physicians offices.

[0025] In the system 100 an application program for performing an application associated therewith is executed on processor 104 of the server computer 102, for example, by executing executable commands stored in memory 106. For example, applications enable: viewing and modifying of medical images, viewing and modifying of a patient file, entering a diagnostic report, and scheduling a patient's appointments. A state of an application is changed, for example, when a user changes information in a patient file or requests a different view of a medical image.

[0026] For example, using the system 100 a radiologist is enabled to perform an image reading for a diagnosis by remotely accessing an imaging application executed on a dedicated PACS workstation and to enter a report into a patient file by remotely accessing a patient file application.

[0027] Typically, an application such as, for example, the patient file application is processed at different client computers 110A, 110B connected via the communication network 108 to the server computer 102. For example, at client computer 110B a medical practitioner enters his diagnosis of a medical image into the patient application file while at client computer 110A a health care worker enters scheduling information for an appointment into the patient application file. The processing of the same patient file application at two different client computers 110A, 110B results in different states of the patient file application.

[0028] The system 100 provides remote access as well as synchronizes the states of an application by executing a method for providing remote access to the state of an application according to embodiments of the invention, as will be described herein below. The method for providing remote access to the state of an application is performed, for example, by executing a remote access program on processors 112A, 112B. For example, the remote access program is performed by executing executable commands stored in memory 114A, 114B. The remote access program is in communication with a remote server access program executed on the processor 104 of the server computer 102, which is in communication with the application program. Alternatively, the remote access program is in direct communication with the application program. When performing the method for providing remote access to the state of an application, the processor of the server computer 102 communicates with respective processors 112A, 112B of the client computers 110A, 110B using standard communication protocols – such as, for example, Hyper Text Transfer Protocol (HTTP) - of the communication network 108, which are well known to those skilled in the art.

[0029] In the method for providing remote access to the state of an application, “view data” of the state of the application are generated and transmitted in a fashion according to hardware capabilities of the client computer – for example, processing capacity, memory size, type of graphical display, and type of user interface. For example, “view data” generated and transmitted

for a personal computer are different from “view data” generated and transmitted for a wireless handheld device. For example, during a remote access a user modifies data of a patient file. First, “view data” indicative of the data of the patient file are generated at the server computer and transmitted to the client computer where the “view data” are displayed. The user using a user interface of the client computer then provides data indicative of a change of data of the patient file as “view data” which are then encoded and transmitted to the server computer. The server computer then changes the state of the application in dependence upon the data indicative of a change provided by the user and the application then stores data in dependence thereupon in the patient file stored in a database. Using “view data” enables presentation of the state of an application without transmitting sensitive and/or voluminous diagnostic data and avoids replicating of the application’s data processing on the client computer.

[0030] Referring to Figure 2, a method for providing remote access to the state of an application according to a preferred embodiment of the invention is shown. At the server computer 102 an application program for performing an application associated therewith is executed – 10 - on processor 104. At client computer 110A connected to the server computer 102 via the communication network 108 a remote access program is executed – 12 – on processor 112A for providing remote access to a state of the performed application. At the client computer 110A a client difference program is generated – 14. The client difference program has encoded data indicative of a change of a state of the application last received from the server computer 102. The data indicative of a change of a state of the application are, for example, received from a user interface 118A, 118B of the client computer 110A, 110B, respectively.

[0031] The difference programs comprise executable commands for execution by a processor. The encoded data are indicative of: a path to the change in the state of the application; a type of the change; and a value of the change. The type of change is, for example, a “modification”, a “deletion”, or an “insertion”, with the value for deletion changes being empty. An example of a difference program is:

Path="Patient/Name/First"

Type="Modification"

Value="Jane"

Path="Patient/Hobby"

Type="Insertion"

Value="Knitting"

Path="Patient/Career"

Type="Deletion"

Value=""

[0032] The client difference program is then transmitted – 16 - to the server computer 102 via the communication network 108. At the server computer 102 the client difference program is executed – 18 - for determining an updated state of the application followed by generating a server difference program. The server difference program has encoded a difference between the updated state of the application and the state of the application last sent to the client computer 110A. This difference also captures changes, for example, received from another client computer – for example, client computer 110B – or generated due to execution of the application program in dependence upon the executed client difference program for determining the updated state of the application. The server difference program is then transmitted – 20 - to the client computer 110A via the communication network 108. At the client computer 110A the server difference program is executed – 22 - for updating the state of the application last received from the server computer 102. Display data indicative of the updated state of the application last received from the server computer 102 are generated – 24 – and displayed in a human comprehensible fashion on display 116A. Optionally, step 24 is omitted, for example, in situations where the updated state of the application does not affect the data displayed on the display 116A.

[0033] Preferably, a remote server access program is executed on the processor 104 of the server computer 102 for executing the client difference program; communicating with the application program for determining the updated state of the application; and generating the server difference program. Provision of the remote server access program enables remote access to off-the-shelf application programs, increases substantially flexibility for the provision of the remote access absent modifications of the application program, and substantially facilitates

implementation of the remote access as a retrofit. Alternatively, the application program is enabled to perform the operations of the remote server access program.

[0034] Further preferably, the method for providing remote access to the state of an application enables limited access to a predetermined portion of the state of the application. For example, the limited access is enabled for:

meeting hardware capabilities – for example, processing capacity, memory size, type of graphical display, and type of user interface - of the client computer – for example, the hardware capabilities of a personal computer are substantially different to the hardware capabilities of a wireless handheld device;

being compatible with different operating systems of different client computers; and,

implementing user access restrictions to information for providing different users with different access to the state of an application – for example, an administrative user for scheduling a patient's appointments is prevented from accessing diagnostic information of the patient.

[0035] Further preferably, the method for providing remote access to the state of an application enables remote access for a plurality of client computers 110A, 110B, which are, for example, simultaneously, connected to the server computer 102 via the communication network 108 for accessing the state of a same application.

[0036] As illustrated in Figure 3, upon receipt a second client difference program received from a second client computer 110B is executed – 26 - at the server computer 102 for determining a second updated state of the application followed by generating a second server difference program. The second server difference program has encoded a difference between the second updated state of the application and the state of the application last sent to the second client computer. The second server difference program is then transmitted – 28 - to the second client computer 110B via the communication network 108. At the second client computer 110B the second server difference program is executed – 30 - for updating the state of the application last received from the server computer 102. Display data indicative of the updated state of the application last received from the server computer 102 are generated – 32 - and displayed in a human comprehensible fashion on display 116B. Optionally, step 32 is omitted, for example, in

situations where the updated state of the application does not affect the data displayed on the display 116B.

[0037] At the server computer 102 a third server difference program is generated – 34. The third server difference program has encoded a difference between the second updated state of the application and the state of the application last sent to the client computer 110A. The third server difference program is then transmitted – 36 - to the client computer 110A via the communication network 108. At the client computer 110A the third server difference program is executed – 38 - for updating the state of the application last received from the server computer 102. Display data indicative of the updated state of the application last received from the server computer 102 are generated – 40 – and displayed in a human comprehensible fashion on display 116A. Optionally, step 40 is omitted, for example, in situations where the updated state of the application does not affect the data displayed on the display 116A. Preferably, the steps 34 to 40 are executed simultaneously to the execution of steps 26 to 32. Alternatively, the steps 34 to 40 are omitted and the changes received from the second client computer 110B are provided to the client computer 110A after receipt of a client difference program from the client computer 110A.

[0038] Upon initiation of the remote access for the client computer a state of the application is provided as illustrated in Figure 4. Upon receipt of a request from the client computer 110A for the state of the application a server application state program is generated – 42 - at the server computer 102. The server application state program has encoded data indicative of the state of the application. The server application state program is then transmitted – 44 - to the client computer 110A via the communication network 108. At the client computer 110A the server application state program is executed – 46 - for determining the server state of the application. Display data indicative of the server state of the application are generated – 48 – and displayed in a human comprehensible fashion on display 116A. Optionally, the steps 42 to 48 are also performed during provision of the remote access to ensure a provision of a same state of the application at the server computer 102 and the client computer 110A. For example, the steps 42 to 48 are performed in predetermined time intervals or after data loss during transmission. Further optionally, step 48 is omitted, for example, in situations where the updated state of the application does not affect the data displayed on the display 116A.

[0039] Referring to Figure 5, a preferred implementation of the method for providing remote access to the state of an application is illustrated - with reference to the method shown in Figure 2. In the memory 106 of the server computer 102 data indicative of a state of the application such as, for example, a patient file application, are stored - 10A - in two states thereof: server data indicative of the server state of the application - the state of the application at the server (and possibly changed by the server); and, server client data indicative of the state of the application last sent to the client computer 110A. In the memory 114A of the client computer 110A data indicative of a state of the application are stored - 12A - in two states thereof: client server data indicative of the state of the application last received from the server computer 102; and, client data indicative of the state of the application after client interaction. In step 14 a difference between the state of the application after client interaction and the state of the application last received from the server computer is encoded - 14A - with the difference being the difference between the client data and the client server data. In step 18: second server client data are determined - 18A - by executing the client difference program on the server client data; updated server data are determined - 18B - by executing the client difference program on the server data; a server difference program is generated - 18C - that encodes a difference between the updated server data and the second server client data; third server client data are determined - 18D - by executing the server difference program on the second server client data; and, the server client data are replaced - 18E - with the third server client data. Optionally, a locking process is provided for locking the server data in order to prevent the same from being modified by other processes prior step 18B, which is released, for example, after step 18C. In step 22: second client server data are determined - 22A - by executing the server difference program on the client server data; second client data are determined - 22B - by executing the server difference program on the client data; the client server data are replaced - 22C - with the second client server data; and, the client data are replaced - 22D - with the second client data. Optionally, a locking process is provided for locking the client server data and the client data during execution of the steps 22A to 22D.

[0040] Referring to Figure 6, an alternative implementation of the method for providing remote access to the state of an application is illustrated - with reference to the method shown in Figure 2. In the memory 106 of the server computer 102 data indicative of a state of the application are stored - 10A - in two states thereof: server data indicative of the server state of

the application – the state of the application at the server computer (and possibly changed by the server computer); and, server client data indicative of the state of the application last sent to the client computer 110B as a server state application program. In step 18: updated server data are determined – 18F - by executing the client difference program on the server data; and, a server difference program is generated – 18G - that encodes a difference between the updated server data and the server client data. In step 22 display data indicative of the updated state of the application last received from the server computer are directly generated – 22E – by executing the received server difference program and displayed – 23 – in a human comprehensible fashion on display 116B. This implementation obviates storage of data related to the state of the application in memory of the client computer 110B which is beneficial for client computers having limited storage capacity such as hand held devices. Preferably, a server application state program that has encoded data indicative of the server state of the application is generated and transmitted, as disclosed herein above, to the client computer, for example, in predetermined time intervals.

[0041] As is evident to those skilled in the art, the embodiments of the system 100 and method for providing remote access to the state of an application according to the invention have been described with respect to a client-server network architecture comprising two client computers for the sake of simplicity but are not limited thereto and are applicable for other network architectures as well as various numbers of client computers.

[0042] The embodiments of the method for providing remote access to the state of an application are performed, for example, by executing executable commands stored in storage mediums – for example, the memory 106 using processor 104 of the server computer 102 and the memory 114A using the processor 112A of the client computer 110A. The executable commands for being stored in the memory of the client computers are, for example, transmitted from the server computer 102 via the communication network 108.

[0043] The present invention has been described herein with regard to preferred embodiments. However, it will be obvious to persons skilled in the art that a number of variations and modifications can be made without departing from the scope of the invention as described herein.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A method comprising:

- a) at a server computer, executing an application program for performing an application associated therewith;
- b) at a client computer connected to the server computer via a communication network, executing a remote access program for providing remote access to a state of the performed application;
- c) at the client computer, generating a client difference program having encoded data indicative of a change of a state of the application last received from the server computer;
- d) transmitting the client difference program to the server computer;
- e) at the server computer, executing the client difference program for determining an updated state of the application and generating a server difference program having encoded a difference between the updated state of the application and the state of the application last sent to the client computer;
- f) transmitting the server difference program to the client computer; and,
- g) at the client computer, executing the server difference program for updating the state of the application last received from the server computer.

2. A method as defined in claim 1 comprising:

at the client computer, storing in memory thereof client server data indicative of the state of the application last received from the server computer and client data indicative of the state of the application after client interaction.

3. A method as defined in claim 2 wherein c) comprises encoding a difference between the state of the application after client interaction and the state of the application last received from the server.

4. A method as defined in claim 3 comprising:

at the server computer, storing in memory thereof server data indicative of a server state of the application and server client data indicative of the state of the application last sent to the client.

5. A method as defined in claim 4 wherein e) comprises:

determining second server client data by executing the client difference program on the server client data;

determining updated server data by executing the client difference program on the server data;

generating a server difference program that encodes a difference between the updated server data and the second server client data;

determining third server client data by executing the server difference program on the second server client data; and,

replacing the server client data with the third server client data.

6. A method as defined in claim 5 wherein g) comprises:

determining second client server data by executing the server difference program on the client server data;

determining second client data by executing the server difference program on the client data;

replacing the client server data with the second client server data; and,

replacing the client data with the second client data.

7. A method as defined in any one of claims 1 to 6 comprising:

at the server computer, storing in memory thereof server data indicative of a server state of the application and server client data indicative of the state of the application last sent to the client as a server application state program; and,

wherein e) comprises:

determining updated server data by executing the client difference program on the server data; and,

generating a server difference program that encodes a difference between the updated server data and the server client data; and,

wherein g) comprises:

generating display data indicative of the updated state of the application last received from the server computer; and,

displaying the display data in a human comprehensible fashion.

8. A method as defined in claim 7 comprising:

at the server computer, generating a server application state program having encoded data indicative of the server state of the application;

transmitting the server application state program to the client computer;

at the client computer, executing the server application state program for determining the server state of the application;

generating display data indicative of the updated state of the server state of the application; and, displaying the display data in a human comprehensible fashion.

9. A method as defined any one of claims 1 to 6 comprising:

at the client computer, generating display data indicative of the updated state of the application last received from the server computer; and,

displaying the display data in a human comprehensible fashion.

10. A method as defined any one of claims 1 to 9 comprising:

at the server computer, executing a second client difference program received from a second client computer for determining a second updated state of the application and generating a second server difference program having encoded a difference between the second updated state of the application and the state of the application last sent to the second client computer;

transmitting the second server difference program to the second client computer; and,

at the second client computer, executing the second server difference program for updating the state of the application last received from the server computer.

11. A method as defined in claim 10 comprising:

at the server computer, generating a third server difference program having encoded a difference between the second updated state of the application and the state of the application last sent to the client computer;

transmitting the third server difference program to the client computer; and,

at the client computer, executing the third server difference program for updating the state of the application last received from the server computer.

12. A method as defined any one of claims 1 to 11 comprising:
at the client computer, receiving from a user interface connected thereto the data indicative of the change of the state of the application.
13. A method as defined any one of claims 1 to 12 wherein the encoded data are indicative of: a path to the change in the state of the application; a type of the change; and a value of the change.
14. A method as defined any one of claims 1 to 13 wherein the difference programs comprise executable commands for execution by a processor.
15. A method as defined any one of claims 1 to 14 comprising executing the application program in dependence upon the executed client difference program for determining the updated state of the application.
16. A method as defined any one of claims 1 to 15 comprising executing a remote server access program, the remote server access program for: executing the client difference program; communicating with the application program for determining the updated state of the application; and generating the server difference program.
17. A method as defined any one of claims 1 to 16 wherein the remote access program enables access limited to a predetermined portion of the state of the application.
18. A method as defined any one of claims 1 to 17 comprising:
at the server computer, generating a server application state program having encoded data indicative of the state of the application;
transmitting the server application state program to the client computer; and,
at the client computer, executing the server application state program for determining the server state of the application.

19. A storage medium having stored therein executable commands for execution on a processor of a client computer, the processor when executing the commands providing remote access to an application performed on a server computer connected thereto, the processor performing:
generating a client difference program having encoded data indicative of a change of a state of the application last received from a server computer;
transmitting the client difference program to the server computer;
receiving from the server computer a server difference program; and,
executing the server difference program for updating the state of the application last received from the server computer.

20. A storage medium having stored therein executable commands for execution on a processor of a server computer, the processor when executing the commands providing remote access to an application performed on the server computer to a client computer connected thereto, the processor performing:
receiving from the client computer a client difference program having encoded data indicative of a change of a state of the application last received from a server computer;
executing the client difference program for determining an updated state of the application and
generating a server difference program having encoded a difference between the updated state of the application and the state of the application last sent to the client computer; and,
transmitting the server difference program to the client computer.