A METHOD AND APPARATUS FOR MAKING A HYPERMEDIUM INTERACTIVE

Field of the Invention

The invention relates generally to the field of computer communications and more specifically to the field of remote application processing.

Background of the Invention

A hypermedium is a graphical display which contains a series of graphic and textual images which are referred to as hyperlinks. Each hyperlink typically corresponds to additional information which is available to a user of the hypermedium. For example a hypermedium display might be an encyclopedic article about flight. If the graphical display of the article includes a picture of a rocket, and if the user is able to obtain information about the engine of the rocket by selecting (typically by using a pointing device referred to as a computer mouse) a portion of the display showing the engine, the portion of the picture containing the rocket engine is referred to as a hyperlink. That is, the portion of the picture containing the engine is a link to information about the engine. The activation of a hyperlink causes the hypermedium to request a data file of the desired information from the program actually controlling the hypermedium.

When a hypermedium is used in conjunction with a network, for example the world wide web or the internet, a user on a client node on the network first accesses what is termed a home page or a web page. This home page or web page is obtained from another node on the net, termed a network server, and is displayed on the client node by a program termed a network browser or web browser. In such a case, when the user selects the hyperlink (in the previous example, the portion of the display showing the rocket engine) a request is made to the network browser for the data file containing the requested information. The network browser on the user's node establishes communication with the network node, the data server, having data corresponding to the requested hyperlink. In this case the node having information about the rocket engine contains the information required by the hyperlink.

Although hypermedium has been used to transfer information to a user in the manner just discussed, it is desirable to be able to use a hypermedium display to interactively execute applications such as database programs located on another computer, an application execution

server, on the network. The present invention relates to a method and apparatus to accomplish this task.

Summary of the Invention

The invention relates to a system of making a hypermedium page interactive to thereby permit a application to be executed on one node and the results displayed and data entered on another node. In one embodiment the system includes a client node, a network server node and an application execution server node interconnected by a communication link. A hyperlink on the hypermedium page is displayed on the client node and a hyperlink configuration file (corresponding to the hyperlink on the client node) is located on the network server node. In one embodiment, a client agent is located on the client node and a server agent is located on the application execution server node. A communication link is created by the client agent between the client agent on the client node and the server agent on the application execution server node in response to data in the hyperlink configuration file. The system also includes an application on the application execution server node which executed on the application execution server node in response to the communications link between the client agent and the server agent. The application running on the application execution node then communicates with the client agent through the server agent. The client agent on the client node is responsible for receiving data input from the user and transferring the data to the application on the application execution node and receiving data from the application on the application execution node and displaying data output to the user on the client node.

The invention also relates to a method of making a hypermedium page interactive. The method includes the steps of selecting a hyperlink on the hypermedium page displayed on a client node; retrieving (from a server node to the client node) a hyperlink configuration file corresponding to the hyperlink and starting a client agent on the client node. The method further includes the steps of creating a communications link between a server agent on an application execution server and the client agent; starting the application on the application execution server in response to the connection; communicating data between the client agent on the client node and application on the application execution node; and managing the display and input of data on the client node.

Brief Description of the Drawings

Other features and benefits of the invention can be more clearly understood with reference to the specification and the accompanying drawings in which:

Fig. 1 is a block diagram of an embodiment of the system of the invention; and

Fig. 2 is a flow chart of the operation of system of the invention shown in Fig. 1.

Detailed Description of the Invention

Referring to Fig. 1, and in brief overview, an embodiment of an interactive hypermedium system of the invention includes a client node 10, a network server node 18 and an application execution server node 24 interconnected by a communications link 32, herein referred to without any loss of generality as a network or web. Although only one client node 10, network server node 18 and application execution server node 24, are shown in Fig. 1 for clarity, an actual network may include many such nodes. Alternatively, the services provided by each of the nodes listed may be combined in one or more nodes. For example, the application execution server and the network server may in fact be the same node. In the extreme it is possible for all functions to be performed by the same node, although such would not typically be the case. In addition, although only one application 36 is shown on the application execution server 24, in reality, an application execution server node 24 typically includes many applications 36. Each node on the network or web 32 includes a processor, which may vary significantly from other processors on the web 32 in terms of computing power and associated hardware. Further, the applications 36 available for execution on each node may be different.

A user on a client node wishing to run the application program 36 which is located on the application execution server 24 on the web 32 does so through a graphical user interface 40, which is herein referred to without any loss of generality as a hypermedium, located on the client node 10. The graphical interface is displayed on a graphical display device 42. Data is entered by the users through a mouse 46 and a keyboard 50 located on the client node 10. The graphical display or page 44 which the user first views on the hypermedium 40 is referred to herein without any loss of generality as the home page or web page of the application 36. A page 44 or home page of the hypermedium 40 includes a graphic link 48 or textual link 56 herein referred to without any loss of generality as a hyperlink. The web page is displayed by a process 64 referred to herein without any loss of generality as a network browser 64 executing on the client node 10.

The network browser 64 obtains the first page or web page 44 from a network server node 18 and displays the web page 44 on the hypermedium 40 for the user to view on the graphical display device 42. When the user selects an application program 36 to execute (by selecting a graphical 48 or textual 56 hyperlink using the mouse 46 or keyboard 50) the network browser 64 obtains a network configuration file 68 corresponding to the selected application 36

from a predetermined network server 18 and starts a client agent 72 which will communicate with the selected application 36. This will be discussed in more detail below.

The client agent 72 reads the configuration file 68 and establishes a communications link to a server agent 80 on the application execution server 24 specified by the configuration file 68. In one embodiment, the configuration file 68 includes the name of the application and the node location of the application 36 corresponding to the hyperlink 48, 56. The The configuration file may also contain optional information such as authentication or authorized user information. Server agent 80 performs the operations necessary (such as authentication) to permit the client agent 72 access to the application 36, and once access is permitted, starts the application 36 requested by the user. Once the application 36 is executing on the application execution server, the application 36 communicates through the server agent 24 directly with the client agent 72 without intervention by the network browser 64. The client agent 72 is then responsible for receiving data from the user through the mouse 46 and keyboard 50 and transmitting it to the application program 36 on the application execution server 24. Similarly, the client agent 72 is responsible for receiving data from the application 36 on the application execution server 24 and displaying the data in an application display window 74 on the graphical display device 42 on the client node 10. It should be noted that the application display window 74 may be located within the boundaries or outside the boundaries of the hypermedium 40. When the application 36 is completed the server agent 80 instructs the client agent 72 to disconnect the communication link 32 between the client agent 72 and the server agent 80 and the server agent waits for the next connection.

Fig. 2 depicts the operation of the system in more detail. Initially, the client agent 72 is registered (Step 1) with the network browser 64 of the client node 10 and an entry is made in the network browser's registration file 88 (Fig. 1). This entry permits the network browser 64 to start the client agent 72 whenever a given file type is requested by the hyperlink 48, 56 of the hypermedium 40. In this case the client agent 72 is designed to permit a user on the client node 10 to execute and interact with a remote application 36 on an application execution server node 24. The client agent 72 would be registered with the network browser 64 such that whenever a hyperlink 48, 56 requested the given file type (for example .RMT for remote execution) from the network browser 64, the network browser 64 would start the client agent 72 which would permit remote execution and interaction with an application 36 resident on an application execution server 24. The invoking of the client agent 72 is discussed in more detail below.

This Sheet Has Been Replaced



THIS SHEET HAS BEEN REPLACED

Next, when a user wishes to execute an application from a hypermedium environment, for example a database program, the hypermedium 40 is displayed in a manner that is well known to those skilled in the art. When the user selects a hyperlink 48, 56 on the page 44 of the hypermedium (Step 2) by using the mouse 46 or keyboard 50 on the client node 10, a request is made to the network browser 64 for the corresponding data file (Step 3). In this example, the filetype (.RMT) is requested.

The network browser 64 obtains the corresponding configuration file 68 from the network server 18 which is specified in the file request made by the hyperlink 48, 56 to the network browser 64 (Step 4). The network browser 64 then compares the obtained configuration file 68 with the registration file 88 of client agent names which it maintains (Step 5). If the client agent 72 specified by the configuration file 68 is found in the registration file 88, the client agent 72 is started (Step 6).

The invoked client agent 72 reads the configuration file 68 (Step 7), and based upon the information in the configuration file 68, begins to establish a communication link with the server agent 80 on the application execution server 24 (Step 8), in this case the sales database application execution server (generally 24).

Considering the process of beginning the communications link of step 8 (Fig. 2) in more detail, communication begins with the server agent 80 monitoring communication activity on the network 32. At this point, no protocol assumptions are made by the server agent 80 beyond those necessary for the transport layer. Similarly, the client agent 72 also makes no assumption of the communications protocol beyond that required by the transport layer. Once the server agent 80 determines that a client agent 72 is attempting to communicate with it, the server agent 80 transmits a message to the client agent 72 indicating that service is available.

Once the client agent 72 determines that service is available on the application execution server node 24, the client agent 72 transmits a message to the server agent 80 indicating that it is ready to proceed with the communication protocol. Once the server agent 80 has responded that it to is ready to continue the communication protocol, the client agent 72 enables the protocol necessary for it to run the application 36. In response to the message from the client agent 72, the server agent 80 also enables the required protocol. The server agent 80 then transmits a message using the required protocol indicating that the client agent's request has been received and accepted.

In response the client agent 72 and the server agent 80 exchange a set of messages which negotiate the parameters under which communications will occur. Once negotiations are

complete, the client agent 72 and the server agent 80 are able to communicate as necessary for the application 36 to be run by the user. Once the communications protocol has been established and the server agent 80 has authenticated the client agent 72 (Step 9) (for example determining that the user has permission to read and write to the database) the application 36 (Step 10) is run on the application execution server 24. At this point application 36 running on the application execution server 24 is communicating via the server agent 80 with the client agent 72 on the client node 10. The client agent 72 is now responsible for transmitting data input by the user using the mouse 46 and keyboard 50 to the application 36 running on the application execution server 24. Further, the client agent 72 is responsible for receiving data for display from the application 36 and displaying that data in the application window 74 on the graphical display device 42 of the client node 10.

It should be noted that the underlying presentation protocol which passes data to a transport layer such as TCP/IP must be capable of transferring graphical information. Examples of such protocols which may be used for interactive hypermedia communication include public domain X-WINDOWS protocol and the proprietary ICA protocol of Citrix Systems Inc.

Thus the above described system permits a user on a client node 10, which may have very limited resources, to start and interact with an application program 36 located on another application execution server node 24. The application 36 then runs on the application execution server node 24 and the data is input and the results displayed on the client node 10.

These and other examples of the concept of the invention illustrated above are intended by way of example and the actual scope of the invention is to be determined solely from the following claims.

CLAIMS

What is claimed is:

1. A method of making a hypermedium page interactive comprising the steps of

selecting a hyperlink on said hypermedium page displayed on a client node;

retrieving a hyperlink configuration file, corresponding to said hyperlink, from a network server to said client node;

starting a client agent on said client node;

creating by said client agent a communication link to an application execution server in response to data in said hyperlink configuration file; and

starting said application on said application execution server in response to said communication link.

2. The method of claim 1 further comprising the step of displaying said application on said client node over said communication link.

3. The method of claim 1 further comprising the step of obtaining a hypermedium page from a network server prior to the step of selecting a hyperlink on said hypermedium page.

4. The method of claim 1 further comprising the step of starting a server agent on said application execution server prior to creating said communication link.

5. The method of claim 1 wherein said communications link used to a graphical presentation protocol.

6. A system of making a hypermedium page interactive comprising:

a client node;

a hyperlink on said hypermedium page displayed on said client node;

a network server node, in communication with said client node;

a hyperlink configuration file on said server node, said configuration file corresponding to said hyperlink on said client node;

a client agent on said client node;

an application execution server node in communication with said client node and said network server node;

This Sheet Has Been Replaced

THIS SHEET HAS BEEN REPLACED

a server agent on said application execution server node; and

an application on said application execution server node,

said client agent establishing a communications link with said server agent, in response to said configuration file, and

said application running in response to said communications link between said client agent and said server agent.

7. The system of claim 6 further comprising a display process displaying said application on said client node over said communications link.

8. The system of claim 6 further comprising a process obtaining said hypermedium page from said network server and providing said hypermedium page to said client node.

9. The system of claim 6 wherein said hyperlink configuration file comprises:

an application name corresponding to said hyperlink; and

an application server address corresponding to said hyperlink.