(54) Title: METHOD FOR USING SOFTWARE AGENTS IN A NETWORK OF COMPUTERS

(57) Abstract: In a method for using software agents in a network of computers, a plurality of user computers can be connected to the network and each user computer can send at least one software agent on the network. A number of computers of the network is adapted to receive and transmit software agents. At least one waiting location for software agents is provided in at least one waiting computer of the network, each waiting location being identified by a given name and the network address of the corresponding waiting computer. A user identification and a waiting location identification are added to the software agent in each user computer and the waiting location identifications used are stored in the corresponding user computer. Each software agent arriving in a waiting computer with one or more waiting locations is checked on a corresponding waiting location identification, wherein the software agent is held in the waiting computer if a match is found between one of the waiting locations of the computer and the waiting location identification of the software agent. A user computer collects its software agent(s) from the waiting computer(s) corresponding to the used waiting location identification(s) stored by making a network connection to said waiting computer(s) and transmitting its user identification to the waiting computer.
For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.
Method for using software agents in a network of computers

The invention relates to a method for using software agents in a network of computers, wherein a plurality of user computers can be connected to the network and each user computer can send at least one software agent on the network, wherein a number of computers of the network is adapted to receive and transmit software agents.

Software agents can be used for a number of applications in a network of computers, in particular the Internet. For example, software agents can be used to search information on the Internet, to distribute specific information to other software agents, to obtain commercial products from virtual shops, and the like. In known methods of the above-mentioned type, it is a problem for software agents to return to their user computers. Many user computers are not continuously connected to the Internet. Further, user computers generally do not have the same network address when they are connected to the network to collect their software agents as the network address of the user computer at the time of sending the software agents on the network. It is therefore not possible to use the network address of the user computer to return to the user computer and to identify the user computer in a secure manner.

Moreover, computer networks of companies or intranets are generally secured by means of a so-called firewall which is adapted to prevent any connection from outside the firewall to the company's network. Although such networks are generally continuously connected to the Internet, software agents can not return automatically as they can not pass the firewall.

The invention aims to provide a method of the above-mentioned type, wherein software agents can safely return to their user computers.

According to the method of the invention at least one waiting location for software agents is provided in at least one waiting computer of the network, each waiting location be-
ing identified by a given name and the network address of the corresponding waiting computer, wherein a user identification and a waiting location identification are added to the software agent in each user computer and the waiting location identifications used are stored in the corresponding user computer, wherein each software agent arriving in a waiting computer with one or more waiting locations is checked on a corresponding waiting location identification, wherein the software agent is held in the waiting computer if a match is found between one of the waiting locations of the computer and the waiting location identification of the software agent, wherein a user computer collects its software agent(s) from the waiting computer(s) corresponding to the used waiting location identification(s) stored by making a network connection to said waiting computer(s) and transmitting its user identification to the waiting computer.

In this manner each software agent can always return to a fixed location within the computer network which is both known to the software agents and its user computer. The user computer can safely collect its software agent or agents from the waiting location using its user identification.

The invention will be further explained by reference to the drawing in which a schematic diagram representing the Internet is shown.

It is noted that the Internet is only mentioned as an example of a world-wide network in which the method of the invention can be used. The method can also be used in other types of networks, such as an intranet. As another alternative the method of the invention can be used in distributed object-oriented database applications.

In the drawing, the Internet is shown, comprising a plurality of user computers 1, connected to the Internet through computers or servers 2 of so-called Internet Service Providers (ISP's). In many cases, the user computers 1 are using a dynamic network address, i.e. a network address which generally changes each time a user computer makes a connection to the computer 2 of the ISP.

The Internet further comprises interconnected comput-
ers 3 and 4, the computers 4 including a software program providing one or more waiting locations for software agents as will be described hereinafter.

A user computer can send a software agent on the Internet, for example with instructions to search the Internet for specific information. Such software agents are software modules, which preferably are of an object-oriented type comprising one or more objects and one or more classes. The software agents as such are not part of the present invention and for a further description reference is made to a co-pending application of the same applicant. A number of the computers 3 and 4 are adapted to receive and transmit software agents preferably in a manner as described in the above-mentioned co-pending application. Each waiting computer 4 is provided with one or more waiting locations or freezing rooms in which software agents moving on the Internet are stored until they are collected by their user computers.

A waiting location is identified by a given name and the network address of the corresponding waiting computer 4. Of course if port numbers are used for sub-addressing, the network address of a waiting location may include a port number within this network address.

When a user computer sends a software agent to the Internet, a waiting location identification, i.e. the given name and the network address, are added to the software agent together with a user identification. This user identification is preferably a public key which is one of two encryption keys of a so-called public key encryption algorithm. The other key of the two encryption keys is stored as secret key in the user computer. It is possible to use the same public key/secret key pair for each software agent or as an alternative, the user computer may generate new public/private keys for each new software agent or group of new software agents. This alternative shows the advantage that it is more difficult for computers 3, 4 visited by the software agents to associate agents with a specific user computer through the public key used.

When a software agent has carried out its task, the software agent will generally return to the computer 4 having
the waiting location corresponding to the waiting location identification added to the agent. When the software agent is transferred to the corresponding waiting computer 4, the waiting computer 4 is adapted to check the waiting location identification of the software agent and if there is a match between this identification and one of its waiting locations, the software agent is held in the waiting computer 4. The software agent is then stored in a database of the waiting computer 4 and the public key used as user identification is stored in the index of the database.

In this manner each software agent sent on the Internet can always return to a predetermined waiting location known both to the software agent and the user computer. When a user wishes to collect one or more of the agents sent on the Internet, the user computer makes a network connection to the network address of the waiting location of its software agents. All waiting location network addresses used by a user computer are stored in the same. When a network connection is made to the waiting computer 4, a user computer 1 transmits a collect request to the waiting computer and its public key and the waiting location name or names. Further a first random number of bits is transmitted to the waiting computer by the user computer. The collect request is received by the waiting computer. After receipt of the collect request, the waiting computer first compares the name of the waiting location(s) with the waiting location(s) present in this waiting computer. If the name received does not correspond with one of its own waiting location, the waiting computer can send an error message to the user computer and will terminate the protocol.

After checking the waiting location name, the waiting computer 4 transmits a second random number of bits to the user computer. The user computer generates a digital signature by means of the private key on the combination of the first and second random numbers of bits. This digital signature is transmitted to the waiting computer 4. The waiting computer 4 checks the signature received by means of the public key received from the user computer. If the signature is valid, the waiting computer 4 compares the public key with the index of public keys
of the database of the corresponding waiting location. If there is no corresponding public key in the index available, a corresponding message "no software agents present" is returned to the user computer and the protocol for collecting a software agent is terminated. If there is a match between the public key received and one or more public keys in the index of the database, all software agents having the same public key are retrieved from the database and returned to the user computer.

If however the digital signature received is not valid, the protocol is terminated and the waiting computer may transmit an error message to the user computer. Optionally the waiting computer generates a security warning and stores the same preferably with as much information on the user computer causing the warning as possible. Such security warnings may be used for providing further protection against unauthorised attempts to retrieve software agents from the waiting location(s).

The advantage of generating the first random number of bits by a user computer and the combination of this first number of bits with the second number of bits provided by the waiting computer is that both the user and waiting computers do know the contents of the message used to generate the digital signature. This means that the user or waiting computer involved in the protocol to collect the software agents can not force the other computer to accept a predetermined message. Using a predetermined message could lead to abuse of the protocol. However, it is possible to use the method described in an embodiment, wherein only the waiting computer generates a random number of bits as a basis for the encrypted signature.

In a slightly amended form, the method described can also be used to retrieve objects from a distributed object-oriented database in a computer system or network. In this method objects are stored in the database of a first computer by means of at least one user computer. A public key is added to each object stored in the database, said public key being one of two encryption keys of a public key algorithm. The other key of the two encryption keys is stored as secret key in a user computer. An object is retrieved from the database by
sends a collect request to the first computer and a user computer adds a first random number of bits to the collect request, wherein, after receipt of a collect request, the first computer transmits a second random number of bits to the user computer. The user computer combines the first random number of bits with the second random number of bits, generates an encrypted signature with its secret key using the combination of bits, and transmits the signature to the first computer. The first computer checks the signature received with the public key received and transfers the objects having the received public key only if the check on the signature indicated a valid signature.

The invention is not restricted to the above-described embodiments which can be varied in anumber of ways within the scope of the following claims.
CLAIMS

1. Method for using software agents in a network of computers, wherein a plurality of user computers can be connected to the network and each user computer can send at least one software agent on the network, wherein a number of computers of the network is adapted to receive and transmit software agents, characterized in that at least one waiting location for software agents is provided in at least one waiting computer of the network, each waiting location being identified by a given name and the network address of the corresponding waiting computer, wherein a user identification and a waiting location identification are added to the software agent in each user computer and the waiting location identifications used are stored in the corresponding user computer, wherein each software agent arriving in a waiting computer with one or more waiting locations is checked on a corresponding waiting location identification, wherein the software agent is held in the waiting computer if a match is found between one of the waiting locations of the computer and the waiting location identification of the software agent, wherein a user computer collects its software agent(s) from the waiting computer(s) corresponding to the used waiting location identification(s) stored by making a network connection to said waiting computer(s) and transmitting its user identification to the waiting computer.

2. Method according to claim 1, wherein a waiting location comprises a database, and wherein a software agent is held at a waiting computer by storing the software agent in a database record and storing the user identification in an index of the database.

3. Method according to claim 2, wherein for collecting a software agent from a waiting computer, the user computer transmits a collect request with its user identification and waiting location name(s) used to the waiting computer and the waiting computer transfers to the user computer all software agents having a corresponding user identification stored in the corresponding database or transmits a message to the user com-
puter indicating that no software agents are present at the waiting location.

4. Method according to claim 3, wherein each user identification is a public key, which is one of two encryption keys of a public key algorithm, wherein the other key of the two encryption keys is stored as secret key in the user computer, wherein, after receipt of a collect request, a waiting computer transmits a random number of bits to the user computer, wherein the user computer generates an encrypted signature with its secret key using the random number of bits received, and transmits the signature to the waiting computer, wherein the waiting computer checks the signature received with the public key received and transfers the software agents having the received public key only if the check on the signature indicated a valid signature.

5. Method according to claim 4, wherein a user computer adds a first random number of bits to the collect request, wherein the user computer combines the first random number of bits with the random number of bits received from the waiting computer as second random number of bits, and uses the combination of bits for generating the encrypted signature.

6. Method according to claim 4, wherein a waiting computer stores a security message if a check of a signature indicated an invalid signature.

7. Method for retrieving objects from a distributed object database in a computer system or network, wherein objects are stored in the database of a first computer by means of at least one user computer, wherein a public key is added to each object stored in the database, said public key being one of two encryption keys of a public key algorithm, wherein the other key of the two encryption keys is stored as secret key in a user computer, wherein an object is retrieved from the database by sending a collect request to the first computer, wherein, after receipt of a collect request, the first computer transmits a random number of bits to the user computer, wherein the user computer generates an encrypted signature with its secret key using the random number of bits received, and transmits the signature to the first computer, wherein the first
computer checks the signature received with the public key received and transfers the objects having the received public key only if the check on the signature indicated a valid signature.

8. Method according to claim 7, wherein a user computer adds a first random number of bits to the collect request, wherein the user computer combines the first random number of bits with the random number of bits received from the first computer as second random number of bits, and uses the combination of bits for generating the encrypted signature.