(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
13 March 2008 (13.03.2008)

PCT

(10) International Publication Number
WO 2008/030494 A2

(51) International Patent Classification:
H04L 12/28 (2006.01)

(21) International Application Number:
PCT/US2007/019402

(22) International Filing Date:
5 September 2007 (05.09.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/842,551 5 September 2006 (05.09.2006) US

(71) Applicant and
(72) Inventor: **DONNELLI, Robert, M.** [US/US]; 7373 E. Doubletree Ranch Road, Scottsdale, AZ 85258 (US).

(74) Agent: **MULLINS, John Jason, Gentry**; P.O. Box 641391, San Jose, CA 95164 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

(54) **Title:** SYSTEM, METHOD AND SOFTWARE FOR CREATING, MAINTAINING, NAVIGATING OR MANIPULATING CLIENT TO SERVER OR PEER TO PEER RELATIONSHIPS AND COMMUNICATIONS CONNECTIONS WITHIN A PRIVATE NETWORK OR PRIVATE VIRTUAL NETWORK

(57) **Abstract:** The present invention provides computer systems, private computer networks or private virtual networks for managed peer to peer communications and to methods and software for accessing, utilizing and managing private computer networks or virtual private networks including servers and clients for the transfer of digital media More particularly, the present invention provides a system, methods and software for creating, maintaining, accessing, navigating and persisting relationship between servers with other servers and/or clients within a private or virtual private network fo managed peer to peer communications with respect to digital file transfers. In one aspect, the invention also provides an application programming interface ("API") an software useful for creating, maintaining, accessing, navigating and persisting such relationships as a complex relationship graph, or as a set of complex relationship graphs to permit management of peer to peer communications and to manage or monitor digital media file transfers during a peer to peer communication.

SYSTEM, METHOD AND SOFTWARE FOR CREATING, MAINTAINING, NAVIGATING OR MANIPULATING CLIENT TO SERVER OR PEER TO PEER RELATIONSHIPS AND COMMUNICATION CONNECTIONS WITHIN A PRIVATE NETWORK OR PRIVATE VIRTUAL NETWORK

FIELD OF THE INVENTION

The field of the present invention relates generally to computer systems, private computer networks or private virtual networks for managed peer to peer communications and to methods and software for accessing, utilizing and managing private computer networks or virtual private networks including servers and clients for the transfer of digital media. More particularly, the present invention relates to a system, methods and software for creating, maintaining, accessing, navigating and persisting relationship between servers with other servers and/or clients within a private or virtual private network for managed peer to peer communications with respect to digital file transfers. In one aspect, the invention also relates to an application programming interface ("API") and software useful for creating, maintaining, accessing, navigating and persisting such relationships as a complex relationship graph, or as a set of complex relationship graphs to permit management of peer to peer communications and to manage or monitor digital media file transfers during a peer to peer communication.

BACKGROUND OF THE INVENTION

Systems for creating or accessing networks for server to peer and peer to peer communications have been used in the past few years, but operate from a fundamental management flaw of providing to a client who has logged into a network a listing with one or more links to peers that might have a digital file that the client wants or might be available for a peer to peer telephone contact via a digital satellite or internet connection. This is a frequent approach to accomplish access to such private networks in an unmanaged, or a limited management of the virtual private networks other than granting permission to access the network and giving the user the addresses to peers or servers on the network.

In prior networks, the actual management of attachments is controlled by the user, which has led to abuse of the system and impermissible activities via unmanaged peer to peer communications where the identity of the peers is not adequately known to the network administrator, and the activity that occurs after peer to peer connections is not managed in real time. U.S. Patent Number 5,884,031, Issued to Ice on March 16, 1999 is an example of such a virtually unmanaged network of the type frequently used by peer to peer individuals that is unmanagemable. Requiring peers who offer connections to other peers to have records monitoring software is not completely satisfactory, since the monitoring does not occur in real time. And, unfortunately these unmanaged or limited managements of peer to peer communications fall short of providing the kind of connections that allow adequate monitoring or management of digital file transfers or telephone converstations via peer to peer communications. Such connections are also not adequately managed for true internet or satellite network telephone conversations management and billing in real time.

In such prior systems the client chooses and manages the connections rather than the network server and rather than the peer to whom the client wishes to connect as part of a network activity. Such connections are actually “off network” once the peer to peer communication begins, and creates file sharing management problems. While links may be set up as hidden from the end user when downloading files, other types of connections may not be, such as peer to peer phone calls. This makes the network merely a facilitator of peer to peer connections or network server connections, rather than being a manager who can control, establish or disconnect the connections of peer to peer communicatons when network rules are violated.

Accordingly, there is a need for a managed and secure network or virtual network of unlimited size with respect to servers and client for better facilitated and managed peer to peer communications where the identity of each client is known to the network, but not necessarily known by the individual peers who communicate with each other. Such a system that will shield peer links and identities from later independent “off-network” or “outside of network” communications by network clients is needed. There is a need for such a network which can provide unlimited communications within the private network or private virtual network within network rules in a way that will meet regulatory and copyright standards with respect to identification of media file transfers and reporting. Also needed is a better managed, real time, internet telephone peer to peer system.

MANAGING NETWORKS AS AN OBJECT GRAPH

A geographic problem of physical locations complicates the creation, access, changing or deleting of a set of network relationships between clients and servers, and between two clients (peer to peer where one or more may also be acting as a server for a particular transaction). The network or virtual network may be distributed over multiple physical computer machine locations or even distributed over multiple Internet website locations including connections between portable devices such as mobile phones, PDA and other wireless devices that may be independent of the core private network or from a central machine. Thus, in the context of this invention, the relationships between such entities may be managed as an object graph of complex object relationships using object based programming languages and software.

Importantly, relationships may exist between a two servers, between a client and a server, or between two clients that may be represented using an object model and object programming software according to the present invention. A relationship between one such entity and another may be member selected from the group of three relationship types consisting of 1 to 1 (1-1), 1 to many (1-M) or many to many (M-M). Complex combinations of these relationships may be thought of in the context of this invention of managing network relationships as an object graphs as existing in data object relationships definitions for a given data object for management. These relationships are described or illustrated in further detail later in this document.

As is clear from the above, objects may logically span multiple relational tables or multiple object databases, and may even be distributed over a logical (or hypothetical) computer system involving multiple physically independent computer systems or even multiple website locations, multiple players, multiple games and multiple locations. The amount of time spent in a peer to peer telephone conversation can be monitored, and it is possible to conduct multiple peer to peer (one to many, or many to many) relationship connections for internet conference calling. An object might also manage such information as the names and types of files being transferred between peers (as well as the peer identies) in real time, or almost virtually in real time on a distributed network depending upon communication lag times to a monitoring network server). This can permit the convenient exchange of digital media files between peers or servers and peers with billing based upon the reported transfer of files. In addition, billing for telephone communications can occur, and the link between peers is a dynamic network link that can avoid "off network" communications that may not be billed. Creating, accessing, maintaining or updating an object application model can require working with multiple translation modules and require tedious and repetitive updating of multiple individual computer systems or multiple data sources in order to do useful work and keep the object application model synchronized. Such approaches are both costly and unwieldy in terms of computing and development resources, particularly with respect to Internet based private network and virtual private network communications and transactions object application models.

Data objects of an object application model are often a feature of eCommerce object programming applications, where information is obtained from a data source and the data is defined as a data object (e.g., as a Java class) for use with another computer application. However, the present invention modifies and applies this practice to a network in order to provide network management utilizing an object programming language with objects describing and encapsulating the relationships of entities utilizing and serving the private network and persisting them to a data store. In practice, a data object or model of data objects may exist only in the random access memory of a computer memory system, or may be saved to either a data source or to some other type of retrievable information repository. A programmer or administrator of an object data application may not be able to exhaustively display the overall model or diagram of data objects for an object application model or to display some of its specific elements. Unfortunately, tools for accessing and displaying data objects and associated data object relationships as a complex data object graph model have not been well implemented in the field of object language programming.

A computer application can execute one or more of the following non-limiting actions with respect to one or more of the members selected from the group consisting of data, a data object, and a data object definition: access data, change data, create data, create a new relationship between one or more data objects by creating or changing at least one data object relationship definition, change or delete a relationship between one or more data objects by changing or deleting at least one data object relationship definition, access a data object relationship definition and use its parameters to access a data source or a data object, and access one or more data object relationship definitions or data objects to create a new data object or data object relationship. Any changes executed by a computer application with respect to one or more of the members selected from the group consisting of data, data object or data object definition may need to be properly persisted (permanently stored) to preserve any changes to one or more of the members selected from the group consisting of data, a data object and a data object definition.

A data object and an associated data object relationship definition may be represented by a complex data object graph as described in published U.S. patent application serial number 10/158,672 to Ward Mullins (assignee Thought, Inc.). A

Complex Data Object Graph (CDOG), for the purposes of this document, may be thought of as a computer program data object graph that represents a data object having at least one relationship with at least one other data object or with itself via a circular link. When the data object of a CDOG is implemented in the Java computer program language, the CDOG may be further defined as being a Java Data Object Graph ("JDOG").

There is a need for an easily navigable tool for detecting any relationships of a data object in the context of a virtual private network that provides for peer to peer communications or file transfers in order to provide both a full description and management of servers and clients, their communications and their file transfers within the network as a CDOG definition and for persisting any changes to at least one member selected from the group consisting of a data object, any data associated with the related object, or any associated CDOG definition (i.e., an changes to the data object, data or to a relationship of the data object with another data object). For example, there is a need to be able to properly create, manage and sever relationships between at least two data objects (such as peer to peer communicators) in a convenient and efficient manner by simply making a change to the CDOG definition for one of the two related data objects.

There is currently no computer programmer or system administrator tool that is efficiently adapted for managing a private network or virtual private network in peer to peer environment, or connections of a client to server environment as an object application using a CDOG model with features that permit convenient and efficient accessing and deleting of a single instance of a complex data object and (without further input) results in the tool automatically deleting all existing associated data objects and relationships and the tool automatically updating all local and distributed instances of the CDOG and its associated data, data objects and data object relationships on the local or distributed computer system, as described above. In particular, there is no such system that defines the relationship between to peers or client and server wherein the network server or the target peer is sent the link for the client who wishes to connect, then the peer or server connects to the link of the connecting client, and manages the connection. This is to be contrasted to a system wherein the connecting client or peer is asking the network to find the connection that it wants, give it a list of possible connections, and then letting the client or peer who is connectiong choose a connection link from a presented list.

Accordingly, there is a strong need in the art for a computer applications system and programmer tools designed to assist a programmer or administrator in the actions of deleting, inactivating or updating of relationships and communications within a private network or a virtual private network as a complex data object graph where the connecting client or peer sends its link address to the network and then the network manages causes the target server or peer to communicate with the connecting client or peer by initiating the connection to the connecting client or peer, wherein the computer applications programmer tool can be configured to automatically delete, inactivate or update at least one member selected from the group consisting of associated data, associated data objects and associated data object relationship definitions to reflect the deleting, inactivating or updating of a CDOG. A particularly strong need exists for such a tool or system having the further ability to be configured to persist, propagate and reflect system wide (in a local or distributed computer system) any such changes to a CDOG instance to all instances of the CDOG and to all instances of associated data, data objects and data object relationships that represent relationships and communications within a private network or a virtual private network for client to network server or peer to peer purposes.

DEFINITIONS

The following non-exhaustive list of definitions is used herein to define terms that may otherwise be confusing or can sometimes have multiple meanings. Each occurrence of a defined term in the above text, in the text that follows, or in the claims of this document, is to be given the meaning ascribed to it in the list of definitions below.

"Instance" as referred to in this document in the context of computer software applications is a single occurrence of a software logical element in the memory of a computer system, such as a "class", an "object", a "data object", and the like.

"Class" as referred to in this document in the context of computer software applications is a logic unit in a computer application or a computer software program where the application or program is based upon an objected oriented programming language (*e.g.*, Java). In practice, a class is a logical unit used as a logical template in an object oriented language from which to allocate new instances of objects.

"Object" as used in the context of this document is a general term referring to a logic unit in a computer application or a computer software program where the application or program is based upon an objected oriented programming language (*e.g.*, Java). The term "object" may ordinarily be used interchangeably with the term "class" as a template or as an instance depending on the context.

"Data object" as referred to in the context of this document represents the concept of the occurrence of an object that holds data within a specific computer application domain and is likely to have its contents stored in a persistent data source of a computer system (e.g., a database server, a binary file, a text file, or even in a combination of two or more of such a persistent data sources of a computer system). A data object may exist as an independent data object without any relationship to any other data object or it may have one or more relationships with itself or with one or more other data objects.

"Complex data object" (or "CDO") as used in the context of this document refers to the occurrence of a data object that has at least one or more relationships with itself, or at least one or more relationships with one or more other data object(s). In a given instance of a CDO at least one relationship is populated as a link, as defined below. A CDO may have a multiplicity of different relationships with itself or with one or more additional CDOs.

"Relationship" or "data relationship" as used in the context of a CDO refers to the type of logical combination that occurs between a data object with itself, or refers to the type of logical combination that occurs between a data object and at least one another data object. Among other references or descriptions, such a relationship is always referred to or partially described by a "relationship type". This term is used in an object oriented language context to reference or describe any expectations, actions and limitations possible between two or more data objects.

"Relationship type" in the context of this document is a label that specifies the possible multiple combinations that can occur between a CDO and itself or with at least one other CDO. The possible relationship type labels are 1-1 (one to one), 1-M (one to many) and M-M (many to many). A given CDO may be simultaneously related to more than one other CDO through several different types of relationship.

"Link" as used in this document with respect to a CDO identifies a particular occurrence of a relationship between a CDO and itself, between a CDO and another CDO. The occurrence of at least one populated link results in an instance of the CDO.

"Circular link" as used in this document with respect to a CDO identifies a particular occurrence of a relationship between a CDO and itself that may be direct or indirect (*e.g.*, linked to itself through another CDO).

"Relationship definition" or "relationship description" in the context of this document and computer software applications refers to information, or an abstraction of information, regarding a "relationship", "data relationship" "relationship type" or a "link" that can be stored, accessed, transferred, communicated, displayed or edited.

"Complex data object graph" or "CDOG" is a term employed herein as an abstraction to logically represent a set of complex data objects and a set of their corresponding relationships.

"Java data object graph" or "JDOG" is a term employed herein as an abstraction to logically represent a set of complex data objects and a set of their corresponding relationships that are part of a Java programming application.

"Application model" or simply "model" are essentially interchangeable terms employed herein as abstractions to logically convey a collective description or other representation for a set of complex data objects and a corresponding description or other representation of their relationships. In one respect, these terms are used logically herein provide a general way of efficiently communicating when referring to set of metadata (i.e., data about data) that describes possible data entities (*e.g.*, objects, database tables, maps, etc,) data relationship types, and data constraints involved in a computer system or application, or in a specific instance of an application. It is important to understand the context in which the terms "application model" and "model" are used in this document. Ordinarily computer engineers refer to the "model" as an abstraction rather than a specific possibility or instance of the model as applied. However, in this document for the ease of communication abstractions of the model, possible implementations of the model and instances of the model are all referred to generally as "application model" or "model". From the context of its use the term will be clear.

"Navigation", "navigating" or "navigated" in the context of the present document refers to an action implementing at least one object to interact with a set of related objects for a certain purpose, such as creation, access, insertion, modification and deletion of an object, or of one of its relationships.

"Navigation model" as used herein is a special type of application model that is applied specifically to a description (or other representation) of how objects can relate to each other and what might be the expected behavior when a CDOG is navigated for a certain purpose.

"Object schema" is a term employed herein as an abstraction referring to the set of data object classes that describe the possible data objects that can be created, modified or maintained in an application, or describing an instance of a set of data object classes in an application.

SUMMARY OF THE INVENTION

An object of the present invention is to provide a system for creating, maintaining, accessing, navigating and persisting relationships between servers, between servers and clients, and between clients, as complex data objects within a complex data object graph (CDOG) model in order to provide an improved system for peer to peer communication management, wherein clients or servers may be large computer systems or networks or may be a system that includes wireless devices such as mobile telephones, PDAs or other wireless communication devices, and wherein the network address of the end user who logs onto the network and request a peer to peer communtication or download is provided to a peer to peer server and the server connects to the end user before the network communicates to the end user that the connection is made instead of the end user choosing from a list of peer to peer servers. In a preferred embodiment, an object of the present invention is to provide such a system that can display and manage such peer to peer relationships within a private network or within a virtual private network as CDOG model for at least one network administrator or manager and the CDOG model may be represented as a graph or table of complex data objects. A further object is to provide such a system that is located on, or is part of, a local or distributed computer system that may include wireless devices not limited to PDA, mobile phones, portable computers or other wireless communication devices capable of communicating within a network.

An object of the present invention is to provide a method for creating, maintaining, accessing, navigating and persisting network relationships as complex data objects stores as a complex data object graph as described with computer software implemented methods for a peer to peer server to connect to an end user rather than an end user to choose one or more peer to peer servers from a list. In a preferred embodiment, an object of the present invention is to provide such a method having the step of displaying or printing a representation of the relationships within a private network and a virtual private network as a CDOG model as a graph or table of complex data objects with respect to internet phone communications, wherein the person being called is provided the internet address or email of the caller and connects to the caller instead of the caller connecting ot the person being called when the person being called is available to receive the call. In a further object, such a method involves a local or distributed computer system wherein individuals are identified by unique indentifiers that may include histories of their online actitivies habits, recorded individual background information, unique passwords, machine identifiers, digitial identifiers such as voice, finger print and eye scans, and the like, or any combination of one or more of such identifiers sufficient to provide a clear identity for any individual who is logging into the network to join into peer to peer activities and peer to peer servers cannot be logged into directly by the end user.

An object of the present invention is to provide a computer software program having at least one user interface and having at least one feature for at least one action selected from the group consisting of creating, maintaining, accessing, navigating and persisting relationships within a private network or within a virtual private network as complex data objects via a complex data object graft model with respect to peer to peer communications and associated data. In a preferred aspect, an object of the present invention is to provide software program with a module or associated software program having a feature for displaying or printing a representation of the relationships and associated data as a CDOG model as a graph or table of complex data objects. In a preferred object of the invention, such a software program has an editing interface for editing a CDOG model, or an editable input or source, such as a file, that can be modified to implement changes to a CDOG model and to create, modify or terminate or monitor peer to peer communications. A further object is to provide such a software program adapted for a local or distributed computer system that permits internet or intranet peer to peer communications in a secure network or virtual network setting.

An object of the present invention is to provide an application programming interface ("API"), as a preferred embodiment of the invention, wherein the API can be accessed to create, maintain, access, navigate and persist network relationships as complex data objects while representing such relationships as a CDOG model, or as an object schema that is adapted for peer to peer communications or file transfers, and file transfer reporting. In one aspect the API can be accessed by a computer software program, or by a computer software program module that is part of a software package including the API, to provide at least one action selected from the group consisting of creating, maintaining, accessing, navigating and persisting network relationships as complex data objects withing a complex data object model adapted for peer to peer communications. In a preferred aspect, an object of the present invention is to provide such an API as part of a software package having a component or method for displaying or printing a representation of such gaming relationships and associated data as a CDOG model illustrated or represented as a graph or as a table(s) of complex data objects. In another preferred object, such a software tool package including the API would be provided with an editing interface for editing the network relationships associated with gaming as a CDOG model, or has an editable input or source, such as a file, that can be modified to implement changes to the complex data object model. It is a particularly preferred object to provide such an interface as a point and click graphical user interface that may be displayed and/or edited. A further object is to provide such a API for operation on a local or distributed computer system that implements secured intranet or internet peer to peer communications.

A preferred object of the present invention is to provide a software tool or system comprising the API according (as described above), adapted for a local network or a distributed network gaming environment, wherein said software tool provides persistence for activities and data within the gaming environment in an object oriented language programming environment transparently by implementing a configurable network component capable of acquiring and persisting network relationships and data associated with intranet and internet peer to peer communications through network APIs.

A further object of the present invention is to a software tool capable of reading a source programming object logic model or a database file in a format selected from the group consisting of a UML data file,m a XMI data file, and a XML file and converting the information into a target member selected from the group consisting of a database definition XML file, a database mapping definition file, and a CDOG definition file with respect to a secured peer to peer communications intranet or internet system. In a preferred object, the software can automatically generate a persistence layer that corresponds to the object model information of the source file that will track activities, data and relationships with respect to peer to peer communications.

Another preferred object of the present invention is to provide multi-level or multilevel gaming system, comprising providing an authorized administrator of the above gaming system to conclusively identify the behavior and actions of an individual user of the system who has giving rights to the administrator to monitor peer to peer communications activities data of the individual without violating the privacy rights of the individual and record participation activities outcomes of the identified user with respect to any peer to peer communications, wherein the participation activity is one or more activities selected from the group consisting of peer to peer file sharing, peer to peer telephone communications, peer to peer email communications and the like, and without violating the privacy right of the user such that the authorized administrator can such data available for other third party users of the system having access rights..

DISCRIPTION THE INVENTION

The present invention provides a system for creating, maintaining, accessing, navigating and persisting relationships between servers, between servers and clients, and between clients, as complex data objects within a complex data object graph (CDOG) model to provide an improved system for managing peer to peer communications, wherein clients or servers may be large computer systems or networks or may be a system that includes wireless devices such as mobile telephones, PDAs or other wireless communication devices, and wherein the network address of the end user who logs onto the network and request a peer to peer communtication or download is provided to a peer to peer server and the server connects to the end user before the network communicates to the end user that the connection is made instead of the end user choosing from a list of peer to peer servers. In a preferred embodiment, the present invention provides such a system that can display and manage such peer to peer relationships within a private network or within a virtual private network as CDOG model for at least one network administrator or manager and the CDOG model may be represented as a graph or table of complex data objects. A further embodiment provides such a system that is located on, or is part of, a local or distributed computer system that may include wireless devices not limited to PDA, mobile phones, portable computers or other wireless communication devices capable of communicating within a network.

The present invention provides a method for creating, maintaining, accessing, navigating and persisting network relationships in a peer to peer environment as complex data objects stores as a complex data object graph, and permits a peer to peer server to connect to an end user in order to better manage network relationships. In a preferred embodiment, the present invention provides such a method having the step of displaying or printing a representation of the relationships within a private network and a virtual private network as a CDOG model as a graph or table of complex data objects with respect to peer to peer communication activities. In a further embodiment, such a method involves a local or distributed computer system wherein individuals are identified by unique indentifiers that may include histories of their online actitivies habits, recorded individual background information, unique passwords, machine identifiers, digitial identifiers such as voice, finger print and eye scans, and the like, or any combination of one or more of such identifiers sufficient to provide a clear identity for any individual who is logging into the network to join into peer to peer activities.

The present invention provides a computer software program having at least one user interface and having at least one feature for at least one action selected from the group consisting of creating, maintaining, accessing, navigating and persisting relationships within a private network or within a virtual private network as complex data objects via a complex data object graft model with respect to peer to peer activities and associated data. In a preferred aspect, the present invention provides software program with a module or associated software program having a feature for displaying or printing a representation of the peer to peer and peer to server relationships and associated data as a CDOG model as a graph or table of complex data objects. In a preferred aspect of the invention, such a software program has an editing interface for editing a CDOG model, or an editable input or source, such as a file, that can be modified to implement changes to a CDOG model. A further embodiment of the invention provides such a software program adapted for a local or distributed computer system that permits internet or intranet peer to peer communications in a secure network or virtual network setting.

In one embodiment the present invention provides an application programming interface ("API"), as a preferred embodiment of the invention, wherein the API can be accessed to create, maintain, access, navigate and persist peer to peer communications network relationships as complex data objects while representing such relationships as a CDOG model, or as an object schema that is adapted for peer to peer communications and file sharing reporting. In one aspect the API can be accessed by a computer software program, or by a computer software program module that is part of a software package including the API, to provide at least one action selected from the group consisting of creating, maintaining, accessing, navigating and persisting network relationships as complex data objects withing a complex data object model adapted for peer to peer communications. In a preferred aspect, the present invention is provides such an API as part of a software package having a component or method for displaying or printing a representation of such gaming relationships and associated data as a CDOG model illustrated or represented as a graph or as a table(s) of complex data objects. In another preferred aspect of the invention, such a software package including the API would be provided with an editing interface for editing the network relationships associated with gaming as a CDOG model, or has an editable input or source, such as a file, that can be modified to implement changes to the complex data object model. A particularly preferred aspect of the invention provides such an interface as a point and click graphical user interface that may be displayed and/or edited. A further aspect of the invention provides such an API for operation on a local or distributed computer system that implements secured intranet or internet peer to peer communications or file sharing.

A preferred object of the present invention is to provide a software tool or system comprising the API according (as described above), adapted for a local network or a distributed network peer to peer communications environment, wherein said software tool provides persistence for activities and data within the peer to peer communications environment in an object oriented language programming environment transparently by implementing a configurable network component capable of acquiring and persisting network relationships and data associated with intranet and internet peer to peer communications through network APIs.

Another aspect of the present invention provides end user software or a software tool capable of reading a source programming object logic model or a database file in a format selected from the group consisting of a UML data file,m a XMI data file, and a XML file and converting the information into a target member selected from the group consisting of a database definition XML file, a database mapping definition file, and a CDOG definition file with respect to a peer to peer communications intranet or internet system. In a preferred aspect, the software can automatically generate a persistence layer that corresponds to the object model information of the source file that will track activities, data and relationships with respect to peer to peer communications or file sharing activities.

In a preferred aspect, the software of the peer to peer communications network monitoring system provides a software module for individuals to obtain an thumbnail view of the data described above with respect to individual users of the system, without violating the privacy rights of the individuals that the data describes

Also, such network monitoring multi-level or multidimensional peer to peer communications system and software makes it possible for online participation in a conference group with others with respect to peer to peer activities.

In one embodiment the present invention provides a process for connecting and centrally managing a plurality of multi-tier server computer systems in a peer to peer communications system as a single virtual structured network of unlimited size having a network ring structure, a tree structure, or a combination of a network ring and network tree structure, and managing the virtual structured network of tiered server systems from a central location that has access to a repository of information with details regarding each server system included within the network, wherein said process comprises the steps of:

(a) creating one to many and many to many connections between a central server or central server network and two or more second tier server systems that may serve as a node whereby they are in turn optionally connected in a similar manner to form a tree structure, a ring network or a combination thereof, wherein nodes on the ring or tree structure can be connected to one or more sub-tier server systems,

(b) loading from or persisting to at least one storage facility system the information and connection parameters with respect to the server systems managed in the virtual structured network that are necessary to manage the virtual structure network,

(c) loading into the memory of at least one computer system in a network one or more software modules capable of:

(i) monitoring any additions, deletions or changes to server computer systems within the virtual structured network,

(ii) monitoring any additions, deletions, or changes to one or more clients connected to a server within the virtual structured network, and

(iii) persisting such information to at least one repository of information utilized to manage the virtual structured network,

(d) connecting to one or more a client systems directly to a central server system portal through an intranet or internet network connection, accessing information in a repository regarding said one or more client systems, and providing the client system internet connection address to one or more tiered servers within the virtual single network system based upon accessed information regarding said one or more client systems such that one or more tiered servers connects to the client system and notifies the client system of the connection or connections, including the steps of:

(i) accessing information in a repository regarding said one or more client systems that have connected to a central server system portal through an intranet or internet connection;

(ii) transmitting a request to transfer a connection with a client system to one or more tiered servers within said plurality of tiered server systems;

(iii) transmitting the address of the client system to one or more server systems without transmitting the one or more addresses of said available server systems connected within said virtual structured network to said client system attached to the portal; and

(iv) establishing managed connections between said one or more tiered server systems and the client system within said virtual structured network by transmitting to the tiered server systems the client system connection address,

(v) monitoring or delegating the monitoring of maintenance of connections between the client system and said one or more tiered server systems within said structured network to one or more servers within the network without providing any network server addresses to the client system that are accessible by an end user of the client system.

In a preferred embodiment, the above described process additionally comprises a step (e) of maintaining a pre-determined number of connections between each client system that is connected to one or more servers in the plurality of tiered server systems and monitoring the connections with the software of (c) to receive information from said tiered server systems and the connected client system adequate to maintain the status of connections, to establish new connections, or to both maintain the status of connections and establish new of the client system with the tiered server system when a disconnection from at least one server occurs.

In another preferred embodiment, the process described above is a process wherein each node on the virtual network that is available for connections with client systems has a pre-determined number of possible connections to existing or new clients systems as a capacity that is stored as information maintained and updated in an information repository that can be accessed by software, which is used to monitor or delegate the monitoring of maintenance of connections between the virtual network and client systems.

In a preferred peer to peer communications system embodiment, the process described above provides a step for verifying and tracking a single end user within the system with the ability to monitor and record any peer to peer activities within the system by that end user.

In another preferred embodiement, the above process is a process wherein said step (c) includes the steps of:

(f) transmitting a disconnection notification from a server system to said monitoring software of (c) that a server system has partly or fully disconnected from a client system within said plurality thereof, wherein said partly disconnected client system has been connected for receiving information from said server system;

(g) said software of (f) responding to said notification of partial disconnection by at least one of the following actions

(i) transmitting instructions to other servers on the network for such servers to initiate creating additional connections to the client system connected to the virtual network,

(ii) transmitting the address and connection information of the partially disconnected client system to other servers in order to permit them to connect with the client sytem within the virtual network, and

(iii) transmitting instructions to the servers having current connections with said client system to disconnect said client system from the virtual network

In one embodiment, a process executing within the virtual network as described above includes a process step for establishing a parent-child relationship between the virtual network and sub-branches of the of virtual network server tree system, between the virtual network and a subordinate circular network, or a combination thereof, and providing a virtual private network for a pre-determined plurality of server systems, client systems or a combination thereof within the virtual network. For example, a parent child relationship such as this can be utilized to permit registered multi-level peer to peer communication participants to be a child to the parent activity of the main peer to peer communications activity. An example, would be for a client system to attend a virtual internet concert as a child to the parent activity.

In a preferred embodiment, a process executing within said server system further providing logic and means for establishing a private network of client systems that provide peer to peer communication activities capability within a plurality thereof such that server systems and client systems may transmit information, receive information, or a combination thereof with respect to peer to peer communications, file sharing, and the like, said process comprising the steps of:

(a) receiving a request for connection to a virtual private network from a client system issuing said request;

(b) determining if said client system has permission to connect to said virtual private network and executing one or more of the following actions

(i) refusing connection to the virtual private network,

(ii) requiring actions from said requesting client system in order to receive permission to connect to said virtual private network, and

(iii) verifying that said client system has permission for connecting to said virtual private network;

(c) after verifying that client system has permission for connecting, determining if a port of said server system within said virtual private network is available for connection with said requesting client system;

(d) if a port of said server system within said virtual private network is available for connection, as determined in step (c), executing actions for connecting said client system issuing said request to said virtual private network;

(e) if a port of said server system is not available for connection within said private network, as determined in step (b), sending the address of the client system to a first pre-determined number of available servers for such servers to attempt connections with the client system within said plurality thereof connected within said private network, or suggesting postponing connection to the virtual private network; and

(f) adding information to one or more repositories within the virtual network regarding the said client system issuing said request and their level of permission to join or participate in the virtual private network.

The present may be embodied in specific forms other than those particularly described above or illustrated by the appended drawings. Upon viewing the present application preferred embodiments and other descriptions herein of the present invention, variations and other implementations that do not depart from the spirit and scope of the present invention will be apparent to one of routine skill in this field. Such variations and other implementations are considered part of the present invention and within the scope of the appended claims. Accordingly, reference should be made to the appended claims, rather than to the forgoing specification and drawings, as indicating the scope of the present invention.

I CLAIM:

1. A process for connecting and centrally managing a plurality of multi-tier server computer systems for peer to peer communications activities as a single virtual structured network of unlimited size having a network ring structure, a tree structure, or a combination of a network ring and network tree structure, and managing the virtual structured network of tiered server systems, individual end users or groups of end users from a central location that has access to a repository of information with details regarding each server system, individual end users or groups of end users included within the network, wherein said process comprises the steps of:

(a) creating one to many and many to many connections between a central server or central server network and two or more second tier server systems that may serve as a node whereby they are in turn optionally connected in a similar manner to form a tree structure, a ring network or a combination thereof, wherein nodes on the ring or tree structure can be connected to one or more sub-tier server systems,

(b) loading from or persisting to at least one storage facility system the information and connection parameters with respect to the server systems managed in the virtual structured network that are necessary to manage the virtual structure network with respect to peer to peer activities,

(c) loading into the memory of at least one computer system in a network one or more software modules capable of:

(i) monitoring any additions, deletions or changes to server computer systems within the virtual structured network,

(ii) monitoring any additions, deletions, or changes to one or more clients connected to a server within the virtual structured network, and

(iii) persisting such information to at least one repository of information utilized to manage the virtual structured network,

(d) connecting one or more a client systems directly to a central server system portal through an intranet or internet network connection, accessing information in a repository regarding said one or more client systems, individual end users, or groups of end users, and delegating the connection to one or more tiered servers within the virtual single network system based upon accessed information regarding said one or more client systems and the network address of the said one or more client systems, individual end users, groups of end users, including the steps of:

(i) accessing information in a repository regarding said one or more client systems, individual end users or groups of individual end users that have connected to a central server system portal through an intranet or internet connection;

(ii) transmitting a request to transfer a connection with a client system to one or more tiered servers within said plurality of tiered server systems;

(iii) transmitting the address of the client system to one or more available server systems connected within said virtual structured network to said client system attached to the portal, and

(iv) establishing connections between said one or more tiered server systems and the client system within said virtual structured network by instructing the one or more servers systems to connect to the network address of the client system and notify the client system of the connection or connections,

(v) monitoring or delegating the monitoring of maintenance of connections between the client system and said one or more tiered server systems within said structured network.

2. The process of claim 1, additionally comprising a step (e) of maintaining a pre-determined number of connections between each client system that is connected to one or more servers in the plurality of tiered server systems and monitoring the connections with the software of (c) to receive information from said tiered server systems and the connected client system adequate to maintain the status of connections, to establish new connections, or to both maintain the status of connections and establish new of the client system with the tiered server system when a disconnection from at least one server occurs.

3. The process of claim 1, wherein each node on the virtual network that is available for connections with client systems has a pre-determined number of possible connections to existing or new clients systems as a capacity that is stored as

information maintained and updated in an information repository that can be accessed by software, which is used to monitor or delegate the monitoring of maintenance of connections between the virtual network and client systems.

4. The process of claim 2, wherein said step (c) includes the steps of:

(f) transmitting a disconnection notification from a server system to said monitoring software of (c) that a server system has partly or fully disconnected from a client system within said plurality thereof, wherein said partly disconnected client system has been connected for receiving information from said server system;

(g) said software of (f) responding to said notification of partial disconnection by at least one of the following actions

(i) transmitting instructions to other servers on the network to creating additional connections of the client system to the virtual network,

(ii) transmitting the network address of the partially disconnected client system to one or more other servers within the virtual network to permit said one or more servers to to connect with the client system within the virtual network, and

(iii) transmitting instructions to the servers having current connections with said client system to disconnect said client system from the virtual network.

5. A process executing within the virtual network of claim 1 for establishing a parent-child relationship between the virtual network and sub-branches of the of virtual network server tree system, between the virtual network and a subordinate circular network, or a combination thereof, and providing a virtual private network for a pre-determined plurality of server systems, client systems or a combination thereof within the virtual network.

6. The process of claim 5, executing within said server system further providing logic and means for establishing a private network of client systems within a plurality thereof such that server systems and client systems may transmit information, receive information, or a combination thereof, said process comprising the steps of:

(a) receiving a request for connection to a virtual private network from a client system issuing said request;

(b) determining if said client system has permission to connect to said virtual private network and executing one or more of the following actions

(i) refusing connection to the virtual private network,

(ii) requiring actions from said requesting client system in order to receive permission to connect to said virtual private network, and

(iii) verifying that said client system has permission for connecting to said virtual private network;

(c) after verifying that client system has permission for connecting, determining if a port of said server system within said virtual private network is available for connection with said requesting client system;

(d) if a port of said server system within said virtual private network is available for connection, as determined in step (c), executing actions for connecting said client system issuing said request to said virtual private network after first obtaining the network address of said client system;

(e) if a port of said server system is not available for connection within said private network, as determined in step (b), sending to a first pre-determined number of addresses of available client system servers within said plurality thereof connected within said private network the connection address of said client system for attempted connections with the requesting client system, or suggesting postponing connection to the virtual private network; and

(f) adding information to one or more repositories within the virtual network regarding the said client system issuing said request and their level of permission to join or participate in the virtual private network.