



(22) Date de dépôt/Filing Date: 2003/05/07

(41) Mise à la disp. pub./Open to Public Insp.: 2003/11/08

(30) Priorité/Priority: 2002/05/08 (10/141,385) US

(51) Cl.Int.⁷/Int.Cl.⁷ G06F 17/30, G06F 17/60, H04L 12/16

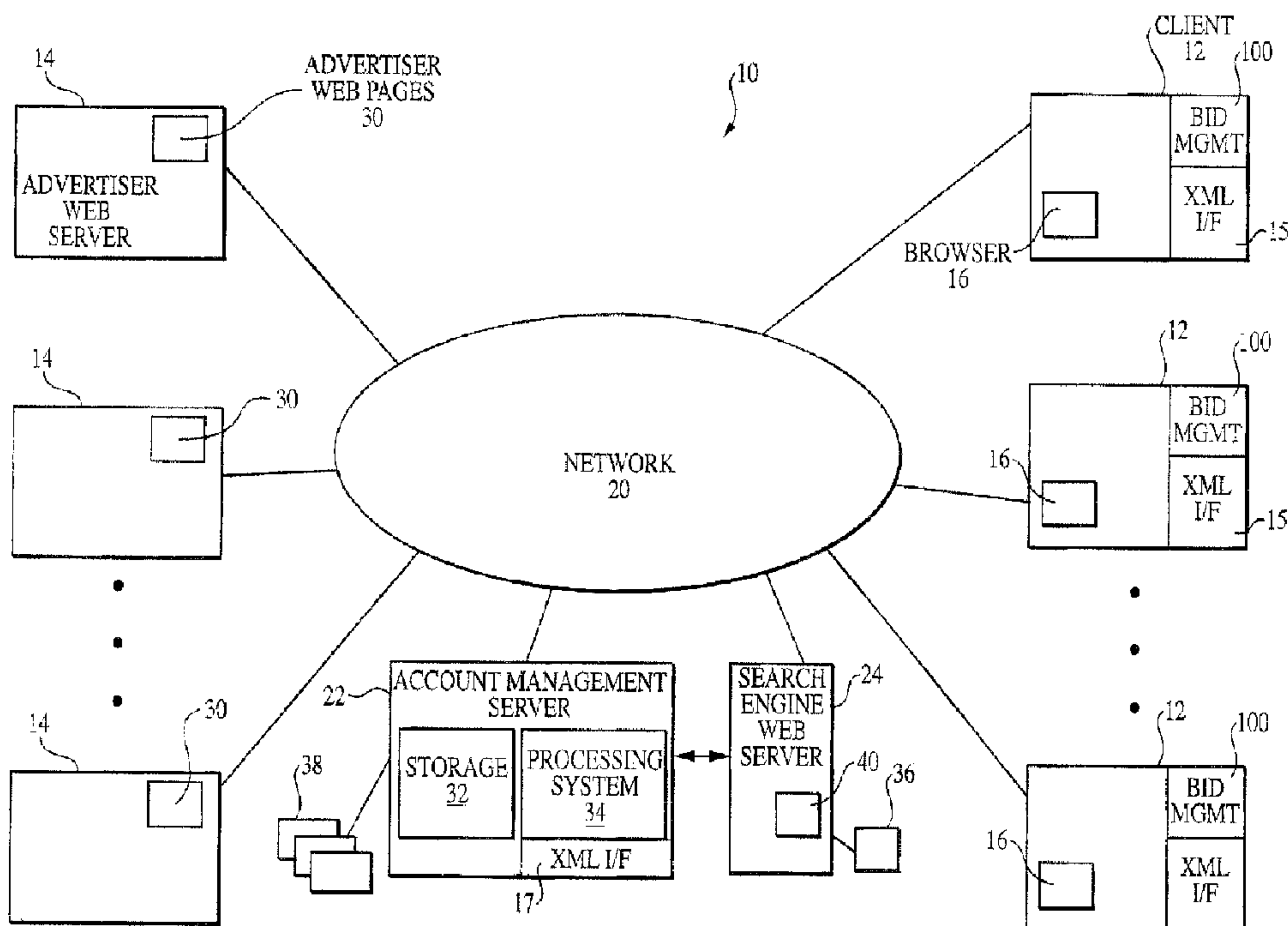
(71) Demandeur/Applicant:
OVERTURE SERVICES, INC., US

(72) Inventeurs/Inventors:
CUNNINGHAM, STEPHAN, US;
MOLINARO, ANTHONY, US;
MARITATO, FRANK JR., US;
ZHAO, PENG, US;
CONRAD, NICK, US

(74) Agent: CASSAN MACLEAN

(54) Titre : UTILISATION D'UN LANGAGE DE BALISAGE EXTENSIBLE DANS UN SYSTEME ET METHODE PERMETTANT D'INFLUER SUR UNE POSITION DANS UNE LISTE DE RESULTATS DE RECHERCHE GENEREE PAR UN MOTEUR DE RECHERCHE DE RESEAU INFORMATIQUE

(54) Title: USE OF EXTENSIBLE MARKUP LANGUAGE IN A SYSTEM AND METHOD FOR INFLUENCING A POSITION ON A SEARCH RESULT LIST GENERATED BY A COMPUTER NETWORK SEARCH ENGINE



(57) Abrégé/Abstract:

A database search apparatus and method for generating a search result list which responds to Extensible Markup Language (XML) requests from a client to a server of an on-line marketplace. A bid management tool is operable on a client computer to manage search listings and account information of one or more advertisers. The client application communicates with the server via an XML-based application program interface. The bid management tool provides fractions for reporting account activity, modifying accounts and manual, timed or event-driven changes to search listings including listings of several advertisers.

ABSTRACT OF THE DISCLOSURE

5 A database search apparatus and method for generating a search result list which responds to Extensible Markup Language (XML) requests from a client to a server of an on-line marketplace. A bid management tool is operable on a client computer to manage search listings and account information of one or more advertisers. The client application communicates with the server via an XML-based application program interface. The bid management tool provides functions for reporting account activity, modifying accounts and manual, timed or event-driven changes to search listings including listings of several advertisers.

USE OF EXTENSIBLE MARKUP LANGUAGE IN A SYSTEM AND METHOD FOR INFLUENCING A POSITION ON A SEARCH RESULT LIST GENERATED BY A COMPUTER NETWORK SEARCH ENGINE

5 BACKGROUND

This application relates generally to the area of database searching. More particularly, this application relates to the use of extensible markup language in a system and method for influencing a position on a search result list generated by a computer network search engine.

10 U.S. Patent number 6,269,361 discloses a system and method for influencing a position on a search result list generated by a computer network search engine. In one disclosed embodiment, the disclosed system and method provide an online advertiser account management tool. Search listings associated with advertisers are stored in a database. Each search listing has an associated
15 search term and an advertiser-specified bid amount. In response to a search query entered by a user, search listings with matching search terms are displayed in a search result list. Search listings are ordered from highest to lowest bid amount and may be followed in the result list by unpaid listings. The bid amount is a money amount charged to the advertiser's account when a user clicks on a search
20 listing in the search result list.

Also in accordance with a disclosed embodiment of this patent, advertisers are provided with on-line, authenticated login access to obtain account information and modify search listings. Examples of advertiser actions include viewing of past transactions, selecting notification options, adding money to the advertiser's
25 account selecting a matching option, changing a bid amount or other component of a search listing, creating or deleting search listings, receiving a cost projection for running a search listing for a specified time or obtaining activity reports. The ability of advertisers to change bid amounts results in dynamic ranking whereby the position of a search listing in a result list can be changed by increasing or
30 decreasing the associated bid amount, or as a consequence of other search listings changing their positions. U.S. Patent number 6,269,361 is commonly assigned

with the present application and is incorporated herein in its entirety by this reference.

The disclosed system thus defines an on-line marketplace operated by a marketplace operator for the benefit of advertisers and potential customers of the advertisers. The marketplace serves as a source of information for potential customers and a source of new customers for the advertisers. The marketplace is highly competitive in that advertisers compete for attention of potential customers by adjusting the bid amounts of their search listings to influence their position on a search result list generated by a search engine in response to a customer search query. One example of such an on-line marketplace is operated by Overture Systems, Inc., and is accessible on the Internet at www.overture.com.

The patented system has been very successful for advertisers seeking to reach new customers and for potential customers trying to learn more about advertiser products. In fact, the patented system has been so successful that many advertisers have placed large numbers of search listings with the on-line marketplace and employ full-time managers to manage their search listings. Third party providers have developed tools that simplify access to search listings on the on-line marketplace for advertisers. The scope of some advertiser participation in the marketplace has created a need for a degree of automation of bid management by or on behalf of advertisers.

In U.S. patent application serial number 09/922,028, filed August 3, 2001 and commonly assigned to the assignee of the present application and entitled "System and Method For Providing Place and Price Protection In a Search Result List Generated By a Computer Network Search Engine," it is proposed to allow advertisers to set a maximum cost per click (CPC) and/or a desired rank in the desired search results. Higher-ranked search listings are displayed earlier to a searcher in a set of search results and it is presumed that a higher ranking is viewed more by potential customers and is therefore more desirable. The system adjusts the CPC for a search listing to maintain the search listing at the desired rank, if that can be done without exceeding the bid or maximum CPC. If the

listing cannot be maintained at the desired rank without exceeding the bid, the system will obtain the next highest rank the bid will allow.

Further, in U.S. patent application serial number 09/963,855, entitled “Automatic Advertiser Notification for a System and Method For Providing Place and Price Protection In a Search Result List Generated By a Computer Network Search Engine,” filed September 26, 2001 and commonly assigned to the assignee of the present application, it is proposed to provide an automated agent that acts on behalf of an advertiser to monitor advertiser-specified conditions. If any condition is met or becomes true, a message is communicated to the advertiser along with some means for the advertiser to correct the undesirable condition. For example, if the agent determines that the rank for a search listing has fallen below a threshold, an E-mail message may be sent to the advertiser with an option to return an E-mail message to the system specifying how the rank condition should be corrected.

While these features provide improved convenience for advertisers trying to manage search listings, they are only limited in their success at aiding the advertiser who has a large number of search listings to manage, or for the third party who seeks to advertise search listings for multiple advertisers. Accordingly, there is a need for an improved system and method for influencing a position on a search result list generated by a computer network search engine.

BRIEF SUMMARY

By way of introduction only, one present embodiment provides a database search apparatus and method for generating a search result list that responds to eXtensible Markup Language (XML) requests from a client. XML is a flexible way to create common information formats and share both the structural model and data over a local or distributed network such as the Internet, intranets and elsewhere. XML is a formal recommendation of the World Wide Web Consortium and is similar to the Hypertext Markup Language (HTML) used in previous web pages. XML is a meta-syntax for designing syntax models that permit the structuring of data. Both XML and HTML are languages that use

markup symbols to describe the contents of a page or a file. HTML, however, describes the content of a web page only in terms of how it is to be displayed and interacted with. XML describes the content in terms of what data is being described and how it relates to the other data structures of the model. HTML and XML both use tags, which are words separated by < >, and attributes. HTML specifies a finite set of tags and meanings or uses for each tag, i.e. what each tag and attribute means. XML uses tags but XML is extensible because, unlike HTML, the tags are unlimited and self-defining.

Another present embodiment provides a bid management tool operable in conjunction with a client computer to manage search listings and account information of one or more advertisers. The bid management tool is preferably a desktop application that reports and manages paid listings on a server of an on-line marketplace. The client application communicates with the server via an XML-based application program interface. The bid management tool provides functions for reporting account activity, modifying accounts and manual, timed or event-driven changes to search listings.

The foregoing discussion of the preferred embodiments has been provided only by way of introduction. Nothing in this section should be taken as a limitation of the following claims, which define the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating the relationship between a large network and one embodiment of a system and method for generating a pay-for-placement search result;

FIG. 2 illustrates functional components of a bid management tool which may be operated in conjunction with a client computer of the system of FIG. 1;

FIG. 3 is a diagram of data for an account record for use with one embodiment of the present system and method; and

FIG. 4 illustrates an example of a search result list generated by one embodiment of the present system and method.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

Referring now to the drawings, FIG. 1 is an example of a distributed system 10 configured as client/server architecture used in one embodiment of the present invention. A client is a member of a class or group that uses the services of another class or group to which it is not related. In the context of a computer network such as the Internet, a client is a process such as a program or task that requests a service which is provided by another process, known as a server program. The client process uses the requested service without having to know any working details about the other server program or the server itself. In networked systems, a client process usually runs on a computer that accesses shared network resources provided by another computer running a corresponding server process. However, it should also be noted that it is possible for the client process and the server process to run on the same computer.

A server is typically a remote computer system that is accessible over a communications medium such as the Internet. The client process may be active in a second computer system and communicate with the server process over a communications medium that allows multiple clients to take advantage of the information-gathering capabilities of the server. Thus, the server essentially acts as an information provider for a computer network.

The block diagram of FIG. 1 therefore shows a distributed system 10 which includes a plurality of client computers 12, a plurality of advertiser web servers 14, an account management server 22, and a search engine web server 24, all of which are connected to a network 20. The network 20 will be hereinafter generally referred to as the Internet. Although the system and method of the present invention is specifically useful for the Internet, it should be understood that the client computers 12, advertiser web servers 14, account management server 22, and search engine web server 24 may be connected together through one or more of a number of different types of networks. Such networks may include local area networks (LANs), other wide area networks (WANs), and regional networks accessed over telephone lines, such as commercial information

services. The client and server processes may even comprise different programs executing simultaneously on a single computer.

The client computers 12 can be conventional personal computers (PCs), workstations, or computer systems of any other size. Each client 12 typically includes one or more processors, memories, input/output devices, and a network interface, such as a conventional modem or network interface card. The advertiser web servers 14, account management server 22, and the search engine web server 24 can be similarly configured. However, advertiser web servers 14, account management server 22, and search engine web server 24 may each include many computers connected by a separate private network. In fact, the network 20 may include hundreds of thousands of individual networks of computers.

The client computers 12 can execute web browser programs 16, such as the Netscape Navigator, Microsoft Internet Explorer, or Mosaic browser programs, to locate the web pages or records 30 stored on advertiser server 14. The browser programs 16 allow the users to enter addresses of specific web pages 30 to be retrieved. These addresses are referred to as Uniform Resource Locators, or URLs. In addition, once a page has been retrieved, the browser programs 16 can provide access to other pages or records when the user clicks on hyperlinks to other web pages. Such hyperlinks are located within the web pages 30 and provide an automated way for the user to enter the URL of another page and to retrieve that page. The pages can be data records including as content plain textual information, or more complex digitally encoded multimedia content, such as software programs, graphics, audio signals, videos, and so forth.

The client computers 12 of the illustrated embodiment include a bid management tool 100. Operation of the bid management tool 100 will be described in greater detail below in conjunction with FIG. 2.

In accordance with one embodiment, the client computers 12 each implement an XML interface 15. The XML interface 15 includes program code configured to communicate with a complementary XML interface 17 of the account management server 22 according to an established XML schema understood between the users of the client software and the operator of an online

marketplace. Examples of such schema are attached hereto as Appendix C and D, but it is understood that these schema are only examples and in no way limit the schema available to practice the present invention. As will be described below, the account management server 22 stores information about the account of each advertiser. The client computers 12 may access and update this information using the XML interface 15 in communication with the XML interface 17 of the account management server 22. A client computer may be operated by an advertiser managing the advertiser's search listings. Alternatively, the client computer may be operated by a third party managing the search listings of one or more advertisers. In this embodiment, the client computers 12 do not interact with the account management server 22 using a browser program but rather using the XML interface 15. An individual operating a client computer 12 may activate a browser program but the actual communication of data is controlled by the XML interface 15.

In one embodiment of the present invention, shown in FIG. 1, client computers 12 communicate through the network 20 with various network information providers, including account management server 22, search engine server 24, and advertiser servers 14 using the functionality provided by a HyperText Transfer Protocol (HTTP), although other communications protocols, such as FTP, SNMP, TELNET, and a number of other protocols known in the art, may be used. Preferably, search engine server 24, account management server 22, and advertiser servers 14 are located on or accessible over the Internet.

As discussed above, at least two types of server are contemplated in an embodiment of the present system and method. The first server contemplated is the account management server 22. This server 22 includes a computer storage medium 32 and a processing system 34. This server 22 further includes a variety of software program codes including the XML interface 17. These program codes are stored on one or more computer readable program storage media of the server 22, such as the storage medium 22.

A database 38 is also stored on the storage medium 32 of the account management server 22. The database 38 contains advertiser account information.

The account information stored in the database 38 includes information about the search listings of each advertiser who participates in the on-line marketplace established by the distributed system 10. This information includes search terms, bid amounts, search listing descriptions and titles and associated URLs and other information as will be described below in greater detail. Further, the account information includes information produced by operation of the marketplace system, such as current rank and current bid for each search listing, the number of clicks recorded for search listings, a calculated click through rate (CTR) and the advertiser's account balance.

It will be appreciated from the description below that the disclosed system and method may be implemented in one or more software program codes that are stored as executable instructions on a computer storage medium, such as memories or mass storage devices, on the account management server 22. XML interface 15 or a conventional browser program 16 running on client computers 12 may be used to access advertiser account information stored on account management server 22. Preferably, access to the account management server 22 is accomplished through a firewall, not shown, which protects the account management and search result placement programs and the account information from external tampering. Additional security may be provided via enhancements to the standard communications protocols such as Secure HTTP or the Secure Sockets Layer.

The second server type contemplated is a search engine web server 24. A search engine program permits network users, upon navigating to the search engine web server URL or sites on other web servers capable of submitting queries to the search engine web server 24 through their browser program 16, to type keyword queries to identify pages of interest among the billions of pages available on the Internet. In a preferred embodiment of the present invention, the search engine web server 24 generates a search result list that includes, at least in part, relevant entries obtained from and formatted by the results of the bidding process conducted by the account management server 22. The search engine web server 24 generates a list of hypertext links to documents that contain information

relevant to search terms entered by the user at the client computer 12. The search engine web server 24 transmits this list, in the form of a web page, to the network user, where it is displayed on the browser 16 running on the client computer 12. An exemplary embodiment of the search engine web server 24 may be found by navigating to the web page at URL <http://www.overture.com/>.

Search engine web server 24 is connected to the Internet 20. In one embodiment, search engine web server 24 includes a search database 40 which includes search listing records used to generate search results in response to user queries. In addition, search engine web server 24 may also be connected to the account management server 22. Account management server 22 may also be connected to the Internet. The search engine web server 24 and the account management server 22 of the present invention address the different information needs of the users located at client computers 12.

For example, one class of users located at client computers 12 may be network information providers such as advertising web site promoters or owners having advertiser web pages 30 located on advertiser web servers 14. These advertising web site promoters, or advertisers, may wish to access account information residing in storage 32 on account management server 22. An advertising web site promoter may, through the account residing on the account management server 22, participate in a competitive bidding process with other advertisers. An advertiser may bid on any number of search terms relevant to the content of the advertiser's web site. In one embodiment of the present invention, the relevance of a bidded search term to an advertiser's web site is determined through a manual editorial process prior to insertion of the search listing containing the search term and advertiser web site URL into the database 40. In an alternate embodiment of the present invention, the relevance of a bidded search term in a search listing to the corresponding web site may be evaluated using a computer program executing at processor 34 of account management server 22, where the computer program will evaluate the search term and corresponding web site according to a set of predefined editorial rules.

Higher bids receive more advantageous placement on the search result list page generated by the search engine 24 when a search using the search term bid on by the advertiser is executed. Generally, the bid for a search term is any economic value given by the advertiser associated with a search term upon occurrence of an agreed upon event. For example, in a pay for impression scheme, the advertiser gives up economic value when the advertiser's search listing is presented in search results sent to the searcher, whether or not the searcher clicks on the search listing. In another scheme, the advertiser gives up economic value when the searcher sees the advertiser's listing, clicks on the listing and then takes some further action, such as registering at the advertiser's web site or providing a credit card number, etc. The economic value may have any convenient and mutually agreeable form, such as a money amount deducted from an account, points or other counters added or subtracted from a log or account of the advertiser, etc.

In one embodiment, the amount bid by an advertiser comprises a money amount that is deducted from the account of the advertiser for each time the advertiser's web site is accessed via a hyperlink on the search result list page, or clicked. A searcher clicks on the hyperlink with a computer input device to initiate a retrieval request to retrieve the information associated with the advertiser's hyperlink. Preferably, each access or click on a search result list hyperlink will be redirected to the search engine web server 24 to associate the click with the account identifier for an advertiser. This redirect action, which is not apparent to the searcher, will access account identification information coded into the search result page before accessing the advertiser's URL using the search result list hyperlink clicked on by the searcher. The account identification information is recorded in the advertiser's account along with information from the retrieval request as a retrieval request event. Since the information obtained through this mechanism conclusively matches an account identifier with a URL in a manner not possible using conventional server system logs known in the art, accurate account debit records will be maintained. Most preferably, the advertiser's web site description and hyperlink on the search result list page is accompanied by an indication that the advertiser's listing is a paid listing. Most

preferably, each paid listing displays information labeled cost to advertiser, which is an amount corresponding to a cost per-click paid by the advertiser for each referral to the advertiser's site through the search result list.

A second class of users at client computers 12 may comprise searchers seeking specific information on the web. The searchers may access, through their browsers 16, a search engine web page 36 residing on web server 24. Alternatively, the communication may be through the XML interface of the client computer. The search engine web page 36 includes a query box in which a searcher may type a search term comprising one or more keywords. Alternatively, the searcher may query the search engine web server 24 through a query box hyperlinked to the search engine web server 24 and located on a web page stored at a remote web server. When the searcher has finished entering the search term, the searcher may transmit the query to the search engine web server 24 by clicking on a provided hyperlink. The search engine web server 24 will then generate a search result list page and transmit this page to the searcher at the client computer 12.

The searcher may click on the hypertext links associated with each listing on the search results page to access the corresponding web pages. The hypertext links may access web pages anywhere on the Internet, and include paid listings to advertiser web pages 18 located on advertiser web servers 14. In a preferred embodiment of the present invention, the search result list also includes non-paid listings that are not placed as a result of advertiser bids and are generated by a conventional Internet search engine, such as the INKTOMI, LYCOS, or YAHOO! search engines. The non-paid hypertext links may also include links manually indexed into the database 40 by an editorial team. Most preferably, the non-paid listings follow the paid advertiser listings on the search results page.

FIG. 2 illustrates functional components of a bid management tool 100 which may be operated in conjunction with a client computer 12 of the system of FIG. 1. The bid management tool 100 in the illustrated embodiment includes a plurality of menus 102, a setup function 104, a reporting function 106, a search listing management function 108 and a help function 110.

The bid management tool 100 cooperates with the XML interface 15 (FIG. 1) to report and manage paid search listings in the on-line marketplace established by the distributed system 10 described above in conjunction with FIG. 1. The bid management tool 100 is a client application that communicates with servers such as the account management server 22 and search engine web server 24 (FIG.1) by means of the XML interface 15 of the client computer 12. The bid management tool 100 provides functions for reporting account activity, modifying accounts, and manual, timed or event-driven bid changes. The bid management tool 100 can manage the search listings of one advertiser or a plurality of advertiser. Despite the convenient name applied here, the bid management tool 100 may be configured to manage all aspects of the accounts of one or more advertisers on the on-line marketplace.

Using XML communication between the client computer and a server, the bid management tool 100 establishes a downstream link from the server to the client and an upstream link from the client to the server. The downstream line carries information about the current market state and client account. The market state includes a set of search listing. Each listing, in one embodiment, includes a current rank of the advertiser's search listing among all search listings for the associated search term, a current bid, a title, a description and a URL. Other information, such as a desired rank or a maximum cost per click, may also be conveyed. Client account information includes, for example, the number of clicks recently billed to the advertiser and the account balance. Other client account information, such as a click through rate (CTR) for some specified period, may be conveyed as well. The upstream link communicates requests from the client, such as a request for a bid change or a request to add one or more new search listings for specified search terms to the advertiser's account.

The bid management tool 100 may be configured to operate on a regular schedule. For example, the bid management tool 100 can poll the remote account management server periodically, such as every five minutes. In another example, the tool 100 allows automatic bid updates to run on a predetermined schedule,

such as hourly. The user of the client computer can also initiate manual bid updates.

The management tool 100 allows a user to define groups of search terms. Such terms may be grouped according to any rules that may be established by the user. The groups of search terms may relate to particular products or services, particular advertisers, if the bids of more than a single advertiser are being managed, or any other convenient market parameter. The tool 100 further allows a user to generate reports for the defined groups and to schedule automatic updates for all terms in the group. An automatic update may adjust the current bid amount, current desired rank, or any other search listing parameter. A single instance of the tool 100 may enable one user to manage multiple advertisers, accounts, and listings. Each advertiser may have multiple accounts and each account usually holds multiple listings.

The bid management tool 100 stores information locally, on a storage medium of the client computer. The stored information includes market state, client account and group definitions. Actions initiated by the bid management tool 100 that change the market state, such as bid changes, are also locally stored and are viewable in reports produced by the tool 100.

The bid management tool 100 may be implemented in any manner appropriate for a given client computer. In one embodiment, the bid management tool 100 includes one or more computer readable program codes stored on a storage device such as a hard disk or memory of the client computer 12. The client computer includes a processor and communications interface. The processor operates in conjunction with the bid management tool program codes to perform the functions described herein. In one preferred embodiment, the bid management tool 100 is an application installable on a personal computer or other processing device operating under one or more versions of the Microsoft Windows operating system. Preferably, the tool 100 has an automatic update function which can initiate a communication session with a web site to determine if a new version of the application is available for download. If so, the user may be prompted to initiate the download and update process, which the proceeds automatically.

In FIG. 2, the bid management tool 100 includes menus 102 which permit user interaction with the bid management tool 100. Preferably, in a client computer operating under the Windows operating system, the menus 102 follow Windows menu conventions and functionality so as to simplify operation by the user. However, the menus 102 may be customized to the particular application of the bid management tool 100. In other operating systems, other menu systems may be substituted.

The menus 102 provide a user interface for data entry and option selection. One menu may be accessed to define the search terms or advertiser accounts to be managed. Another menu may be accessed to specify the format of a report. Yet another menu may be accessed to initiate an operation. Other types of menus may be provided as well. The menus interact with other data and applications stored on or accessible from the client computer, such as the XML interface 15 (FIG. 1).

Each respective menu includes appropriate fields or pop-up sub-menus of the type known in the art to receive and record entered data provided by the user. Data may be typed or otherwise entered in specified fields or selected from options provided by a pop-up menu. In addition, the menus may provide options which allow the user to simply specify all accounts of a particular advertiser. If this information is not stored locally, the bid management tool 100 may initiate a request to the account management server to obtain account identification information for the specified advertiser. For example, the bid management tool 100 may pass identifying information for the advertiser to the XML interface of the client computer. The XML interface initiates a properly formatted request and communicates the request to the account management server. Subsequently, the response is received and stored by the XML interface and the requested data is passed to the bid management tool 100.

The setup function 104 of the bid management tool 100 provides functionality for initializing and modifying the operation of the bid management tool 100. This includes defining advertisers and their associated accounts to monitor, for example, by receiving from a user a text identifier for an advertiser

and determining the advertiser's account number or receiving and storing a plurality of search terms to monitor.

The setup function 104 further permits defining groups of search terms and advertisers which may be related in any convenient way. A group is a user-defined collection of search listings. A single group can include listings from multiple accounts and advertisers. A listing may appear in more than one group. In one embodiment, all group definitions are stored locally at the client computer. In other embodiments, the group definitions may be stored all or in part at a remote location such as the account management server of the on-line marketplace. From the account management server's perspective, a group transaction will include a list of operations on individual search listings. Group contents and parameters may be specified using one or more of the menus 102 or may be established by importing a text file to the bid management tool 100 from elsewhere.

The setup function 104 further permits specifying polling operations to be conducted by the bid management tool 100. Examples include time-scheduled polling according to a predetermined schedule or polling period and event-driven polling in response to the occurrence of some specified event. The setup information which serves as inputs to the setup function 104 generally is obtained using one or more menus 104. The setup information may also be obtained from storage at the client computer or by accessing the account management server 24 using the XML interface 15 of the client computer (FIG. 1). Preferably, a password or similar information is required for access to each advertiser's account information.

Also, as noted above, the setup function 104 includes an automatic upgrade function. This may be omitted or disabled at the convenience of the user.

The bid management tool 100 further includes reporting function 106. The reporting function 106 prepares reports of using information about the advertisers, accounts and listings being managed by the bid management tool 100. Exemplary reporting formats include tabular formats in which raw data are presented and graphical formats in which the raw reporting data have been processed to provide

a more clearly understood illustration of the market state and client account information. Appearance and generation of the reports may be controlled by the menus 102.

5 The reporting function 106 in one embodiment also permits viewing of data logs maintained by the bid management tool 100. Every time a bid change is requested, manually by the user or on schedule by the bid management tool 100, an entry is added to a log file. The log file is stored at the client computer or any other convenient location. The log entry will describe either an exception, such as
10 successful bid change, including advertiser, account, term, old bid, old rank, new bid and new rank. Other information may be logged as well. The reporting function 106 permits view of the log data or reports made interpreting and presenting the log data.

15 The bid management tool 100 further includes the search listing management function 108. This function 108 implements the primary function of the bid management tool 100, the management of search listings, specifically making bid changes. In other embodiments, the search listing management function 108 also controls other transactions such as adding and deleting listing.

20 The search listing management function 108 performs both manual and automatic bid changes. Manual changes are specified by a user. A manual change is requested by identifying the listing, the account and the advertiser, and the new bid amount or other search listing parameter to be changed. This information may be entered using the menus 102. The search listing management function 108
25 responds to the manual change by interacting with the XML interface 15 which, in turn, initiates a request to the account management server 22 (FIG. 1). After the change has been made, a confirmation is conveyed from the server to the client. The confirmation is received by the XML interface 15, logged and an indication may be provided to the user.

30 By a process of automatic bid changing, the search listing management function 108 updates specified parameters of specified search listings of specified advertisers. The specifics of any automatic bid process may be established using

the menus 102. Any parameter of a search listing may be changed, including the bid amount, the desired rank, the title of the search listing, etc. Search listings to be changed may be specified by specifying a group identifier if the group contents have already been defined. The timing or events which initiate a bid change operation may be specified to control the automatic bid change process.

Each application of the bid changing function includes the following operations:

1. Wake up (start) at the sched time (e.g., once per hour).
2. See whether the local copy of the market state information is current.
3. If the local copy is out of date, update the local copy.
4. Compare the market state to specified rules to identify necessary changes.
5. Send changes to the server and log success or failure.

A user may alternatively specify intra-day and day-of-week preferences, in which the advertiser is willing to pay more per click, for example, at certain times of day or on certain days of the week. The automatic bid change function may be arranged to automatically implement these preferences.

The bid management tool 100 also includes the help function 110. The help function 110 provides conveniently available, on-line access to reference information that may be required by a user of the bid management tool 100. Examples of the information that may be provided include a list of frequently asked questions (FAQ), an index of help topics, a search function for searching the information provided by the help function, and an about routine which provides revision and other information about the bid management tool 100.

In one embodiment, the presently disclosed system is embodied as a computer readable storage medium such as a CD-Rom, hard disk drive, memory or other storage device. The storage medium includes a first program code which implements the bid management tool for managing search listings on an account management server of an on-line marketplace and a second program code which implement an extensible markup language (XML) interface for communicating with a complementary XML interface of the on-line marketplace. The program codes may be source code, object code or in any other format. The bid

management tool is preferably as described herein, but may include or omit various features and still provide equivalent functionality. The function of managing search listings on an account management server includes one or more of: retrieving search listings; retrieving the market state, retrieving a set of account
5 identifiers of one or more advertisers; modifying a bid amount or other parameter of one or more search listings; adding one or more search listings associated with an advertiser; and deleting one or more search listings associated with an advertiser.

As noted, the client computer in the illustrated embodiments communicates
10 with the account management server according to an interface 17 using XML. This interface 17 supports desktop applications of the client computer and automated tools for managing accounts with an on-line marketplace of the type described herein. The interface 17 provides a common, secure external interface at the account management server 22 (FIG. 1) for interacting with advertiser
15 systems of the server 22. The XML interface 17 of the server 22 and the XML interface 15 of the client computer are complementary so as to provide reliable, two-way communication of requests from the client to the server and responses from the server to the client.

The design and implementation of this interface 17 rely on several
20 assumptions. The interface 17 is a web page provided by the operator of the on-line marketplace. Requests to the interface 17 will be "Posted" to it following the HTTPS protocol. The client and the server send commands and replies using XML and UTF-8 character encoding. All communications follows the XML specification as defined by <http://www.w3c.org/XML/>. All applications should
25 use an XML parser that allows for a variable amount of white space, element and attribute names and values. All parties avoid trying to manually extract data from the XML documents by using patterns that require specific field names, etc. All requests sent to the server are validated against an official request schema. All responses from the server are validated against the response schema. Any requests
30 coming in to the account management server that do not follow the request schema are immediately rejected.

The examples provided herein relate to the DirecTraffic Center advertiser facility provided by Overture Services, Inc. It is well within the purview of those ordinarily skilled in the art to modify and extend these examples for application to other systems and other service providers.

5

Posting to the account management server

The interface 17 defines a number of HTTP headers and parameters that are required when expecting a response from the account management server 22. The Content-Type header is required for all POST requests to the server. In one
10 embodiment, the value of this header is "application/x-www-form-urlencoded". Also the Content-Length header should be specified and reflect the number of bytes being sent to the server. More information is available at the HTTP1.1 specification at <ftp://ftp.isi.edu/in-notes/rfc2616.txt>. The following is a list of other parameters used for posting to the account management server 22 and a brief
15 description of each.

xml

Required. This parameter contains the XML document to be sent to the account management server. If the Content-Type header being sent is "application/x-www-form-urlencoded", then the value of this parameter must be URL encoded.

20

/go2/xml/XMLRequestHandler.submit

_D:/go2/xml/XMLRequestHandler.submit

Required. In this embodiment, the application server uses these parameters internally. The values specified for each should be " " (space).

25

contentType

Optional. The value of this parameter can be either "text/plain" or "text/xml" (the default).

30

Example POST:

POST /s/dtc/xml/index.jhtml?_DARGS=%2Fs%2Fdtc%2Fxml%2Findex.jhtml

HTTP/1.0

Content-Length: 404

Content-Type: application/x-www-form-urlencoded

5

xml=%3c%3fxml+version%3d%221.0%22+encoding%3d%22UTF-
8%22%3f%3e%3cDTCRequest++xmlns%3axsi%3d%22http%3a%2f%2fwww.w3.or
g%2f2001%2fXMLSchema-
instance%22++version%3d%221.0%22++username%3d%22gototest%22++password
10 %3d%22qblahblaht%22%3e++%3cActions%3e++++%3cGetAccountIds%2f%3e++
%3c%2fActions%3e%3c%2fDTCRequest%3e&_D:/go2/xml/XMLRequestHandler.s
ubmit=+&/go2/xml/XMLRequestHandler.submit=&contentType=text%2fplain

Order of Operations

15

In general, there is no specific order in which commands need to be submitted to the XML server. The server processes requests in the order in which they are received. There is, however, a logical order that clients to the XML server may want to follow.

20

Before any listings can be retrieved or bid prices adjusted, the client computer retrieves the set of account identifiers to work with. In one embodiment, the server provides the account identifier and the marketplace that the account is valid for.

25

Once the client computer has the list of account identifiers to work on, the client computer may retrieve the set of listings for that account. This will provide the important listingId attribute that will be necessary for SetListing transactions. This listingId is static (i.e. it does not change), so the same listingId may be used forever to refer to a specific listing. If the listing is deleted and that listingId is used, an error will be returned. This function also provides the searchTerm attribute that will be necessary for using the market state functionality.

30

Once the client computer has the set of listings and search terms, the client computer may get the current market state for the listings of interest. This function provides the set of search listings in the order as it would appear to a searcher receiving search results in response to a search query to the search engine web server 24 (FIG. 1). This set of search listings includes listings that do not

belong to the current advertiser. The server designates the listings that the current advertiser owns by supplying the listingId.

Based on the market state, the client computer may set the bid price for each listing. One embodiment only allows a one time, fixed bid price changes requests for a listing. Other embodiments allow more than the attribute or parameter of a search listing to be changed.

Authentication

In the illustrated embodiment, the first bit of information that must be provided for every request is a version string, a login username and a password. This information must be provided in the root level DTCRequest XML tag sent by the client. All commands sent to the server should be contained in this root level tag. If any of the information in the root tag is missing or incorrect, the request will be rejected and all commands contained therein will be ignored.

For example,

```
<DTCRequest version="1.0" username="testuser" password="test password">
```

```
<!-- queries and commands go here... -->
```

```
</DTCRequest>
```

The version is a string describing the version of the XML interface 17. If it does not match the version the account management server 22 is using, an error will be sent and all commands contained in the DTCRequest will be ignored.

The username corresponds to a pre-existing username. The password should be the same password the user would use to login to account management server. If the username or password is not provided or incorrect, a response will be sent immediately and all commands contained within the DTCRequest will be ignored. The response may have the form:

```
<DTCResponse success="false" reason="Login failed"/>
```

In embodiments which give access to administrators, if the username and password provided belong to an administrator, the administrator will have the ability to perform any of the below actions for any user account.

If the login and version verification phases succeed, a successful response will be sent and all contained commands will be processed:

```
<DTCResponse success="true">
...<!-- processed command responses here -->
</DTCResponse>
```

Getting the Set of Account IDs

It may be possible that the user does not know the set of account IDs required for future commands. This function permits a query for a list. Administrators will need to provide a username for which to retrieve account IDs. For example,

```
<Actions>
  <GetAccountIds dtcUsername="joebob"/>
</Actions>
```

Normal, non-administrative users would not provide the username because the server will have it from the DTCRequest tag:

```
<Actions>
  <GetAccountIds/>
</Actions>
```

If a non-administrator user specifies a dtcUsername, it will be rejected with an error code of "Permission Denied".

The response to the above request will look like this:

```
<ActionsResponse>
  <GetAccountIdsResponse success="true">
    <Account id="12345" market="US"/>
    <Account id="af3456" market="UK"/>
  </GetAccountIdsResponse>
</ActionsResponse>
```

The market field is an enumeration (defined in the Schema) denoting which marketplace the account is set for.

Retrieving Listings

In order to change the characteristics of a listing, the user must first do a query to retrieve the listing. Any requests for listings is contained in an Actions XML tag. The Actions tag contains the accountId for which all the contained queries and commands apply to. The accountId is validated against the allowed list of accountIds for a normal user. Administrators may work on any set of accountIds.

It is possible to grab a set of listings based on certain criteria, or if no criteria is specified, all listings for the specified accountId. The maximum number of listings returned if the maxCount attribute is not specified is 40. If no starting index is specified, it starts with result 1. This function by default does not return current rank for each listing. To get this information, the attribute withRank is specified with a value of "true".

Examples:

1. Get all listings (up to the maximum) for account id 12345

```
<Actions accountId="12345">
  <GetListings/>
</Actions>
```

2. Get all listings to a maximum of 10 that have "car" contained in the search term for account id 12345

```
<Actions accountId="12345">
  <GetListings searchTerm="car" maxCount='10'/>
</Actions>
```

3. Get all listings to the maximum allowed that have a bid price between 0.05 and 0.10 with the current rank information.

```
<Actions accountId="12345">
  <GetListings lowBid="0.05" highBid="0.10" withRank="true"/>
```

</Actions>

Other valid criteria to search on include:

- Url
- Title
- Description

Search criteria not based on bid price match if the provided string "is contained" in that field of the search listing. Search criteria based on bid price will select listings 'greater than or equal to' the price specified in the lowBid attribute and 'less than or equal to' the price specified in the highBid attribute.

Upon successful completion, a response similar to the following will be returned:

```
<ActionsResponse success="true">
  <GetListingsResponse success="true">
    <Listing index='1' listingId="a2311" .../>
    <Listing index='2' listingId="123ac345" rank="3".../>
  </GetListingsResponse>
</ActionsResponse>
```

The listingId should be used to refer to a particular line ad when changing its characteristics as with a SetListing request (described below).

Getting the Market State

The GetMarketState function is designed to give a snapshot of the current state for a particular search term. This can be helpful in viewing the price differences between the different ranks so one can change their bid accordingly. This function takes a marketplace id (required) and a search term (required) and returns the market state as reported by the overture consumer site. For example,

1. Show me the current listings ranked 1-5 for the US marketplace and the search term "cars".

```
<GetMarketState market='0' searchTerm="cars" maxCount='5'/>
```

The response would be something like this:

```

5  <GetMarketStateResponse success="true">
    <Listing rank="1" title="InvoiceDealers.com - Buy New Cars Direct"
description="Quick, easy, painless... It&apos;s new car buying made easy at
InvoiceDealers.com! Get new car pricing before you visit the dealer at
InvoiceDealers.com." siteHost="www.invoicedealers.com" bid="0.43"
currency="USD" />
10  <Listing rank="2" title="AutoMall Online - Instant Online Prices"
description="Since 1994! The smartest way to buy a car. Online instant dealer price
quotes with registration. Guaranteed lowest prices on the Internet. Over 5,000 quality
dealers." siteHost="www.automallonline.com" bid="0.42" currency="USD" />
15  <Listing rank="3" title="Extended Warranty for New or Used Cars"
description="Get extended car warranty coverage for up to seven years or 150,000
miles. Save up to 60% off dealer prices. Click here for a free quote from the No. 1
online provider." siteHost="www.warrantygold.com" bid="0.38" currency="USD" />
20  <Listing rank="4" title="New Car - Get Lowest Dealer Price Fast"
description="Ready to buy? Get multiple price quotes on a new car from local and
online dealers fast. Submit simple, no-obligation forms powered by the leading
automobile sites. Compare for best deal." siteHost="www.pricequotes.com"
bid="0.37" currency="USD" />
25  <Listing rank="5" title="Lexus.com - Official Site" description="Explore the
models, build your Lexus, search for a certified pre-owned Lexus, or find a dealer."
siteHost="mojofarm.mediaplex.com" bid="0.36" currency="USD" />
</GetMarketStateResponse>

```

Setting the Bid Price For A Listing

30 In one embodiment the XML interface only allows one-time, fixed bid price changes for a particular listing. Other embodiments allow modification of other fields, other bid behaviors, etc.

35 To change the bid price, the user provides an Actions tag with the account number that contains the listing(s) that will be changed. The accountId attribute is validated against the username and password provided in previous steps. In the SetListing tag, the listingId, as provided in the GetListings response, is specified. The next element required is the BidBehavior element, followed by the 'Fixed' element that requires the bid to be specified as an attribute.

For example,

```

40  <Actions accountId="123">
    <SetListing listingId="a123b455">

```

```

    <BidBehavior>
      <Fixed bid="0.50"/>
    </BidBehavior>
  </SetListing>
</Actions>

```

In one embodiment, referred to as Bid to Premium, the user can specify that the search listing always appear in the first three search listings presented with the search results. If such a change is desired, the 'B2P' element is supplied instead of the 'Fixed' element. For the B2P element, the desired rank and the maxCap (the maximum amount the advertiser is willing to pay to get to the desired rank) are required. For example,

```

<Actions accountId="123">
  <SetListing listingId="a123b455">
    <BidBehavior>
      <B2P rank="1" maxCap="0.50"/>
    </BidBehavior>
  </SetListing>
</Actions>

```

Upon successful completion, a response will come back similar to the following:

```

<ActionsResponse success="true">
  <SetListingResponse listingId="a123b455" success="true"/>
</ActionsResponse>

```

If it is not successful, the system provides a sentence describing the failure:

```

<ActionsResponse success="true">
  <SetListingResponse listingId="a123b455" success="false" reason="Bid must be in
the format ###"/>
</ActionsResponse>

```

Appendix A, attached hereto, provides a set of exemplary requests that might be posted by a client to the account management server. Similarly, attached Appendix B provides a set of exemplary responses that might be returned to the client from the server in response to a posted request. Appendix C provides an exemplary XML schema for requests submitted by a client to the server.

Appendix D provides an exemplary XML schema for responses by the server to the client. Each of these appendices are intended to be illustrative only and not limiting to the scope of the invention.

FIG. 3 is a diagram showing the types of information contained in each advertiser account record 300 in the search database 40 (FIG. 1). This database 40 includes search listing records used to generate search results in response to user queries. First, an advertiser account record 300 contains a username 302 and a password 304, used for online authentication as described above. The account record also contains contact information 310 such as contact name, company name, street address, phone, e-mail address.

Contact information 310 is preferably utilized to direct communications to the advertiser when the advertiser has requested notification of key advertiser events. The account record 300 also contains billing information 320, such as current balance, credit card information. The billing information 320 contains data accessed when the advertiser selects the option to add money to the advertiser's account. In addition, certain billing information such as the current balance, may trigger events requiring notification under the notification option. The audit trail section 325 of an account record 300 contains a list of all events wherein the account record 300 is accessed. Each time an account record 300 is accessed or modified by an administrator or advertiser a short entry describing the account access and/or modification event will be appended to the audit trail section 330 of the administrator or advertiser account that initiated the event. The audit trail information may then be used to help generate a history of transactions made by the account owner under the account.

The advertising information section 330 contains information needed to conduct the online bidding process of the on-line marketplace, wherein a position is determined for a web site description and hyperlink within a search result list generated by a search engine. The advertising data 330 for each user account 300 may be organized as zero or more subaccounts 340. Each subaccount 340 includes at least one search listing 344. Each search listing corresponds to a bid on a search term. An advertiser may utilize subaccounts to organize multiple bids

on multiple search terms, or to organize bids for multiple web sites. Subaccounts are also particularly useful for advertisers seeking to track the performance of targeted market segments. The subaccount superstructure is introduced for the benefit of the advertisers seeking to organize their advertising efforts, and does not affect the method of operation of the disclosed system and method. Alternatively, the advertising information section need not include the added organizational layer of subaccounts, but may simply include one or more search listings.

The search listing 344 corresponds to a search term and associated bid and contains key information to conduct the online competitive bidding process. In one embodiment, each search listing comprises the following information: search term 352, web site description 354, URL 356, bid amount 358, and a title 360. The search term 352 comprises one or more keywords which may be common words in English or any other language. Each keyword in turn comprises a character string. The search term is the object of the competitive online bidding process. The advertiser selects a search term to bid on that is relevant to the content of the advertiser's web site. Ideally, the advertiser may select a search term that is targeted to terms likely to be entered by searchers seeking the information on the advertiser's web site, although less common search terms may also be selected to ensure comprehensive coverage of relevant search terms for bidding.

The web site description 354 is a short textual description of the content of the advertiser's web site. The description 354 may be displayed as part of the advertiser's entry in a search result list. The search listing 344 may also contain a title 360 of the web site that may be displayed as the hyperlinked heading to the advertiser's entry in a search result list. The URL 356 contains the Uniform Resource Locator address of the advertiser's web site. When the user clicks on the hyperlink provided in the advertiser's search result list entry, the URL is provided to the browser program. The browser program, in turn, accesses the advertiser's web site by redirecting the browser to the web site specified by the URL. The URL may also be displayed as part of the advertiser's entry in a search result list.

The bid amount 358 in one embodiment is a money amount bid by an advertiser for a listing. This money amount is deducted from the advertiser's prepaid account or is recorded for advertiser accounts that are invoiced for each time a search is executed by a user on the corresponding search term and the search result list hyperlink is used to refer the searcher to the advertiser's web site. In other embodiments, the bid amount may be any other type of economic value given by the advertiser or received by the operator of the on-line marketplace.

Finally, a rank value is a value generated dynamically, preferably by the processing system 34 of the account management server 22 shown in FIG. 1, each time an advertiser places a bid or a search enters a search query. The rank value of an advertiser's search listing determines the placement location of the advertiser's entry in the search result list generated when a search is executed on the corresponding search term. Preferably, rank value is an ordinal value determined in a direct relationship to the bid amount 358; the higher the bid amount, the higher the rank value, and the more advantageous the placement location on the search result list. Most preferably, the rank value of 1 is assigned to the highest bid amount with successively higher ordinal values (e.g., 2, 3, 4, . . .) associated with successively lower ranks and assigned to successively lower bid amounts.

An example of a search result list display used in an embodiment of the present invention is shown in FIG.4, which is a display of the first several entries resulting from a search for the term "zip drives". As shown in FIG. 4, a single entry, such as entry 710a in a search result list consists of a description 720 of the web site, preferably comprising a title and a short textual description, and a hyperlink 730 which, when clicked by a searcher, directs the searcher's browser to the URL where the described web site is located. The URL 740 may also be displayed in the search result list entry 710a, as shown in FIG. 4. The click through of a search result item occurs when the remote searcher viewing the search result item display 710 of FIG. 4 selects, or clicks on the hyperlink 730 of the search result item display 710. In order for a click through to be completed, the searcher's click should be recorded at the account management server and redirected to the advertiser's URL via the redirect mechanism discussed above.

Search result list entries 710a-710h may also show the rank value of the advertiser's search listing. The rank value is an ordinal value, preferably a number, generated and assigned to the search listing by the processing system 34 of FIG. 1. Preferably, the rank value is assigned through a process, implemented in software, that establishes an association between the bid amount, the rank, and the search term of a search listing. The process gathers all search listings that match a particular search term, sorts the search listings in order from highest to lowest bid amount, and assigns a rank value to each search listing in order. The highest bid amount receives the highest rank value, the next highest bid amount receives the next highest rank value, proceeding to the lowest bid amount, which receives the lowest rank value. Most preferably, the highest rank value is 1 with successively increasing ordinal values (e.g., 2, 3, 4, . . .) assigned in order of successively decreasing rank. The correlation between rank value and bid amount is illustrated in FIG. 4, where each of the paid search list entries 710a through 710f display the advertiser's bid amount 750a through 750f for that entry. Preferably, if two search listings having the same search term also have the same bid amount, the bid that was received earlier in time will be assigned the higher rank value. Unpaid listings 710g and 710h do not display a bid amount and are displayed following the lowest-ranked paid listing. Preferably, unpaid listings are displayed if there are an insufficient number of listings to fill the 40 slots in a search results page. Unpaid listings are generated by a search engine utilizing objective distributed database and text searching algorithms known in the art. An example of such a search engine may be operated by Inktomi Corporation. The original search query entered by the remote searcher is used to generate unpaid listings through the conventional search engine.

From the foregoing, it can be seen that the presently disclosed embodiments provide an improved method and apparatus for controlling the display of search results in a search result list. The system has been improved through addition of an XML interface at the account management server and at client computers. Communication between the server and clients to control the search listings of one or more advertisers is in accordance with one or more

predetermined XML schemas. The schemas define the parameters and possible data values to be used in managing advertiser accounts and search listings. In this manner, groups of search listings for multiple advertisers may be managed effectively by a single user. Moreover, automatic operations may be specified for updating search listings, obtaining the market state, receiving account information and producing reports. The disclosed system and method may be used both by advertisers to manage their own accounts and search listings and by third parties to manage the accounts and search listings of one or more advertisers.

It is therefore an advantage of the present invention to provide reliable, two-way communication of requests and responses between an account management bid tool client and an account management bid tool server through the use of complementary XML interfaces on both the client and server sides. It is a further advantage of the present invention to provide across a distributed network a common, secure external server interface for use by advertiser client systems in order to perform account management functions with an online advertising marketplace including retrieving search listings, retrieving the market state, retrieving a set of account identifiers of one or more advertisers, modifying a bid amount or other parameter of one or more search listings, adding one or more search listings associated with an advertiser and deleting one or more search listings associated with an advertiser. A further advantage of the present invention is to allow the automation of such account management functions by providing a common schema for creating requests to the server and another common schema for understanding responses from the server.

While a particular embodiment of the present invention has been shown and described, modifications may be made. It is therefore intended in the appended claims to cover all such changes and modifications which fall within the true spirit and scope of the invention.

Appendix A: Request examples**Get Account Ids (Normal user)**

```

5    <?xml version="1.0" encoding="UTF-8"?>
    <DTCRequest
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      version="1.0"
      username="testacct"
10   password="fictionalpass">

      <Actions>
        <GetAccountIds/>
      </Actions>
15  </DTCRequest>

```

Get Account Ids (Admin user)

```

20  <?xml version="1.0" encoding="UTF-8"?>
    <DTCRequest
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      version="1.0"
      username="testadminuser"
      password="fictionalpass">
25
      <Actions>
        <GetAccountIds dtcUsername="jimbob"/>
      </Actions>
    </DTCRequest>
30

```

Get Listings

```

    <?xml version="1.0" encoding="UTF-8"?>
    <DTCRequest
35   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      version="1.0"
      username="testacct"
      password="fictionalpass">

40   <Actions accountID="10078815">
      <!-- get all listings by search term -->
      <GetListings maxCount="40" searchTerm="coupon"/>
      <!-- get all listings by url -->
      <GetListings maxCount="40"
45   url="http://www.goto.com"/>
    </Actions>
  </DTCRequest>

```

```

    <!-- get all listings by title words with current
rank info -->

```

```

    <GetListings maxCount="40" title="zero"
withRank="true"/>

```

```

5    </Actions>

```

```

</DTCRequest>

```

Get Market State

```

10

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<DTCRequest

```

```

  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

  version="1.0"

```

```

15

```

```

  username="testacct"

```

```

  password="fictionalpass">

```

```

  <Actions>

```

```

    <GetMarketState searchTerm="coupon" market="US"/>

```

```

20

```

```

  </Actions>

```

```

</DTCRequest>

```

Set Listings

```

25

```

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<DTCRequest

```

```

  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

  version="1.0"

```

```

30

```

```

  username="testacct"

```

```

  password="fictionalpass">

```

```

  <Actions accountID="10078815">

```

```

    <!-- Change bid to $1.50 -->

```

```

35

```

```

    <SetListing listingID="29153393">

```

```

      <BidBehavior>

```

```

        <Fixed bid="1.50"/>

```

```

      </Bidbehavior>

```

```

    </SetListing>

```

```

40

```

```

    <SetListing listingID="29153323">

```

```

      <BidBehavior>

```

```

        <B2P maxCap="1.50" rank="1"/>

```

```

      </BidBehavior>

```

```

    </SetListing>

```

```

45

```

```

  </Actions>

```

```

</DTCRequest>

```

Appendix B: Server Response Examples**Get Account Ids (Normal user)**

```

5      <?xml version="1.0" encoding="UTF-8"?>
      <DTCResponse
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:noNamespaceSchemaLocation="dtc.xsd"
        success="true">
          <ActionsResponse>
10         <GetAccountIdsResponse success="true">
              <Account id="12345" market="US"/>
              <Account id="af3456" market="UK"/>
            </GetAccountIdsResponse>
          </ActionsResponse>
15    </DTCResponse>

```

Get Account Ids (Admin user)

```

20    <?xml version="1.0" encoding="UTF-8"?>
      <DTCResponse
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:noNamespaceSchemaLocation="dtc.xsd"
        success="true">
          <ActionsResponse>
25         <GetAccountIdsResponse success="true">
              <Account id="12345" market="US"/>
              <Account id="af3456" market="UK"/>
            </GetAccountIdsResponse>
          </ActionsResponse>
30    </DTCResponse>

```

Get Listings

```

35    <?xml version="1.0" encoding="UTF-8"?>
      <DTCResponse
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:noNamespaceSchemaLocation="dtc.xsd"
        success="true">
          <ActionsResponse>
40         <GetListingsResponse success="true">
              <Listing listingID="29153391"
                url="http://mappedtocouponurl.com/"
                searchTerm="best web site for coupon"
                bid="0.13" title="Title mapped to
45         'coupon'"
                description="Desc mapped to 'coupon'"

```

```

        market="US" online="true"/>
    <Listing listingID="29153393"
        url="http://mappedtocouponurl.com/"
        searchTerm="coupon" bid="0.49"
5         title="Title mapped to 'coupon'"
        description="Desc mapped to 'coupon'"
        market="US" online="true"/>
    </GetListingsResponse>
    <GetListingsResponse success="true">
10     <Listing listingID="26929544"
        rank="3"
        url="http://www.goto.com/"
        searchTerm="gototest123456789"
        bid="0.05" title="test"
15     description="test." market="US"
        online="true"/>
    </GetListingsResponse>
    </ActionsResponse>
    </DTCResponse>
20

    Get Market State

    <?xml version="1.0" encoding="UTF-8"?>
    <DTCResponse
25     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="dtc.xsd"
        success="true">
        <ActionsResponse>
            <GetMarketStateResponse success="true">
30                <Listing rank="1" title="Print Free Coupons from
Your Computer!" description="Print free coupons from
your computer at CoolSavings! You&apos;ll save big on
groceries, clothes, baby and kid&apos;s stuff, home
items and much more! Click here to enroll. It&apos;s
35                free!" siteHost="www.coolsavings.com" bid="0.39"
                    currency="USD" />
                <Listing rank="2" title="Get Free Local Coupons
at ClipACoupon!" description="It&apos;s totally free!
Enroll now to print free money saving coupons when you
40                want or need them. Print free coupons or receive great
                    online deals from our local and national merchants."
                    siteHost="www.clipacoupon.com" bid="0.27"
                    currency="USD" />
                <Listing rank="3" title="The Online Coupon
45                Resource" description="Click here to visit
100GreatCoupons.com. We can help to save you money on

```

every online purchase from major online retailers like
Amazon.com, BarnesandNoble.com, and Half.com."
siteHost="www.100greatcoupons.com" bid="0.27"
currency="USD" />

5 </GetMarketStateResponse>
 </ActionsResponse>
 </DTCResponse>

Set Listings

10 <?xml version="1.0" encoding="UTF-8"?>
 <DTCResponse
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="dtc.xsd"
15 success="true">
 <ActionsResponse success="true">
 <SetListingResponse listingId="29153393"
20 success="true"/>
 <SetListingResponse listingID="29153323"
 success="true"/>
 </ActionsResponse>
 </DTCResponse>

Appendix C: Exemplary Request Schema

```

5  <?xml version="1.0" encoding="UTF-8"?>
    <!--
    *****
    ** -->
    <!-- Copyright 2001, Overture
    -->
    <!--
    -->
10  <!-- An XML Schema for bidding tools to programmatically access the
    features -->
    <!-- of DTC.
    -->
    <!--
15  *****
    ** -->

    <xsd:schema
20      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      elementFormDefault="qualified">

        <xsd:element name="DTCRequest" type="DTCRequestType"/>

        <!-- ***** Request Types ***** -->
25  <xsd:complexType name="RequestType">
      <xsd:attribute name="aux" type="NonEmptyString" use="optional"/>
    </xsd:complexType>

    <xsd:complexType name="DTCRequestType">
30  <xsd:complexContent>
      <xsd:extension base="RequestType">
        <xsd:sequence>
          <xsd:element name="Actions"
35  type="ActionType"
            minOccurs='1'
            maxOccurs='unbounded' />
        </xsd:sequence>
        <xsd:attribute name="version" type="NonEmptyString"
40  use="required"/>
        <xsd:attribute name="username" type="NonEmptyString"
        use="required"/>
        <xsd:attribute name="password" type="NonEmptyString"
        use="required"/>
        </xsd:extension>
45  </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="ActionType">
50  <xsd:complexContent>
      <xsd:extension base="RequestType">
        <xsd:sequence>
          <xsd:element name="GetAccountIds"
55  type="AccountIdType"
            minOccurs='0'
            maxOccurs='unbounded' />
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexType>

```

```

5      <xsd:element name="GetMarketState"
        type="MarketStateType"
        minOccurs="0"
        maxOccurs="unbounded"/>
      <xsd:element name="GetListings"
        type="GetListingType"
        minOccurs="0"
        maxOccurs="unbounded"/>
10     <xsd:element name="SetListing"
        type="SetListingType"
        minOccurs="0"
        maxOccurs="unbounded"/>
      <xsd:element name="AddListing"
        type="AddListingType"
        minOccurs="0"
        maxOccurs="unbounded"/>
15     <xsd:element name="DeleteListing"
        type="DeleteListingType"
        minOccurs="0"
        maxOccurs="unbounded"/>
20   </xsd:sequence>
      <xsd:attribute name="accountId" type="NonEmptyString"
use="optional"/>
    </xsd:extension>
25  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="AddListingType">
  <xsd:complexContent>
30    <xsd:extension base="RequestType">
      <xsd:attribute name="title"
        type="NonEmptyString"
        use="required"/>
      <xsd:attribute name="description"
35        type="NonEmptyString"
        use="required"/>
      <xsd:attribute name="url"
        type="NonEmptyString"
        use="required"/>
40    <xsd:attribute name="searchTerm"
        type="NonEmptyString"
        use="required"/>
      <xsd:attribute name="bid"
45        type="BidType"
        use="required"/>
      <xsd:attribute name="isAdult"
        type="xsd:boolean"
        use="optional"/>
    </xsd:extension>
50  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="DeleteListingType">
  <xsd:complexContent>
55    <xsd:extension base="RequestType">
      <xsd:attribute name="listingId" type="NonEmptyString"
use="required"/>
    </xsd:extension>

```

```

    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="AccountIdType">
5    <xsd:complexContent>
      <xsd:extension base="RequestType">
        <!-- The dtcUsername attribute is valid only for
administrative -->
10      <!-- users. Any other time the username is specified, it
-->
        <!-- will be ignored.
-->
        <xsd:attribute name="dtcUsername" type="NonEmptyString"
use="optional"/>
15      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="SetListingType">
20    <xsd:complexContent>
      <xsd:extension base="RWListingType">
        <xsd:sequence>
          <xsd:element name="BidBehavior"
25            type="BidBehaviorType"
            minOccurs="0"
            maxOccurs="1"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
30  </xsd:complexType>

  <xsd:complexType name="MarketStateType">
    <xsd:complexContent>
      <xsd:extension base="RequestType">
35        <xsd:attribute name="searchTerm" type="NonEmptyString"
use="required"/>
        <xsd:attribute name="market" type="MarketType"
use="required"/>
        <xsd:attribute name="startIndex" type="xsd:integer"
40 use="optional"/>
        <xsd:attribute name="maxCount" type="xsd:integer"
use="optional"/>
      </xsd:extension>
    </xsd:complexContent>
45  </xsd:complexType>

  <xsd:complexType name="GetListingType">
    <xsd:complexContent>
      <xsd:extension base="RequestType">
50        <xsd:attribute name="title" type="NonEmptyString"
use="optional"/>
        <xsd:attribute name="description" type="NonEmptyString"
use="optional"/>
        <xsd:attribute name="url" type="NonEmptyString"
55 use="optional"/>
        <xsd:attribute name="lowBid" type="BidType"
use="optional"/>

```

```

        <xsd:attribute name="highBid"      type="BidType"
use="optional"/>
        <xsd:attribute name="maxCount"    type="xsd:integer"
use="optional"/>
5      <xsd:attribute name="searchTerm"    type="NonEmptyString"
use="optional"/>
        <xsd:attribute name="market"      type="MarketType"
use="optional"/>
10     <xsd:attribute name="startIndex"    type="xsd:integer"
use="optional"/>
        <xsd:attribute name="withRank"    type="xsd:boolean"
use="optional"/>
        </xsd:extension>
        </xsd:complexContent>
15    </xsd:complexType>

    <xsd:complexType name="BidBehaviorType">
        <xsd:complexContent>
            <xsd:extension base="RequestType">
20          <xsd:sequence>
              <xsd:choice>
                  <xsd:element name="Fixed"
                      type="FixedType" minOccurs='1' maxOccurs='1' />
                  <xsd:element name="B2P"
25                      type="B2PType"  minOccurs='1' maxOccurs='1' />
              </xsd:choice>
            </xsd:sequence>
          </xsd:extension>
        </xsd:complexContent>
30    </xsd:complexType>

    <xsd:complexType name="FixedType">
        <xsd:complexContent>
            <xsd:extension base="RequestType">
35          <xsd:attribute name="bid" type="BidType" use="required"/>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

40    <xsd:simpleType name="BidType">
        <xsd:restriction base="xsd:token">
            <xsd:pattern value="[0-9]+\.[0-9]{0-9}"/>
        </xsd:restriction>
    </xsd:simpleType>
45

    <xsd:complexType name="B2PType">
        <xsd:complexContent>
            <xsd:extension base="RequestType">
                <!-- The requested rank -->
50          <xsd:attribute name="rank" type="xsd:positiveInteger"
use="required"/>
                <!-- How much the advertiser is willing to pay for the rank -
-->
                <xsd:attribute name="maxCap" type="xsd:float"
55 use="required"/>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

```

```

    <xsd:complexType name="RWListingType">
      <xsd:complexContent>
        <xsd:extension base="RequestType">
5          <xsd:attribute name="listingId" type="NonEmptyString"
use="required"/>
          <xsd:attribute name="title" type="NonEmptyString"
use="optional"/>
10          <xsd:attribute name="description" type="NonEmptyString"
use="optional"/>
          <xsd:attribute name="url" type="NonEmptyString"
use="optional"/>
        </xsd:extension>
      </xsd:complexContent>
15    </xsd:complexType>

    <xsd:simpleType name="MarketType">
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="US"/>
20        <xsd:enumeration value="UK"/>
        <xsd:enumeration value="DE"/>
      </xsd:restriction>
    </xsd:simpleType>

25    <xsd:simpleType name="NonEmptyString">
      <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
      </xsd:restriction>
    </xsd:simpleType>
30  </xsd:schema>
```

Appendix D: Exemplary Response Schema

```

5  <?xml version="1.0" encoding="UTF-8"?>
    <!--
    *****
    ** -->
    <!-- Copyright 2001, Overture
    -->
    <!--
    -->
10  <!-- An XML Schema for bidding tools to programmatically access the
    features -->
    <!-- of DTC.
    -->
    <!--
15  *****
    ** -->

    <xsd:schema
20  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">

        <xsd:element name="DTCResponse" type="DTCResponseType"/>

        <xsd:complexType name="ResponseType">
25  <xsd:attribute name="aux" type="NonEmptyString"
    use="optional"/>
        </xsd:complexType>

        <xsd:complexType name="StatusResponseType">
30  <xsd:complexContent>
            <xsd:extension base="ResponseType">
                <xsd:attribute name="success" type="xsd:boolean"
    use="required"/>
                <xsd:attribute name="reason" type="NonEmptyString"
35  use="optional"/>
            </xsd:extension>
        </xsd:complexContent>
        </xsd:complexType>

40  <xsd:complexType name="DTCResponseType">
        <xsd:complexContent>
            <xsd:extension base="StatusResponseType">
                <xsd:sequence>
45  <xsd:element name="ActionsResponse"
                    type="ActionsResponseType"
                    minOccurs='0'
                    maxOccurs='unbounded' />
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
50  </xsd:complexType>

        <xsd:complexType name="ActionsResponseType">
            <xsd:sequence>
55  <xsd:element name="GetAccountIdsResponse"
                    type="GetAccountIdsResponseType"
                    minOccurs='0'

```

```

        maxOccurs='unbounded' />
<xsd:element name="GetMarketStateResponse"
  type="MarketStateResponseType"
  minOccurs='0'
5    maxOccurs='unbounded' />
<xsd:element name="GetListingsResponse"
  type="GetListingResponseType"
  minOccurs='0' maxOccurs='unbounded' />
<xsd:element name="SetListingResponse"
10   type="ListingResponseType"
  minOccurs='0' maxOccurs='unbounded' />
<xsd:element name="AddListingResponse"
  type="ResponseType"
  minOccurs='0' maxOccurs='unbounded' />
15   <xsd:element name="DeleteListingResponse"
    type="ListingResponseType"
    minOccurs='0' maxOccurs='unbounded' />
  </xsd:sequence>
  <xsd:attribute name="accountId" type="NonEmptyString"
20 use="optional" />
</xsd:complexType>

<xsd:complexType name="GetAccountIdsResponseType">
  <xsd:complexContent>
25   <xsd:extension base="StatusResponseType">
    <xsd:sequence>
      <xsd:element name="Account"
        type="AccountType"
        minOccurs='0'
30        maxOccurs='unbounded' />
    </xsd:sequence>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
35

<xsd:complexType name="AccountType">
  <xsd:complexContent>
    <xsd:extension base="ResponseType">
      <xsd:attribute name="id" type="NonEmptyString"
40 use="required" />
      <xsd:attribute name="market" type="MarketType"
        use="required" />
    </xsd:extension>
  </xsd:complexContent>
45 </xsd:complexType>

  <xsd:complexType name="MarketStateResponseType">
    <xsd:complexContent>
      <xsd:extension base="MSListingResponseType">
50        <xsd:attribute name="market" type="MarketType"
          use="required" />
        <xsd:attribute name="searchTerm" type="NonEmptyString"
          use="required" />
      </xsd:extension>
55    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="MSListingResponseType">

```

```

5      <xsd:complexContent>
        <xsd:extension base="StatusResponseType">
          <xsd:sequence>
            <xsd:element name="Listing"
              type="MSListingType"
              minOccurs="0"
              maxOccurs="100"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
10    </xsd:complexType>

    <xsd:complexType name="GetListingResponseType">
      <xsd:complexContent>
15        <xsd:extension base="StatusResponseType">
          <xsd:sequence>
            <xsd:element name="Listing"
              type="GetListingType"
              minOccurs="0"
20              maxOccurs="100"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
25

    <xsd:complexType name="ListingResponseType">
      <xsd:complexContent>
        <xsd:extension base="StatusResponseType">
          <xsd:attribute name="listingId"
30            type="NonEmptyString"
            use="required"/>
          <xsd:attribute name="confirmationNumber"
            type="NonEmptyString"
            use="optional"/>
35        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="RequiredListingType">
40      <xsd:attribute name="title" type="NonEmptyString"
        use="required"/>
      <xsd:attribute name="description" type="NonEmptyString"
        use="required"/>
      <xsd:attribute name="bid" type="BidType"
45      use="required"/>
      <xsd:attribute name="market" type="MarketType"
        use="required"/>
      <xsd:attribute name="searchTerm" type="NonEmptyString"
        use="required"/>
50    </xsd:complexType>

    <xsd:complexType name="MSListingType">
      <xsd:complexContent>
        <xsd:extension base="RequiredListingType">
55          <xsd:attribute name="listingId" type="NonEmptyString"
            use="optional"/>
          <xsd:attribute name="url" type="NonEmptyString"
            use="optional"/>

```

```

        <xsd:attribute name="currency" type="CurrencyType"
use="optional"/>
        <xsd:attribute name="rank" type="xsd:integer"
use="optional"/>
5      </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="GetListingType">
10      <xsd:complexContent>
        <xsd:extension base="RequiredListingType">
            <xsd:attribute name="listingId" type="NonEmptyString"
use="required"/>
            <xsd:attribute name="url" type="NonEmptyString"
15 use="required"/>
            <xsd:attribute name="online" type="xsd:boolean"
use="required"/>
            <xsd:attribute name="currency" type="CurrencyType"
use="optional"/>
20            <xsd:attribute name="rank" type="xsd:integer"
use="optional"/>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

25    <xsd:simpleType name="CurrencyType">
        <xsd:restriction base="NonEmptyString">
            <xsd:enumeration value="USD"/>
            <xsd:enumeration value="GBP"/>
30            <xsd:enumeration value="EUR"/>
        </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="BidType">
35    <xsd:restriction base="xsd:token">
        <xsd:pattern value="[0-9]+\.[0-9][0-9]"/>
    </xsd:restriction>
    </xsd:simpleType>

    <xsd:simpleType name="NonEmptyString">
40    <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
    </xsd:restriction>
    </xsd:simpleType>

45    <xsd:simpleType name="MarketType">
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="US"/>
            <xsd:enumeration value="UK"/>
50            <xsd:enumeration value="DE"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
55

```

What is claimed is:

1. A system for enabling an advertising web site promoter using a computer network to update information relating to a search listing within a search result list generated by an Internet search engine, the system comprising:

a computer system having stored thereon a database having at least one account record for each of a plurality of advertising web site promoters using the computer network, the account record including:

at least one search listing including

a search term having at least one keyword,

a modifiable bid amount,

a Uniform Resource Locator (URL) corresponding to the address of a document residing on a network server,

a description, and

a title;

an account balance;

extensible markup language (XML) programming code on said computer system configured to add a search listing to an account of an advertising web site promoter upon receiving a request from said advertising web site promoter;

programming code configured to process a search request from a remote computer, the search request including at least one keyword, the search request being received over the computer network from the remote computer through a web site that is publicly accessible without authentication; and

programming code configured to generate a search result list in response to the search request, the search result list including search listings from the accounts on the database, wherein the search term for each search listing in the search result list generates a match with the search request, the search listings in the search result list arranged in an order determined using the bid amounts of the search listings.

2. The system of claim 1 further comprising:

XML programming code on said computer system configured to delete a search listing to an account of an advertising web site promoter upon receiving a request from said advertising web site promoter.

3. The system of claim 1 further comprising:

XML programming code on said computer system configured to modify in substantially real time the search listing of an advertising web site promoter upon receiving a request from said advertising web site promoter.

4. The system of claim 1 further comprising:

XML programming code configured to generate an activity report for an advertising web site promoter upon receiving a request from said advertising web site promoter.

5. The system of claim 1 further comprising:

XML programming code configured to provide the advertising web site promoter with login access in response to authentication, wherein the advertising web site promoter's login access grants the advertising web site promoter access to modify the advertising web site promoter's account, the advertising web site promoter not being provided with access to modify the accounts of others.

6. The system of claim 1 further comprising:

XML programming code on said computer system for adding money to the account of an advertising web site promoter in substantially real time upon receiving a request from said advertising web site promoter.

7. The system of claim 6 further comprising:

a history of search listings included in the advertising web site promoter's account record; and

payment processing information, wherein said payment processing information is accessible to the computer system and isolated from public access via the computer network; and

a payment history.

8. The system of claim 1 further comprising:

an XML interface configured to

receive XML requests from the advertising web site promoter; and

communicate XML response to the advertising web site promoter in response to a received XML request.

9. The system of claim 1 further comprising:

an XML schema.

10. A method of enabling a web site promoters using a computer network to update information relating to a search listing within a search result list generated by a search engine in response to a search request received from a remote computer over the computer network, comprising:

storing at least one account record for each of a plurality of web site promoters of the computer network, said at least one account record including an account identifier, and

at least one search listing having a search term and a modifiable bid amount;

providing a web site promoter with authenticated login access, wherein the web site promoter's login access permits the web site promoter to modify the web site promoter's account record;

receiving an extensible markup language (XML) request from said web site promoter;

modifying a search listing of the account record upon receiving the XML request;

receiving information entered by a user through an input device;

searching the stored at least one account record and identifying at least some search listings representing a match with the information entered by the user; and

ordering a search result list of identified search listings in an order corresponding to the bid amounts of the search listings.

10 11. The method of claim 10 further comprising:
receiving the XML request from the web site promoter; and
parsing the XML request against an XML schema to identify the XML request.

15 12. The method of claim 11 further comprising:
providing an XML response to the web site promoter confirming modification of the search listing.

20 13. A method of enabling a web site promoters using a computer network to manage information relating to a search listing within a search result list generated by a search engine in response to a search request received from a remote computer over the computer network, comprising:
storing an account database having at least one account record for each of a plurality of web site promoters of the computer network, said at least one account record including

25 an account identifier, and
at least one search listing having a search term and a modifiable bid amount;

30 storing an extensible markup language (XML) schema;
receiving one or more XML requests from a web site promoter for management of one or more search listings of the web site promoter; and

operating on the at least one account record in response to an XML request.

14. The method of claim 13 further comprising:

providing a market state to the web site promoter in response to the XML
5 request.

15. The method of claim 14 wherein providing the market state
comprises:

formatting an XML response including information about the market state;
10 and
communicating the XML response to the web site promoter.

16. The method of claim 13 wherein operating on the at least one
account record comprises:

15 modifying the modifiable bid amount in response to the XML request.

17. A method for managing search listings of an on-line marketplace
using a client computer in data communication with an account database server
computer of the on-line marketplace, the method comprising:

20 at the client computer, formatting an extensible markup language (XML)
request to set a bid amount of a search listing stored at the account database
server; and

at the client computer, communicating the XML request to the account
management server of the on-line marketplace.

25 18. The method of claim 17 wherein formatting the XML request
comprises:

formatting an XML message with an account identifier associated with the
search listing, a listing identifier corresponding to the search listing and a desired
30 bid behavior.

19. The method of claim 17 further comprising:

receiving an XML response indicating successful completion of the request to set a bid amount.

5 20. A method for managing search listings of an on-line marketplace using a client computer in data communication with an account database server computer of the on-line marketplace, the method comprising:

10 at the client computer, formatting an extensible markup language (XML) request to receive a set of account identifiers corresponding to accounts associated with the advertiser and stored at the account database server; and

communicating the XML request from the client computer to the account database server computer of the on-line marketplace.

15 21. The method of claim 20 wherein formatting the XML request comprises:

formatting an XML message with a username associated with the advertiser and an XML tag requesting the set of account identifiers associated with the username.

20 22. The method of claim 21 further comprising:

at the client computer, receiving an XML response including the account identifiers associated with the advertiser.

25 23. A method for managing search listings of an on-line marketplace using a client computer in data communication with an account database server computer of the on-line marketplace, the method comprising:

at the client computer, formatting an extensible markup language (XML) request to retrieve market state of the on-line marketplace; and

30 communicating the XML request from the client computer to the account database server computer of the on-line marketplace.

24. The method of claim 23 wherein formatting the XML request comprises:

formatting an XML message with a marketplace identifier and a search term.

5

25. The method of claim 24 further comprising:

at the client computer, receiving from the account database server computer an XML response including search listing information for one or more search listings associated with the search term in the marketplace associated with the identifier.

10

26. A method for managing search listings of an on-line marketplace using a client computer in data communication with an account database server computer of the on-line marketplace, the method comprising:

15

formatting an extensible markup language (XML) request to retrieve search listings associated with an advertiser of the on-line marketplace; and

communicating the XML request to an account management server of the on-line marketplace.

20

27. The method of claim 26 wherein formatting the XML request comprises:

formatting an XML message with an account identifier associated with the advertiser.

25

28. The method of claim 26 wherein formatting the XML request comprises:

at the client computer, formatting an XML message with an account identifier associated with the advertiser and one or more of

30

a search term,

a specified bid amount,

a uniform resource locator,

a title, and
a description.

29. A bid management tool for managing search listings stored on an
account management server of an on-line marketplace from a remote client
computer, the bid management tool comprising:

first computer readable program code and data operating on the remote
client computer configured for data communication with the account management
server, the first computer readable program code and data forming a menu system;
and

second computer readable program code and data operating on the remote
client computer, the second computer readable program code and data forming a
search listing management function cooperating with the menu system to manage
one or more search listings according to user requirements specified through the
menu system.

30. The bid management tool of claim 29 further comprising:
third computer readable program code and data forming a setup function
configured to receive user entries specifying advertiser identifiers and account
identifiers for management.

31. The bid management tool of claim 29 further comprising:
fourth computer readable program code and data forming a reporting function.

32. A client computer operable in conjunction with an account
management server of an on-line marketplace, the account management server
storing search listings associated with advertisers; the client computer comprising:
a bid management tool; and
an extensible markup language (XML) interface configured to
communicate XML requests to the account management server.

33. The client computer of claim 32 wherein the bid management tool comprises:

a menu system; and

a search listing management function cooperating with the menu system to manage one or more search listings on the account management server according to user requirements specified through the menu system.

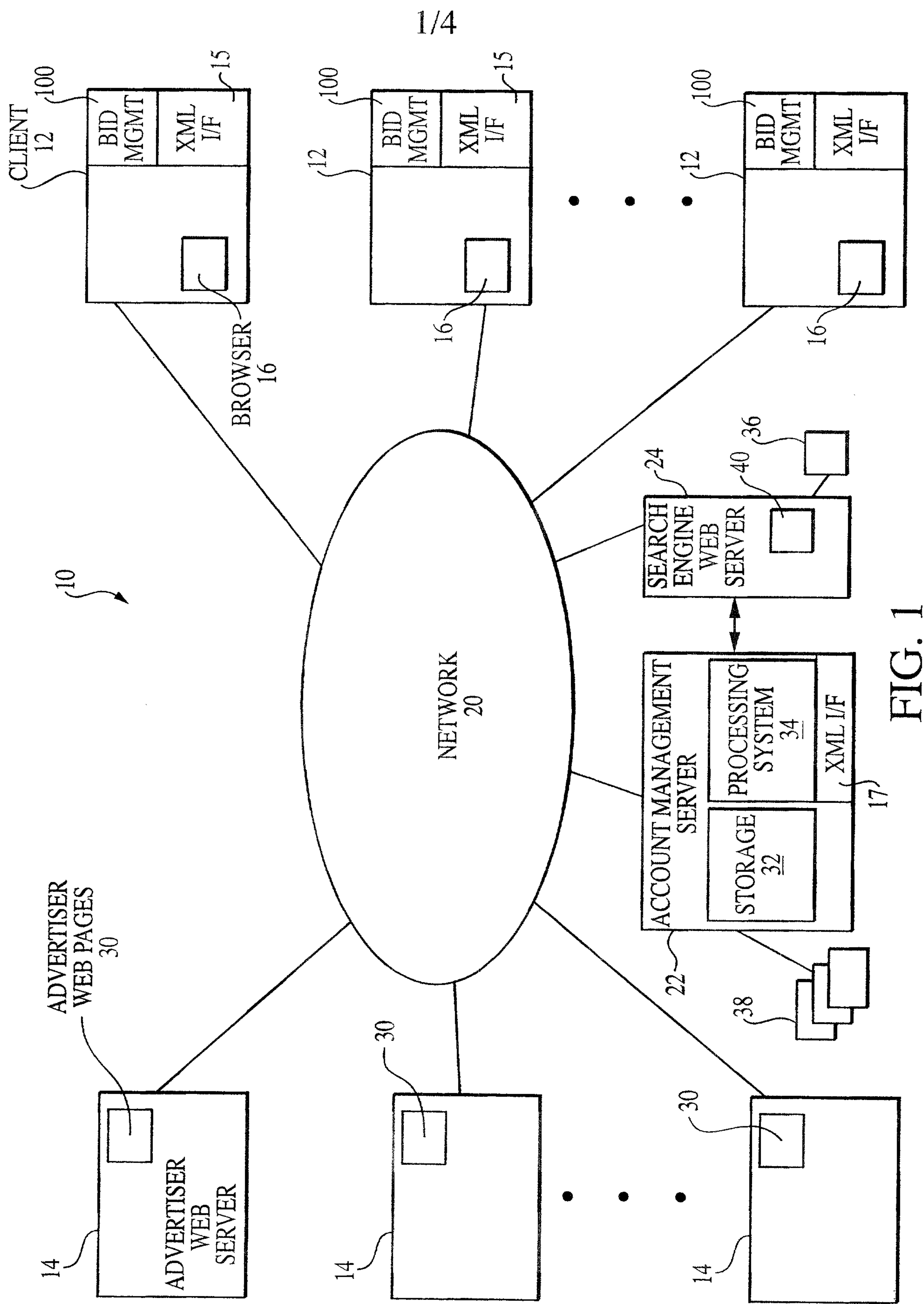
34. A computer readable storage medium comprising:

first program code implementing a bid management tool for managing search listings on an account management server of an on-line marketplace; and

second program code implementing an extensible markup language (XML) interface for communicating with a complementary XML interface of the on-line marketplace.

35. A computer readable storage medium comprising:

program code implementing an extensible markup language (XML) schema for reference when communicating XML requests from a client computer operable in an on-line marketplace to an account management server of the-line marketplace.



2/4

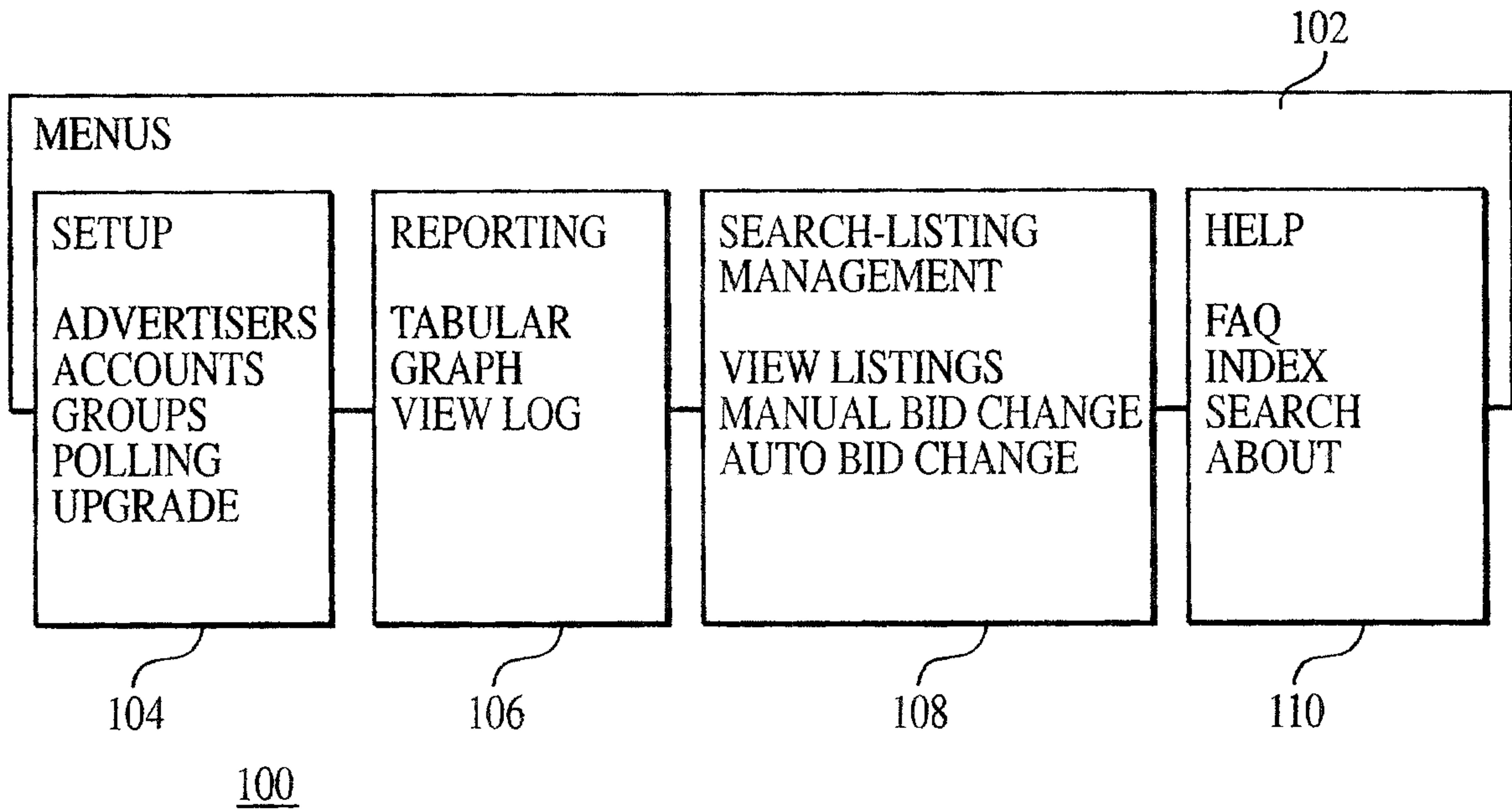


FIG. 2

3/4

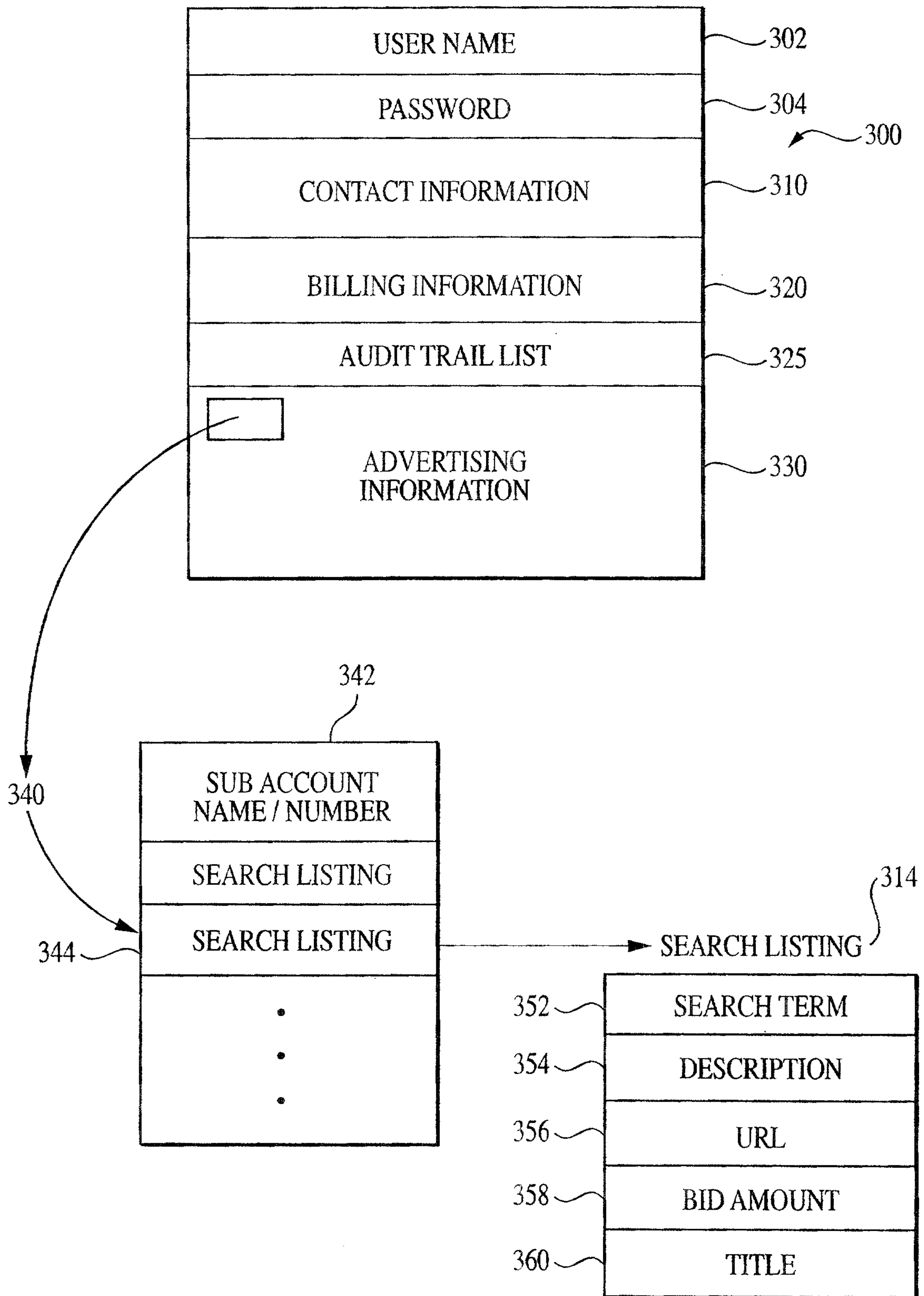
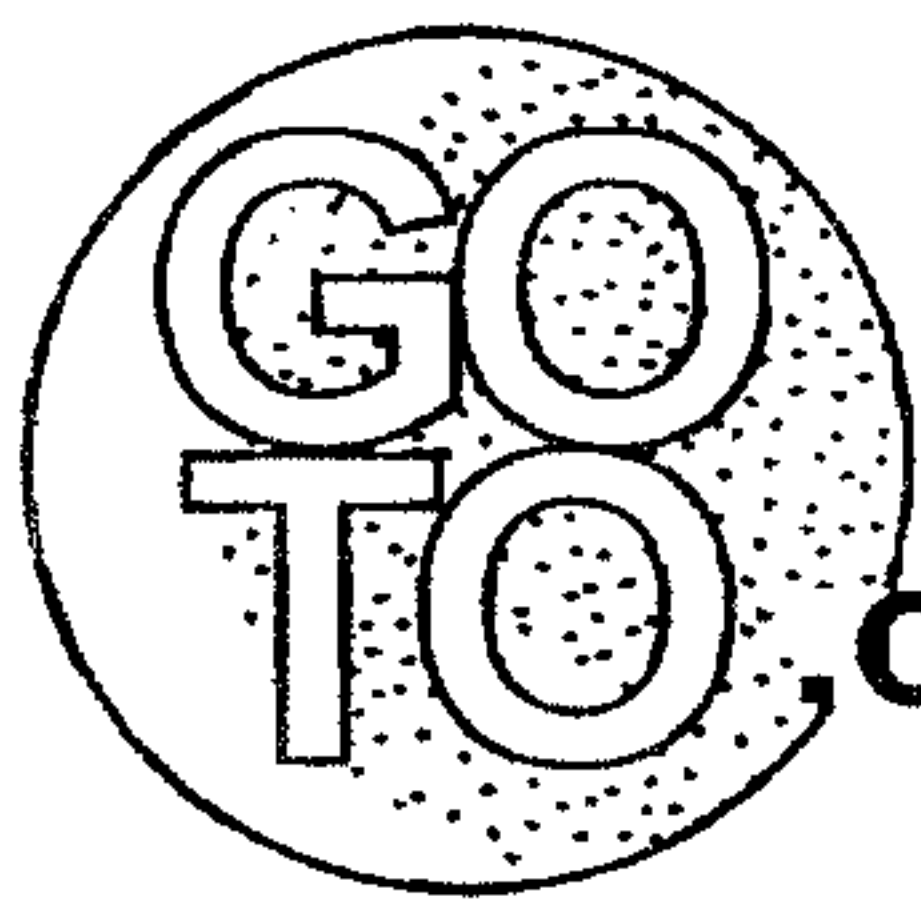


FIG. 3

4/4


 Find it on *at hand*

ZIP DRIVES

Find It!

Type what you're looking for and click Find It!

760a

[Try Go To Remote!](#)

730

[Add Go To to Favorites](#)1. [We Buy And Sell Zip Drives](#) 720

720

Buy Sell or Trade used computers, memory, peripherals, hardware and software. We offer the best brand names, competitive prices, large inventory, and reliability. We can ship anywhere.

<http://www.micro-tradingpost.com/> (Cost to advertiser: \$0.08) 750a

2. [MacinFind.com - Click Here!](#)

760b

Online guide to apple computer retailers. Find Imacs, power macintosh G3s, powerbooks, mac software and much more!

<http://www.macinfind.com/> (Cost to advertiser: \$0.07) 750b

3. [Custom Computer Systems, Inc \(CCS\)](#)

760c

CCS, a wholesale distributor of computer systems, parts & accessories. Online shopping and same day shipping! 800.379.1227

<http://www.wvccsi.com/> (Cost to advertiser: \$0.06) 750c

4. [Best Price on Zip Drives](#)

760d

ATMAN - USA.COM Computer on-line shopping. Huge selection, competitive price, reliable service. Major credit cards are welcome.

<http://www.atman-usa.com/> (Cost to advertiser: \$0.02) 750d

5. [Computer Beat...One - Step Web Directory](#)

760e

A One-Stop Web Directory featuring top computer, PC, Mac web sites on the Internet. Spotlights Computer News, Help Desks, HTML, Internet and Computer Shopping!

<http://www.search-beat.com/> (Cost to advertiser: \$0.01) 750e

6. [Macnet Online Computer Store](#)

760f

Computer retailer of hardware software peripherals consumerable and accessories for PC/Macintosh users. See Powermac, Imac, DVD, digital camera, monitors, printers, scanners, networking, PDA, notebook, and more.

<http://www.applemacnet.com/> (Cost to advertiser: \$0.01) 750f

7. [hard drives for sale, western digital, maxtor, seagate, samsung, ibm, iomega zip drives. Ide, scsi, Psa Inc reseller](#)

760g

Offering computer systems, memory, cpu sales, motherboards, computer upgrades at volume and dealer pricing. A distributor, reseller, broker for all pc needs from corporations to individual personal computer users. PSA Inc. - computer sales and upgrades.

<http://www.psai.com/>

8. [Tape Drives, Zip Drives, Floppy Drives](#)

760h

Wholesale distributor in Cleveland Ohio of computer hardware, computer systems, computer components, peripherals, cases, keyboards, mice, tape backup and floppy drives. Free technical support.

<http://www.bitsandbytesinc.com/>

[http://www.../\\$sessionId\\$MEPOIJQAA4BYLQFIEE1APUQ?type=home&Keywords=zip+drive](http://www.../$sessionId$MEPOIJQAA4BYLQFIEE1APUQ?type=home&Keywords=zip+drive)

FIG. 4

