UNIVERSAL CONSUMPTION SERVICE

Client System

Client Application(s) 104

Display 106

Client System 102-1

Network 110

Purchase Request 114

Vendor Confirmation Request 116

Product Add Request 118

Vendor(s) 117

Vendor Confirmation Request 116

Product Add Request 118

Vendor(s) 117

Vendor Database 150

Vendor Profiles 152

User Profile Database 190

Vendor Selection Module 132

Purchasing Module 128

Vendor Scoring Module 128

Request Processing Module 124

Communication Module 122

Product Database 140

Server System 120

Client-Server Environment 100

Client System 102-2

Related U.S. Application Data


Publication Classification

Int. Cl. G06Q 30/06 (2012.01)

U.S. Cl.

CPC ................................................. G06Q 30/0635 (2013.01)

USPC ................................................... 705/26.81

ABSTRACT

In accordance with some implementations, a method for selecting a vendor in accordance with some implementations is disclosed. The method is performed on a server system having one or more processors and memory storing one or more programs for execution by the one or more processors. The server system detects stores one or more vendor profiles, wherein each vendor profile is associated with a particular vendor and includes one or more vendor category scores. The server system then receives a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased. The server system then determines a vendor to supply the requested product based on the stored vendor profiles. The server system purchases the requested product from the determined vendor. The server system then arranges for delivery of the purchased product or service.
Client Device 102

CPU 202

Network Communication Module

Operating System

Display Module

Client Application Module(s)

Browser Application

Product Request Module

Product Add Request Module

...

Client Data

User Message Data

User Profile Data

...

FIGURE 2
Create vendor profiles

Determine one or more vendor scoring categories

Generate a vendor category score for one or more determined vendor scoring categories based on past vendor performance

Past vendor performance is based on user supplied feedback

Past vendor performance is determined based on data directly measured by the computer system

Data measured by the computer system includes product delivery time, product price, and the number of products in stock

Store one or more vendor profiles, wherein each vendor profile is associated with a particular vendor and includes one or more vendor category scores

Receive a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased

Determine a vendor to supply the requested product based on the stored vendor profiles

Generate an overall vendor score based on the one or more vendor category scores wherein the one or more vendor category scores are weighted based on the priority associated with each category for each vendor in the plurality of vendors

Rank the plurality of vendors based on the generated overall vendor score

FIGURE 6
Store user profiles for each user of the computer system; wherein user profiles include priorities associated with vendor categories

Prior to purchasing the requested product, send a request to the user to confirm the determined vendor

Receive a response from the user confirming the determined vendor

Purchase the requested product from the determined vendor

Arrange for delivery of the purchased product or service

Receive feedback from the user regarding the determined vendor

Update the vendor profiles based on the received feedback

The server system includes a product database

Receive a request to add a product to the product database, wherein the request includes a vendor associated with the product

FIGURE 7
UNIVERSAL CONSUMPTION SERVICE

RELATED APPLICATIONS


TECHNICAL FIELD

[0003] The disclosed implementations relate to the field of online commerce generally and to enhancing optimizing commerce experiences for users in particular.

BACKGROUND

[0004] Over the last two decades, the buying and the selling of products through computer networks such as the Internet has increased dramatically. A significant portion of all commerce is now conducted online through the Internet. As the amount of commerce conducted online grows, the number of online commerce vendors also grows. As such, an increasing number of online vendors compete with each other to offer users the best user experience. An important way to differentiate from other online retailers is to provide the most convenient user experience.

[0005] One result of the growth of online commerce is that the sheer number of vendors and websites can be overwhelming to users who want to purchase a product or service. It can be difficult for users to process so much information to find the best product or service from the best vendor for their particular needs. Indeed, often users never find out about opportunities or vendors that would closely match their needs.

[0006] Personal assistants, wedding planners, travel agents, and other consumer agents offer a contractual relationship with the end consumer to help them fulfill their consumer needs for products and services, brokering the deal with the ultimate product or service providers. For example, a wedding planner might arrange for a service or a service provider (e.g., reserving and contracting a band). Similar to these services, our service acts as an Agent on behalf of the consumer to broker a “consumptive experience for a product/service combination (broadly defined) that they pay for (most of the time). It would be useful to have a service that would curate the large number of products and services available and the various vendors to provide the best experience to a user in a way similar to the way that personal assistants work or a personal shopper.

SUMMARY

[0007] In some implementations, the system described herein, bridges the gap between “social” and “commerce” by enabling a universal ability to consume, irrespective of merchant or product or service, and seamlessly making the consumption easy. This allows for the free expression of social intent, while enabling commerce to occur where it is already most strongly encouraged: from the people we trust.

[0008] In some implementations, the system brokers consumptive experiences on behalf of end consumers, for combinations of products and services, where the consumer registers the intent to consume (and typically transacts) with the service, authorizing the service to act as an agent on their behalf to find and coordinate the fulfillment of the consumptive experience from various service and product providers, merchants, retailers, distributors, dealers, manufacturers, or any other means or method of consumptive fulfillment. Products are defined as consumer objects that can be used or consumed by the purchaser and includes both physical products (e.g., a chair) and digital products (an application, a piece of digital media etc.). Services can include any purchasable service (e.g., a restaurant reservation or a house painter painting a normal sized house or any other service).

[0009] In some implementations, the service acts like an agent/broker/concierge/personal shopper on behalf of the member, who can express their intent to consume a product or service, can provide a payment method, and the agent thereafter ensures that the product and/or service seamlessly is delivered for the individual. In some implementations, the service has three main areas of systems and methods. The first involves analyzing, capturing, and representing the product/service intended for consumption. The second involves the management of the payment, processing, merchant/vendor relationships, and fulfillment of the product/service. The third covers the optimization and selection of merchants/vendors/providers to optimally obtain the best product for the user.

[0010] In accordance with some implementations, a method for curating online content to enhance a user’s experience is disclosed. The computer system described within customizes and curates both products and services and the vendors that provide those products and services. In some implementations, the system provides recommendations to a user based on, among other things, the user’s preferences, interests, social network relationships, trusts and overall product popularity. In addition to providing product recom-
mendations, once a user has instructed the computer system to purchase a particular product or service, the computer system assists the user in determining the optimal vendor from which to purchase the indicated product or service.

[0011] In some implementations, the product purchase request from a user includes a vendor preference. In some implementations, the purchase request includes a user specified vendor with instructions to use only the specified vendor. In other implementations, the purchase request includes a user specified vendor, but no instructions to use only the specified vendor. In some implementations, the level of preference is included in the product request and the stronger the level of the request, the more likely that the computer system will use the specified vendor. In some implementations, the purchase request includes either no vendor or no stated preference. In this case, the computer system selects a vendor from the plurality of possible vendors.

[0012] In some implementations, the computer system stores a database of vendor profiles. Each vendor profile includes a plurality of vendor category scores. Each vendor category score rates a vendor on a particular aspect or attribute associated with providing products to users. For example, vendor score categories include such categories as average cost, delivery time, delivery cost, consumer satisfaction, distance from the user, shipping methods, stock status, return policy, and any other category relevant to providing products or services to consumers. For each product order received, the computer system uses the plurality of vendor category scores to generate an overall weighted vendor score for each vendor for each product order. In some implementations, the various vendor category scores are weighted in accordance with priorities associated with each category score. Once an overall weighted score is generated for each vendor, the vendors are ranked in accordance with their respective overall weighted vendor score.

[0013] In some implementations, the computer system determines a vendor from which to purchase the requested product or service. In some implementations, the vendor with the highest overall weighted vendor score is selected. In some implementations, each product is stored in a product database and has one or more associated vendors. The computer system selects the associated vendor stored in the product database. In some implementations, the user specifies a vendor when the product purchase request and the computer system selects the user specified vendor.

[0014] In some implementations, the computer system generates an overall vendor score for both the user submitted vendor and the plurality of vendors stored by the computer system. The computer system then compares the overall vendor score of the user submitted vendor and the highest scored vendor in the plurality of vendors. If the overall vendor score for the highest scored vendor of the plurality of vendors exceeds the overall vendor score of the user submitted vendor by at least a predetermined amount, the computer system selects the highest scored vendor instead of the user selected vendor. In some implementations, if the computer system selects a vendor other than the user submitted vendor, the system sends a confirmation request to the user prior to purchasing the requested product from the new vendor. If the user responds and rejects the new vendor, the computer system reverts to the user selected vendor.

[0015] In some implementations, the priorities associated with the vendor category scores are determined by the user and submitted with the associated product purchase request. In some implementations, the computer system has default values associated with each vendor category and the computer uses the default priority vendor scores when generating the overall vendor scores. In some implementations, priorities are determined based on past user behavior that has been recorded. The determined priorities are then stored in the user profile. In some implementations, the computer system purchases the product from the selected vendor and arranges for delivery of the product according to the user’s instruction.

[0016] In accordance with some implementations, a method for selecting a vendor is disclosed. The method is performed on a server system having one or more processors and memory storing one or more programs for execution by the one or more processors. The server system stores one or more vendor profiles, wherein each vendor profile is associated with a particular vendor and includes one or more vendor category scores. The server system then receives a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased. The server system then determines a vendor to supply the requested product based on the stored vendor profiles. The server system purchases the requested product from the determined vendor. The server system then arranges for delivery of the purchased product or service.

[0017] In accordance with some implementations, a server system for selecting a vendor is disclosed. The server system has one or more processors, and memory storing one or more programs to be executed by the one or more processors. The one or more programs include instructions for selecting a vendor and includes one or more vendor category scores. The one or more programs further include instructions for receiving a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased. The one or more programs further include instructions for determining a vendor to supply the requested product based on the stored vendor profiles. The one or more programs further include instructions for purchasing the requested product from the determined vendor. The one or more programs further include instructions for arranging for delivery of the purchased product or service.

[0018] In accordance with some implementations, a non-transitory computer readable storage medium storing one or more programs configured for execution by a server system is disclosed. The one or more programs also include instructions for storing one or more vendor profiles, wherein each vendor profile is associated with a particular vendor and includes one or more vendor category scores. The one or more programs further include instructions for receiving a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased. The one or more programs further include instructions for determining a vendor to supply the requested product based on the stored vendor profiles. The one or more programs further include instructions for purchasing the requested product from the determined vendor. The one or more programs further include instructions for arranging for delivery of the purchased product or service.

BRIEF DESCRIPTION OF THE DRAWINGS

[0019] FIG. 1 is a block diagram illustrating a client-server environment in accordance with some implementations.

[0020] FIG. 2 is a block diagram illustrating a client system in accordance with some implementations.
FIG. 3 is a block diagram illustrating a server system in accordance with some implementations.

FIG. 4 depicts a block diagram of an exemplary data structure for a user profile database for storing information related to users of the server system in accordance with some implementations.

FIG. 5 depicts a block diagram of an exemplary data structure for a vendor profile database for storing information related to vendors of the server system in accordance with some implementations.

FIG. 6 is a flow diagram illustrating the process for selecting a vendor in accordance with some implementations.

FIG. 7 is a flow diagram illustrating the process for selecting a vendor in accordance with some implementations.

Like reference numerals refer to corresponding parts throughout the drawings.

DESCRIPTION OF IMPLEMENTATIONS

In some implementations, a computer system operates a commerce system for allowing users to easily and quickly find and purchase goods and services that meet their specific needs and desires. In some implementations, the commerce system is available online via a web site, mobile electronic device, etc. In some implementations, the commerce system operates by the computer system uses user profiles and user profiles to select the most appropriate vendor to provide the product or service requested by the user at the time of the transmission. The computer system then purchases the desired product or service from the selected vendor and arranges for delivery of the product or service to the user or another person indicated by the user (e.g., as a gift).

FIG. 1 is a block diagram illustrating a client-server environment 100 in accordance with some implementations. The client-server environment 100 includes one or more client systems 102-1 and 102-2, a server system 120, and one or more vendors 170, all connected over a network 110. In some implementations, the client system 102 includes one or more client applications 104 and a display 106. The server system 120 includes a communication module 122, a request processing module 124, a vendor scoring module 126, a purchasing module 128, a vendor selection module 130, a product database 140, a vendor database 150, and a user profile database 160. The network 110 may consist of any one or more of any of a variety of networks, including local area networks (LAN), wide area networks (WAN), wireless networks, wired networks, the Internet, or any combination of such networks.

In accordance with some implementations, the client system 102 includes one or more client applications 104. The one or more client applications 104 include, but are not limited to, a web browsing application for connecting to the server system 120. The client system 102 also includes a display 106. In some implementations, the client system is a computing device with the display integrated directly into the device itself, such as a laptop, a smart phone, a personal computer, a tablet computer, or another computing device. In other implementations, the display is connected to, but not integrated into, the client system. For example, desktop computer systems often do not have an integrated display and instead connect to a standalone display.

In some implementations, the client system 102 transmits a purchase request 114 to a server system 120. In some implementations, the purchase request 114 identifies a particular product or service that the user requests the server system to purchase on his or her behalf. In some implementations, the purchase request 114 includes a specific vendor associated with the product or service requested. In other implementations, the purchase request includes a vendor preferred by the user, which is able to be changed by the server system 120. In some implementations, the purchase request 114 specifies a type or category of product, but not a specific product. In this implementation, the server system 120 uses the included type or category information to select a product or service as well as a vendor to satisfy the user request 114.

In some implementations, the purchase request 114 does not specify a particular vendor, but includes a list of vendor attributes with associated priority information. The list of vendor attributes with the associated priority information is then used by the server system 120 to select an appropriate vendor 170 to supply the requested product or service. For example, if the purchase request 114 indicates that low price is the highest priority, the server system 120 then chooses the vendor that offers the lowest price on the requested product or service. In another example, if the purchase request 114 indicates that fast shipping is the highest priority, then the server system chooses the vendor that will deliver the product or service in the shortest amount of time (e.g., the product must be delivered in time for a special occasion, like Mother’s Day).

In some implementations, the client system 102 sends a product add request 118 to the server system 120. In some implementations, the product add request 118 includes a product to be added to the product database 140 of the server system 120. In some implementations, the product add request 118 specifies a product that is not already stored in the product database 140. In this implementation, the server system adds the product included in the product add request 118 to the product database 140. In other implementations, the product database 140 already includes the product specified in the product add request 118. In this implementation, the server system 120 updates the product database 140 with information in the product add request 118. For example, if the product add request identifies a vendor for the product that the server system 120 did not have associated with the product, the server system 120 edits the relevant product profile to include the new potential vendor. In some implementations, the server system 120 verifies the information included in the product add request with a third party prior to editing the information stored in the product database.

In some implementations, the product add request 118 includes information describing the characteristics of the included product. Information describing the characteristics of a product includes, but is not limited to, product specifications, price, size, color, one or more vendors that stock the item, time availability, and product stock amounts. In some implementations, the server system 120 adds a new entry in the product database 140 for the product included in the product add request 118 (e.g., when no entry for the product already exists). In this case, the information from the user describing the characteristics of the product is used to fill out the new entry in the products database 140. Once the entry is added to the product database 140, the server system 120 uses the newly added product to show and/or recommend to other users of the server system 120.

In some implementations, the product database 140 already includes the product described in the product add request. In this case, the server system 120 updates the product entry in the product database 140 to include any new...
information included in the product add request 118. For example, the product database 140 will be updated to include a new vendor associated with the product, to change the price of the product based on the price included in the product add request 118, or to add a new product characteristic to the entry in the product database 140. In some implementations, the product add request 118 is received as part of a user recommendation and the product database is updated to include the user who submitted the product add request 118 as a recommending user. In some implementations, the server system 120 verifies the information included in the product add request 118 with a third party prior to adding or updating the new information to the product database.

[0035] In some implementations, the client system receives a vendor confirmation request 116 from the server system 120. In some implementations, the vendor confirmation request 116 lists a purchase request 114 the user has made and a vendor the server system 120 has selected to complete the purchase, and requests that the user confirm that the selected vendor 170 is acceptable to the user. In some implementations, the user confirms the selected vendor 170 is acceptable to the user of the client system 102 and the client system 102 sends a confirmation to the server system 120. In other implementations, the user rejects the selected vendor 170 and a rejection is sent by the client system 102 to the server system 120.

[0036] In some implementations, when the client system 102 sends a rejection of the selected vendor 170 to the server system 120, the rejection also includes a vendor 170 selected by the user of the client system 102. For example, when the user rejects vendor A, the user also sends a request for vendor B. In other implementations, the rejection instructs the server system 120 to select a different vendor. In some implementations, the rejection includes a list of vendor attributes that each have an associated priority. The server system 120 then is able to select the vendor that most closely matches the priorities set out in the rejection. For example, the rejection states that customer review ratings above 4 stars are the highest priority and that price is the second highest priority. In this case, the server system 120 screens out the vendors with average customer review rating below 4 stars and then selects the vendor with the lowest price among all the vendors 170 with customer service review scores that average above 4 stars.

[0037] In accordance with some implementations, the server system 120 includes a communication module 122, a request processing module 124, a vendor scoring module 126, a purchasing module 128, a vendor selection module 130, a product database 140, a vendor database 150, and a user profile database 160. The communication module is configured to send and receive communications over the network 110 to one or more client systems 102 and one or more vendors 170.

[0038] In accordance with some implementations, the communication module 122 handles all communication with the one or more client systems 102. In some implementations, the communication module 122 receives a purchase request 114 from a client system 102. The communication module 122 sends the purchase request 114 to the request processing module 124. In some implementations, the communication module 122 sends an acknowledgement back to the client system 102, indicating receipt of the purchase request 114. In some implementations, the communication module 122 receives a product add request 118 from a client system 102. In some implementations, the communication module 122 transmits the received product add request 118 to the product database 140 directly. In other implementations, the communication module transmits the received product add request 118 to the request processing module 124.

[0039] In some implementations, the communication module 122 receives a vendor confirmation request 116 from the vendor selection module 130. In some implementations, the communication module 122 transmits the vendor confirmation request 116 to the client system 102. In some implementations, the communication module 122 is able to use one or more of several different communication methods to transmit the recommendation 116 to the client system 102. The server system 120 stores contact information for each user in the user profile database 160.

[0040] In some implementations, the communication module 122 determines the communication method for communicating with a user based on past communications with the user of the client system 102. Thus, the communication module 122 chooses the communication method that a particular user of the client system 102 uses most often when communicating with the server system. For example, if a user communicates with the server system 120 through email in 75% of communications, the server system 120 will use an email messaging system to deliver a vendor confirmation request 116 to the user. In some implementations, the communication module 122 uses a particular user’s most recent communication method to communicate with that particular user. In some implementations, each user selects one or more preferred communication methods and this preference is stored in the user profile database 160. For example, a user selects text messaging as the preferred method for receiving messages and in response the communication module 122 will then use text messages to send vendor confirmation requests 116 to the user, unless directed otherwise. In some implementations, the communication module 122 will use multiple communication methods to send vendor confirmation requests 116 to the user.

[0041] In some implementations, the communication module 122 chooses the communication method based on the time and date that the vendor confirmation request 116 is to be sent. For example, the communication module 122 uses email messages to deliver vendor confirmation requests 116 during the standard workday (9 am-5 pm Monday through Friday) and uses text messages to deliver vendor confirmation requests 116 during evening and on weekends. In some implementations, no vendor confirmation requests 116 will be sent during typical sleeping hours. In some implementations, each user selects time and data preferences for the communication module 122 to use and these preferences are stored in the user profile database 150. The communication module then follows the user selected preferences when communicating with the client system 102 associated with the user.

[0042] In some implementations, the communication module 122 receives a response message from a client device either confirming the selected vendor or rejecting the selected vendor. In either case, the response message is transmitted to the vendor selection module 130. In some implementations, the communication module 122 sends a vendor confirmation request 116 that includes alternative vendors to the vendor selected by the server system 120. In this case, the response message received from the client system 102 included user’s
selection of one of the alternative vendors. If so, the server system 120 uses the selected alternative vendor. In some implementations, the response message from the client system 102 also confirms or rejects other aspects of the purchase request 114. For example, the user rejects or accepts the shipping terms, the specific product SKU selected, or the shipping address.

In accordance with some implementations, the request processing module 124 is configured to receive requests from a plurality of client systems 102 via the communication module 122. In some implementations, the client system 102 receives a purchase request 114 from a client system 102. In some implementations, the request processing module 124 determines whether the requested product is stored in the product database 140. If the requested product is already stored in the product database, the request processing module 124 requests the user profile of the user associated with the purchase request from the user profile database 160. Once the user profile is received by the request processing module, the request processing module 124 transmits the purchase request 114 and the retrieved user profile to the vendor selection module 130. In some implementations, the product requested in the purchase request 114 is not already stored in the product database 140. In this case, the request processing module 124 determines whether the purchase request 114 also includes a product add request 118. In some implementations, the purchase request 114 includes a product add request 118 and the request processing module 124 transmits the data necessary to create a product entry in the product database 140 to the product database 140. In other implementations, the purchase request 114 does not include a product add request 118. In this case, the request processing module 124 requests information for the requested product from a plurality of vendors 170 through the network 110. In some implementations, the product database 140 requests at least the basic information necessary to recommend the product to a user (e.g., name of the produce, price, shipping terms and price, and at least a basic description of the items attributes.) By contacting various vendors 170 or third party sites (e.g., catalog aggregation sites or product information database sites) for product information, the request processing module 124 is able to fill out a product entry for the new product in the product database 140 and gather a list of vendors that stock the product.

In some implementations, the request processing module 124 receives a product add request 118 from a client system 102. In some implementations, in response to receiving a product add request 118, the request processing module 124 determines, for each respective product of the one or more products included in the product add request 118, whether the respective product included in the product add request 118 is already included in the product database 140. If the respective product is included in the product database, the request processing module 124 determines whether any of the information included in the product add request 118 differs from the corresponding product entry in the product database 140. The request processing module 124 updates the information stored in the product database 140 to include any differences. For example, if the product price for a product in the product add request 118 is lower than the price indicated in the product entry in the product database 140, the product database 140 is updated to reflect the lower price. In some implementations, each product included in a product add request 118 has an associated vendor 170. Correspondingly, each product entry in the product database includes a list of vendors that stock the product. In response to receiving a product add request 118, the product entry for each product included in the product add request 118 is updated to include a vendor associated with the product in the product add request 118 in the list of vendors.

In some implementations, the request processing module 124 transmits the received purchase request 114 to a vendor selection module 130 to determine the optimal vendor to supply the requested product or service. In some implementations, the vendor selection module 130 determines a list of all potential vendors 170 that are able to supply the requested product or service to the user. In some implementations, the vendor selection module 130 selects the list of potential vendors to the vendor scoring module 126 for scoring.

In some implementations, the purchase request 114 includes a list of vendor category priorities (e.g., to be used to weight the various vendor category scores). In this implementation, the vendor scoring module 126 sends the set of vendor category priorities to the vendor scoring module 126.

In some implementations, the vendor scoring module 126 generates scores for one or more vendors 170. In some implementations, the vendor scoring module 126 collects data about vendors. In some implementations, this data is gathered from the vendors 170 themselves. For example, the vendor scoring module 126 requests the location of the vendor, the country of origin, the prices of the products supplied by the vendor 170, shipping policies, shipping options and prices, inventory status, and return policies from the vendors directly. In some implementations, the vendor scoring module 126 gathers data from users. For example, users can submit feedback to the server system 120 rating the customer service quality of the vendor 170, the shipping speed, and the return policies of the vendor 170.

In some implementations, vendor scoring module 126 uses the gathered information to generate a plurality of vendor category scores. Each respective vendor category score is associated with a particular attribute of a respective vendor 170. In some implementations, each vendor category score is a number that rates a particular attribute of a vendor on a numerical scale. For example, a particular vendor category score is customer service quality and each vendor category score is a number between 1 and 10, with 1 being the worst and 10 being the best. In some implementations, the purchase request 114 includes a set of priorities from the user that are used to weight the various vendor category scores and determine an overall vendor score for each vendor. In some implementations, the purchase request 114 does not include a set of priorities. In this case, the vendor scoring module 126 uses a predetermined set of default priorities. In some implementations, the server system 120 defaults to weighing each vendor attribute or category equally. In some implementations, the server system 120 defaults to prioritizing price as the highest priority category or factor.

In some implementations, the vendor scoring module 126 accesses the user profile database 160 to determine the set of category priorities for the user that submitted the purchase request 114. In this case the priorities are based on, among other priorities, past user purchase requests, the user’s budget, the user’s location, the priorities of users that have been trusted by the user, and other relevant information stored in the user profile.
In some implementations, the vendor scoring module 126 receives a request from the vendor selection module 130 to generate overall vendor scores for a plurality of vendors 170. In some implementations, the vendor scoring module 126 receives priorities associated with the one or more vendor categories from the vendor selection module 130. In some implementations, the vendor scoring module 126 generates overall vendor scores by using the received category priorities to weight the vendor category scores. For example, vendor profile 152 has three category scores: shipping time, price, and customer service. Vendor A has a score of 8 for shipping time, a score of 3 for price, and a score of 10 for customer service. User A has priorities set such that customer service is twice as important as price, and shipping time is not important at all. Thus, weighing the score would be (shipping score*shipping weight+price score*price weight+customer service score*customer service weight)/total number of factors and would result in (0*8+1*3+2*10)/3 = 7.66.

In some implementations, once the overall vendor scores have been calculated for the plurality of vendors, the vendor scoring module 126 sorts the plurality of vendors into a list that is sorted from highest overall score to lowest overall score. Once the vendor scoring module 126 sorts the plurality of vendors into a list, the vendor scoring module 126 sends the sorted list to the vendor selection module 130. In some implementations, the vendor scoring module 126 also sends calculated scores to the vendor selection module 130.

In some implementations, the vendor selection module 130 receives a sorted list of vendors from the vendor scoring module 126. In some implementations, the vendor selection module 130 selects the vendor that most closely matches the user’s needs. In some implementations, the user has specified a vendor 170. In some implementations, the vendor selection module 130 always uses the user specified vendor when a vendor has been specified by the user. In some implementations, the user has indicated that only the specified vendor may be considered by the server system 120. In other implementations, the user has indicated that server system 120 can consider alternatives to the specified vendor and the vendor selection module 130 compares the overall score of the vendor specified by the user and the vendor with the best overall vendor score per the ranked list of vendors provided by the vendor scoring module 126. If the vendor specified by the user has an overall score that is lower than the top scored vendor by at least a predetermined score, the vendor selection module 130 will select the top scored vendor instead of the vendor specified by the user. For example, the user has indicated a preference for vendor A, and vendor A receives an overall score of 5.5 (out of ten) and vendor B receives an 8.6, which is the highest score from the vendor scoring module 126. The difference is 3.1. If the vendor selection module 130 has a predetermined maximum difference value of 2, the vendor selection module 130 will select vendor B, despite the user’s indicated preference. In some implementations, the maximum difference value will vary depending on the strength of the preference that the user has for the indicated vendor. If the user has indicated a very strong preference, a larger difference in score will be necessary to justify switching to a new vendor. If the user has only a weak preference, a small difference in score will be sufficient to justify a switch. In some implementations, the user preference will be so strongly preferred that the vendor selection module 130 will not change vendors regardless of other vendor scores.

In some implementations, if the vendor selection module 130 selects a vendor that is different from the user specified vendor, the vendor selection module 130 sends a vendor confirmation request 116 to the client system 102. The vendor confirmation request 116 will identify the selected vendor, provide an explanation why the vendor was selected (e.g., this vendor provides a lower price, with the same shipping time), and requests that the user of the client system 102 confirm the choice of new vendor. In some cases, the client system 102 responds and confirms the vendor choice. In other cases, the client system 102 responds and specifies that the user has rejected the selected vendor. In this case, the vendor selection module 130 will select another vendor. In some implementations, the vendor selection module 130 reverts to the user specified vendor.

In some implementations, the user has not specified a preference for a particular vendor, but the vendor selection module 130 determines one or more vendors favored by the user based on the user’s past purchasing history. If a user has given positive feedback for a vendor in the past, the vendor selection module 130 will give that vendor an advantage in future consideration. For example, if Vendor A and Vendor B both score a 7.5 overall vendor score for a particular purchase request 114, and the user associated with the purchase request 114 has previously given Vendor B a positive review, the vendor selection module 130 will modify the overall score by a specific amount to reflect this positive experience. The exact amount of benefit given to the vendor will depend on the number of times the user has given positive feedback for a vendor and how positive the feedback was. Thus, as a user consistently gives positive feedback to a specific vendor over time, the amount that the vendor scoring module 126 adjusts the specific vendors overall score increases. Similarly, in some implementations, the vendor selection module 130 will determine that a user has given poor feedback to a specific vendor. In this case, the vendor selection module 130 will adjust the vendors overall score down in response. The amount that the vendors overall score is adjusted up or down varies based on the number of positive or negative reviews received from the user and the degree to which the review is positive or negative.

In some implementations, the vendor selection module 130 determines user favored vendors based on the vendors having been recommended and reviewed positively by other users of the server system 120 in which the purchasing user has indicated trust. Thus, vendors recommended or reviewed positively by other trusted users will have their overall scores modified up and vendors reviewed negatively will have their overall scores modified down. In some implementations, the amount of modification depends on the level of trust between the two users. For example, User A has indicated a high level of trust in User B and User B has highly reviewed Vendor C. When the vendor selection module 130 is selecting a vendor for a user A purchase request 114, the overall score of Vendor C is modified up.

In some implementations, when the server system 120 receives user feedback, the communication module 122 updates the vendor database 140 to reflect the feedback. In this way, vendor scores can be adjusted up or down based on the received user feedback.
In some implementations, the vendor selection module 130 will select two or more potential vendors and transmit a vendor selection message to the client system 102. The user can then select a vendor from the potential vendor choices. In some implementations, the vendor selection module 130 sends the user indicated vendor and the highest scoring vendor. In some implementations, the vendor selection module 130 also transmits an explanation of why each vendor is offered as an option. In some implementations, the explanation is displayed in response to the user input having requested one or more explanations. For example, the vendor selection module 130 transmits a vendor selection message to the client system 102 Vendor A and Vendor B, noting that Vendor A has a lower price but with slower shipping and that Vendor B has a more expensive price but with faster shipping. In some implementations, the vendor selection message includes the specific price and supply times to help the user make the best decision.

In some implementations, the purchasing module 128 is configured to receive a purchase request from the vendor selection module 130. The purchase request includes a product to be purchased and a vendor to be used. In some implementations, the purchasing module 128 then enlists the communication module 122 to purchase the desired product from the determined vendor 170 available over the network 110. In some implementations, the purchasing module 128 arranges for delivery of the product or service according to the instructions from the user of the client system 102. In some implementations, purchases from vendors are made through an application programming interface (API). In other implementations, purchases are made manually by workers associated with the server system either in person, on the phone, or electronically (e.g., an employee calls a vendor to place an order for a server system that is available through the server system). In some implementations, the purchasing module 128 allows vendors to make bids on products. In this way, once a user has issued a buy product request, the purchase module 128 can notify one or more vendors, receive the vendors bids, and choose the vendor that offers the best value. In this way, vendors can have a publicly listed price, but then can bid at a lower price in certain situations without publicly publishing the lower price.

In some implementations, the purchasing module 128 purchases the requested products from users of the server system 120. In this way, users of the server system 120 can both buy and sell products through the service. In this way, the server system is able to find the best value for users of the server system 120. In some implementations, the purchasing module 128 negotiates with the right holders and organizing the product of a particular good or service. In some implementations, the purchasing module 128 is able to take orders for products and services that are created on demand or would normally not be available from a traditional commerce system.

In some implementations, the purchase request from a user includes a blend of “product” and “service” from various or multiple vendors. For example, the server system could coordinate the delivery of a swing set from a seller on Craigslist, and have a service provider to deliver it, sand it, re-stain it, and install it. In this case, the purchase module 128 would negotiate or order from each vendor and then arrange for co-ordination between the various providers to ensure that the product and services are delivered promptly and at a low cost. In this way users have a single point of contact (the server system) to get a good price on a complicated set of services and goods. This has the benefit of reducing price and difficulty.

In some implementations, the product or service requested is delivered repeatedly (e.g., a bacon of the month club) or contracted over a long time (e.g., a 2 year cell phone plan). In this case the purchase module 128 arranges for both payment and delivery on a regular schedule to provide convenience to both the vendor and the user.

In some implementations, the purchase module 128 can use coupons, discounts, and other means of adding marketplace value to a transaction on behalf of the user, even when the user is unaware of the coupons prior to placing a purchase order.

In some implementations, the product database 140 contains a plurality of product entries. Each product entry is associated with a single product and contains information related thereto. In some implementations, the product entries store the unique identifier of the product (e.g., bar code, name and module number, SKU number, or other product identifier), the specifications of the product (size, color, capacity, etc.), a list of vendors that can supply the product, the price of the product, and any other pertinent information. In some implementations, different vendors will have different prices and thus the product entry in the product database 140 will store a list of vendors and a vendor specific price for each vendor. For example, if a product is available from 4 vendors, the product database will store a separate price for each vendor.

In some implementations, the products in the product database 140 are added by partner vendors 150. In some implementations, the products in the database 140 are added by users and recommenders in the server system 120. In other implementations, the product database 140 is filled by crawlers using algorithms specified by the server system 120. In yet other implementations, the product database 140 is filled through a combination of crowd sourcing (e.g., users can submit products and associated information) and paid human labor (e.g., services where workers are paid a small fee to add new products to the database.) The products database 140 includes information for each product, including, but not limited to, the price of the product, its specification (size, color, etc.), the vendor or vendors from which it is available, whether the item is in stock, and whether it is available for delivery. In some implementations, the vendor selection module 130 and the purchasing module 128 use the information stored in the product database 140 to purchase the selected product. In some implementations, the product database 140 searches websites of vendors, or other databases maintained by vendors that are available over a network 110, for the desired product and updates the product database 140 based on information stored on websites or in catalogs. In some implementations, the information in the product database includes a time stamp or other date indication. Thus, if a product is on sale or otherwise time limited, the product database can notify users or make recommendations at the appropriate time.

In some implementations, the vendor database 150 includes a database of all the vendors associated with the server system 120. In some implementations, the vendor database includes vendor profiles 152. In some implementations, a vendor profile 152 includes the name of the vendor, vendor location, vendor nationality, one or more vendor category scores, user feedback for the vendor, and any other relevant
information. In some implementations, a vendor profile includes contact information submitted by the vendor or stored by the system after contact with the vendor including but not limited to websites, email addresses, phone numbers, user names, instant message user IDs, and social networking sites sufficient to allow the service to contact the vendor or an agent of the vendor. In some implementations, the communication module uses information stored in the vendor database to send communications to the vendor.

In some implementations, a vendor profile includes contact information submitted by the vendor or stored by the system after contact with the vendor including but not limited to websites, email addresses, phone numbers, user names, instant message user IDs, and social networking sites sufficient to allow the service to contact the vendor or an agent of the vendor. In some implementations, the communication module uses information stored in the vendor database to send communications to the vendor.

In some implementations, the user profile database contains user profiles for users who have registered to use the service provided by the server system. In some implementations, a user profile includes the name, ID, demographic information (gender, age, location, etc.), purchasing history, social network information, and trust information. The user profile database includes trust graph data. In some implementations, the trust graph data includes information describing a directed trust graph. The trust graph includes nodes, which represent users, and edges connecting nodes, which represent the trust level between the two users represented by the two nodes that the edge connects.

The user profile database includes trust graph data. In some implementations, the trust graph database includes information describing a directed trust graph. The trust graph includes nodes, which represent users, and edges connecting nodes, which represent the trust level between the two users represented by the two nodes that the edge connects.

In some implementations, the vendors are commercial organizations with products to sell. The vendors can receive orders to purchase products from the server system and fulfill those orders.

FIG. 2 is a block diagram illustrating a client system in accordance with some implementations. The client system typically includes one or more processing units (CPUs), one or more network interfaces, memory, and one or more communication buses for interconnecting these components. The client system includes a user interface. The user interface includes an associated display device and optionally includes an input means such as a keyboard, mouse, touch-sensitive display, or other input buttons. Optionally, the display device includes an audio device or other information delivery device. Furthermore, some client systems use a microphone and voice recognition to supplement or replace the keyboard.

Memory includes high-speed random access memory, such as DRAM, SRAM, DDR RAM or other random access solid state memory devices; and may include non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory may optionally include one or more storage devices remotely located from the CPU(s). Memory, or alternately the non-volatile memory device(s) within memory, includes a non-transitory computer readable storage medium. In some implementations, memory includes the computer readable storage medium of memory stores the following programs, modules and data structures, or a subset thereof:

- an operating system that includes procedures for handling various basic system services and for performing hardware dependent tasks;
- a network communication module that is used for connecting the client system to other computers via the one or more communication network interfaces (wired or wireless) and one or more communication networks, such as the Internet, other wide area networks, local area networks, metropolitan area networks, and so on;
- a display module for enabling display of media content on a display associated with the client system;
- one or more client system applications modules for enabling the client system to perform the functions offered by the client system, including but not limited to:
- a browser application for sending purchase requests to a server system and displaying the information (for example web pages) returned by the server system or a smart phone or other computer system application that performs the same function;
- a product request module for enabling a user to send a product request to a server system and receiving a vendor confirmation request from the server system to confirm the vendor selected by the server system;
- a product add request module for enabling a user to send a product add request to a server system to add a product to the product database; and/or
- one or more client data modules for storing data related to the client system, including but not limited to:
- client message data, including data representing messages to be sent to the server system and messages received from the server system;
- user profile data, including information concerning users of the client system such as a user profile, user preferences and interests, user contact information, and other information relevant to providing services to the user.

FIG. 3 is a block diagram illustrating a server system in accordance with some implementations. The server system typically includes one or more processing units (CPUs), one or more network interfaces, memory, and one or more communication buses for interconnecting these components.

Memory includes high-speed random access memory, such as DRAM, SRAM, DDR RAM or other random access solid state memory devices; and may include non-volatile memory, such as one or more magnetic disk storage devices, optical disk storage devices, flash memory devices, or other non-volatile solid state storage devices. Memory may optionally include one or more storage devices remotely located from the CPU(s). Memory, or alternately the non-volatile memory device(s) within memory, includes a non-transitory computer readable storage medium. In some implementations, memory includes the computer readable storage medium of memory stores the following programs, modules and data structures, or a subset thereof:

- an operating system that includes procedures for handling various basic system services and for performing hardware dependent tasks;
- a network communication module that is used for connecting the server system to other com-
computers via the one or more communication network interfaces 304 (wired or wireless) and one or more communication networks, such as the Internet, other wide area networks, local area networks, metropolitan area networks, and so on;

[0086] one or more server application module(s) 314 for enabling the server system 120 to perform the functions offered by the server system 120, including but not limited to:

[0087] a request processing module 124 for receiving purchase requests (FIG. 1, 114) and product add requests (FIG. 1, 118) from a client system (FIG. 1, 102) and processing the requests before passing relevant data to the vendor selection module 130 and the vendor scoring module 126;

[0088] a vendor scoring module 126 for generating overall category scores for vendors based on a variety of criteria, generating overall scores for vendors based on the criteria scores and priority information supplied by the user or the server system 102, and ranking vendors based on their overall scores;

[0089] a purchasing module 128 for purchasing a product or service from one of the vendors (FIG. 1, 170);

[0090] a vendor selection module 130 for identifying vendors (FIG. 1, 170) that offer a product or service received from the request processing module 124, transmitting the identified vendors to the vendor scoring module 126 to score the list of identified vendors, and selecting a vendor based on user preferences and overall vendor scores; and/or

[0091] one or more server data module(s) 330 for storing data related to the server system 120, including but not limited to:

[0092] user profile data 150 including user preferences and interests, user contact information, and user history, including past user purchases, searches, page views, previous product recommendations, previous product reviews, social connections of the user, preferred vendors, vendor reviews, and any other information related to the user;

[0093] vendor data 150 including a database of all the vendors associated with the server system 120 and the vendor profiles data 152 for each vendor;

[0094] product data 140 including information for products that may be purchased through the server system 120, including, but not limited to, the price of the product, its features (size, color, etc.), recent changes in price, and the vendor or vendors from which it is available; and/or

[0095] vendor profile data 152 including the name of the vendor, vendor location, vendor nationality, one or more vendor category scores, user feedback for the vendor, and any other relevant information.

[0096] FIG. 4 depicts a block diagram of an exemplary data structure for a user profile database 160 for storing information related to users of the server system (FIG. 1, 120). In accordance with some implementations, the user profile database 160 includes a plurality of user profiles 402-1 to 402-P, each of which corresponds to a user registered with the server system (FIG. 1, 120). In some implementations, each user profile 402 contains a user profile ID 404, a user history 406, trust information 408, recommendations 410 made by the user, user contact information 412, and vendor preferences 414.

[0097] In some implementations, user profile ID 404 is a value that uniquely identifies a specific user of the server system (FIG. 1, 120). In some implementations, this value is chosen by the user and is a user name. In other implementations, this value is assigned to the user by the server system (FIG. 1, 120) as part of the registration process.

[0098] In some implementations, the user history 406 stores information about the user's past interactions with the server system (FIG. 1, 120), including past user purchases and purchase details, searches, page views, previous product recommendations, previous product reviews, and social connections of the user including previously recorded trust information for other users and/or social information received from social networking sites.

[0099] In some implementations, trust information 408 includes data describing the social connections of the user and includes a trust level for other users of the server system (FIG. 1, 120). A trust level is a value indicating the degree to which a user trusts the opinions and recommendations of another user. In some implementations, trust information is explicitly submitted by the user, in other situations the server system (FIG. 1, 120) infers trust information from the actions of users, and in yet other situations trust information includes information from both user submissions and server inferences.

[0100] In some implementations, recommendation 410 data includes product purchases and product recommendations previously submitted by the user. In some implementations, contact information 412 includes a list of contact information for contacting the user through a plurality of communication methods and information describing if and when the server system (FIG. 1, 120) should use that method. For example, the contact information 412 includes but is limited to, email addresses, phone numbers, Twitter handle, social network ID, physical address, and any other information that helps the server system (FIG. 1, 120) contact the user. The user may select that the server system (FIG. 1, 120) should never use social network messaging to contact the user and should use email address at all times except for the weekend and that a text message to a mobile phone should be used on the weekend. In some implementations, the contact information 412 includes login and password information for one or more social networking sites.

[0101] In some implementations, vendor preference data 414 includes data indicating the user's vendor preferences. In some implementations, vendor preference data 414 includes vendors that the user has explicitly indicated as preferred vendors and vendors that the user has implicitly indicated as non-preferred vendors. In some implementations, vendor preference data 414 is determined based on previous user behavior (e.g., vendors that the user has repeatedly selected). In some implementations, a user's vendor preference is based on feedback or reviews from the user. Vendors that received positive reviews will be noted as potential favored vendors and vendors that receive negative reviews will be noted as potential disfavored vendor.

[0102] In some implementations, contact information stored for the user includes one or more ways to contact the user including, but not limited to, email addresses, phone
number, Twitter handle, social network ID, physical address and any other information that helps the server system (FIG. 1, 120) contact the user.

[0103] FIG. 5 depicts a block diagram of an exemplary data structure for a vendor database 150 for storing information related to vendors associated with the server system (FIG. 1, 120). In accordance with some implementations, the vendor database 150 includes a plurality of vendor profiles 152-1 to 152-15, each of which corresponds to a vendor associated with or accessible by the server system (FIG. 1, 120) (e.g., vendors from whom the server system can purchase products or services). In some implementations, each vendor profile 152 contains a vendor profile ID 504, a vendor name 506, vendor location data 508, user feedback data 510, vendor category scores 512, and contact information 514 for the vendor.

[0104] In some implementations, vendor profile ID 504 is a value that uniquely identifies a specific vendor associated with or accessible to the server system (FIG. 1, 120). In some implementations, this value is chosen by the vendor. In other implementations, this value is assigned to the vendor by the server system (FIG. 1, 120).

[0105] In some implementations, the vendor name 506 is a string that identifies the vendor to users of the server system (FIG. 1, 120). In some implementations, the vendor name is a name associated with the vendor that is well known to users. For example, if a brand name is more commonly known to users than the official corporate or legal name of the vendor, the brand name could be included as the vendor name 506.

[0106] In some implementations, vendor location 508 includes data describing the location of the vendor, including country of origin and the location of its headquarters and any distribution points. In some implementations, the vendor location 508 also includes estimated shipping times. In some implementations, the product information 510 includes data describing the products available from the vendor. In some implementations, product information 510 also includes prices, inventory status, delivery options, product customization options, and any other information related to the products offered by the vendor.

[0107] In some implementations, category score profiles 512 include a plurality of scores for a plurality of categories associated with the vendor. For example, relevant vendor category scores include separate scores for shipping time, user experience, average price, return policy, and any other categories that might be useful. In some implementations, the categories are predetermined by the server system.

[0108] In some implementations, each category score profile 512 includes a category number 520 that identifies the category. In some implementations, the category number 520 is an integer that identifies the same category in all vendor profiles 152. For example, the “average price” category is assigned the category number 21, the server system knows that every vendor category score with the category number 21 is average price. In some implementations, the category data 522 is text that identifies the type of category score. For example, category data would include a text description such as “user experience” or “return policy.”

[0109] In some implementations, each category score profile 512 includes a category score 524. In some implementations, a category score is a number within a range that represents how well the vendor performs on the metric measured by the category. In some implementations, all the category scores use the same scale. In other implementations, each category score profile 512 has a customized scale. For example, all category scores are a number between 0 and 10 with 0 being the lowest score and 10 being the highest. Each vendor is evaluated in one or more categories and is assigned a category score for each category. In some implementations, category scores 524 are updated as new information is received by the server system (FIG. 1, 120). In some implementations, new information is received in the form of user submitted feedback and reviews. In other implementations, new information is received from the vendors themselves (e.g., new price information or new shipping information).

[0110] In some implementations, each category score profile 512 includes a default weight 526. In some implementations, the default weight is the weight given to the category score when no specific weight for the category has been indicated by a user. For example, the default category weight for the category “average price” is high, while the default category weight for the category “return policy” is lower. In some implementations, the category score profile includes user feedback 528 received from users relevant to the category associated with the category score profile 512.

[0111] FIG. 6 is a flow diagram illustrating the process for click-less purchasing in accordance with some implementations. Each of the operations shown in FIG. 6 may correspond to instructions stored in a computer memory or computer readable storage medium. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders). In some implementations, the method described in FIG. 6 is performed by the server system (FIG. 1, 120).

[0112] In some implementations, the server system (FIG. 1, 120) creates (602) one or more vendor profiles. In some implementations, vendor profiles store information related to a vendor accessible to the server system (FIG. 1, 120). Stored information includes, but is not limited to, the name of the vendor, products offered by the vendor, price information for the products offered by the vendor, product ordering procedures for the vendor, vendor location, and shipping times and procedures. In some implementations, the vendor profiles store any information necessary for purchasing products or services from the vendor.

[0113] In some implementations, the vendor profiles include scores for the vendor for one or more attribute categories. In some implementations, creating vendor profiles includes determining (604) one or more vendor scoring categories. In some implementations, vendor scoring categories include average price, delivery time, and user satisfaction. In some implementations, the server system (FIG. 1, 120) then, for one or more determined vendor scoring categories, generates (606) a vendor category score based on past vendor performance. In some implementations, a vendor category score is an indication of how well the vendor performs in that category. For example, if the category was return policies, vendors with good return policies would receive high scores and vendors with poor return policies would receive low scores.

[0114] In some implementations, past vendor performance is based on user supplied feedback (608). In some implementations, user supplied feedback includes user reviews, user ratings, and user submitted recommendations. For example, a user submits ratings for a vendor based on a recent purchase and gives the vendor 5 stars out of 5 for shipping time and 4 out of 5 stars for customer service. The user ratings are then used to adjust the corresponding category scores.

[0115] In some implementations, past vendor performance is determined based on data directly measured by the server
system (FIG. 1, 120) (610). For example, the server system (FIG. 1, 120) tracks prices for products offered by a vendor at regular intervals and compares those prices against prices offered for the same products at other vendors during the same intervals. Based on the tracked prices, the server system (FIG. 1, 120) can generate a price average category score for each vendor. Thus, if the server system (FIG. 1, 120) periodically checks the current price for a specific product with a plurality of vendors, it can determine an average price. Based on the average price, the server system (FIG. 1, 120) can rate each vendor as below or above the average price and by collecting ratings for a plurality of products, the server system gives each vendor an average price score. In some implementations, the data measured by the computer system includes product delivery time, product price, and number of products in stock.

In some implementations, the server system (FIG. 1, 120) stores (614) one or more vendor profiles, wherein each vendor profile is associated with a particular vendor and includes one or more vendor category scores (612). Each category score is associated with a particular attribute of a vendor. In some implementations, the server system (FIG. 1, 120) receives (616) a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased. For example, the purchase request includes a request for a copy of season 2 of “Breaking Bad” on DVD. In some implementations, the request also includes a user preferred vendor for fulfilling the request. In some implementations, the purchase request does not include a preferred vendor, and the server system (FIG. 1, 120) accesses the product database to determine a list of vendors that sell the requested product. In some implementations, the product database includes a preferred vendor for the requested product.

In some implementations, the server system (FIG. 1, 120) determines (618) a vendor to supply the requested product based on the stored vendor profiles. In some implementations, the server system (FIG. 1, 120) selects the vendor included in the purchase request. In some implementations, the server system (FIG. 1, 120) selects a vendor from among a list of preferred vendors stored in the user profile of the user who submitted the request. In some implementations, the server system (FIG. 1, 120) selects a vendor by determining vendors preferred by users of the server system (FIG. 1, 120) that are trusted by the requesting user.

In some implementations, when the server system (FIG. 1, 120) determines (620) a vendor to supply the requested product based on the stored vendor profiles, the server system (FIG. 1, 120) further, for each vendor in the plurality of vendors: generates a overall vendor score based on the one or more vendor category scores wherein the one or more vendor category scores are weighted based on the priority associated with each category. In some implementations, the weights are based on category priorities received from the user. In some implementations, the server system (FIG. 1, 120) has default category weights that are used when no priorities are received from the user or stored in the user’s profile.

In some implementations, the server system (FIG. 1, 120) ranks (622) the plurality of vendors based on the generated overall vendor score. Thus the list includes a list of vendors ordered from highest overall vendor score (e.g., best vendor match) to lowest overall vendor score (e.g., worst vendor match). In some implementations, the priorities associated with the one or more vendor categories are predetermined by the computer system.

FIG. 7 is a flow diagram illustrating the process for click-less purchasing in accordance with some implementations. Each of the operations shown in FIG. 7 may correspond to instructions stored in a computer memory or computer readable storage medium. Optional operations are indicated by dashed lines (e.g., boxes with dashed-line borders). In some implementations, the method described in FIG. 7 is performed by the server system (FIG. 1, 120).

In some implementations, the server system (FIG. 1, 120) stores (702) user profiles for each user of the computer system; wherein user profiles include priorities associated with vendor categories. In some implementations, the server system (FIG. 1, 120), prior to purchasing the requested product, sends (704) a request to the user to confirm the determined vendor. In some implementations, the server system (FIG. 1, 120) receives (706) a response from the user confirming the determined vendor. In some implementations, the response authorizes the use of the determined vendor. In other implementations, the response from the user includes a rejection of the determined vendor.

In some implementations, the server system (FIG. 1, 120) purchases (708) the requested product from the determined vendor. In some implementations, the server system (FIG. 1, 120) only purchases the request product in response to a request from the user that authorizes the use of the determined vendor. In some implementations, the server system (FIG. 1, 120) arranges (710) for delivery of the purchased product or service.

In some implementations, the server system (FIG. 1, 120) receives (712) feedback from the user regarding the determined vendor. In some implementations, the server system (FIG. 1, 120) updates (714) the vendor profiles based on the received feedback. In some implementations, the server system (FIG. 1, 120) includes a product database (716). In some implementations, the server system (FIG. 1, 120) receives (718) a request to add a product to the product database, wherein the request includes a vendor associated with the product. In some implementations, the request to add a product to the product database is a recommendation from a user. In some implementations, the request to add a product to the product database is included in the purchase request.

Trust Graph

In some implementations, a server system for improving the reliability of an Internet based commerce service through social network based product recommendations using a trust graph. A trust graph represents trust between users. The server system maintains a record of the trust relationships for each user registered with the service provided by the server system. In some implementations, the server system then uses this information to identify the most trusted users in at least one area of a particular user’s trust graph.

In some implementations the server system receives a trust indication from a first user. The trust indication includes information identifying a second user of the plurality of users registered with the server system and a trust level that represents the level of trust that the first user has for the second user. In some implementations the trust graph is a directed graph, such that a particular trust indication only indicates the level of trust that the first user has for the second user and not the level of trust between the second user and the first user. As the first user sends more trust indications to the
server, the directed trust graph of the first user includes trust information and becomes more useful. For example, a first user Bob sends three trust indications to the server system, indicating Joe, Phil, and Deborah respectively, each with a high trust level. The server system will then store these trust indications in a directed trust graph for Bob. Because the trust graph is directed, the server system does not infer trust levels for Joe, Phil, and Deborah toward Bob based on Bob’s trust level towards them.

[0126] In some implementations, trust level is represented by a numerical value between 0 and 1, with 0 indicating no trust and 1 indicating maximum trust. In other implementations, the trust level is represented by a numerical value between -1 and 1, wherein 1 indicates maximum trust, -1 indicates maximum distrust, and 0 indicates no current trust information. For example, if Jim totally trusts his friend Pam, he would indicate a trust level of 1. For Jim’s friend Andy, who he trusts, but not as much as Pam, Jim would indicate a trust level score between 0 and 1, such as 0.5. Further, for a distrusted person, such as Dwight, Jim would indicate a score below 0 but above -1, such as -0.5.

[0127] In accordance with some implementations, the server system can infer trust information by identifying actions taken by a particular user and determining the trust level indicated by the identified actions. For example, if the server system determines that Jim has received a recommendation for a pair of shoes from Andy, the server system can measure Jim’s response to determine a trust level between Andy and Jim. If Jim purchases the recommended pair of shoes, the system will determine an increased level of trust from Jim to Andy. If Jim takes no action based on the recommendation the trust level will either remain unchanged or, if a pattern of ignoring recommendations from Andy is detected, slightly the level of trust from Jim and Andy.

[0128] In some implementations, each user of the system has a “trustworthiness” score. The trustworthiness score represents an overall rating of the quality and usefulness of that user’s recommendations. In some implementations the trustworthiness score is represented by a number between 0 and 1. In other implementations, the trustworthiness score is a value with a lower bound of 0, but no upper bound. In yet other implementations, the trustworthiness score could have a negative value. In this case, a respective user’s trustworthiness score increases in response to indications that the respective user’s recommendations are good. For example, as users of the system chose to buy a product or service in response to a recommendation from a particular user, the trustworthiness score for the particular user would increase. Correspondingly, recommendations that are ignored or explicitly rejected could result in either no change of trustworthiness score or in a lowered trustworthiness score.

[0129] In some implementations a higher trustworthiness score represents a higher level of trustworthiness within the system. In some implementations, if a user’s trustworthiness score exceeds a predetermined level the user will be noted as a tastemaker or a very influential user.

[0130] In some implementations, trust levels can be transitive between users. For example, if Jim trusts Pam with a trust level of 1 and Pam trusts Michael with a score of 0.75, the server system can calculate a trust level from Jim to Michael. In some implementations the server system calculates transitive trust by taking the trust level from Pam to Michael and discounting it by some fixed amount. For example, the trust level may be reduced by 5% for each level of removal from Jim. So Jim’s trust level for Michael would be 0.75*0.95=0.7125. In some implementations, transitive trust values are calculated before any specific need for those calculated values arises. In other implementations, only direct trust values are stored in the trust graphs and transitive trust values are only calculated when required, such as when a request for a recommendation is received by the server system.

[0131] In some implementations transitive trust is calculated through multiple users. For example, User A trusts User B, User B trusts User C, and User C trusts User D. Transitive trust can be calculated for both Users C and D, and additionally for any users trusted by Users B, C, and D. In some implementations transitive trust is calculated for as many other users and through as many connections as possible. In other implementations, the server system limits the number of connections through which an implicit trust calculation is made. By limiting the number of connections (for example, no more than 10 connections) the server system avoids using resources on very tenuous connections.

[0132] In some implementations the transitive trust calculations result in more than one possible value of transitive trust from first user to a second user. For example, if user A trusts Users B and C, and both Users B and C trust User D, the value of the implicit trust calculation will depend on whether the connection is made through user B or user C. In some implementations all possible trust levels are averaged to determine the actual implicit trust level. In other implementations the server system selects either the highest or lowest trust level. In yet other implementations, the server system selects the implicit trust value that relies on the fewest number of connections. If multiple implicit trust values are tied for the fewest number of connections, the multiple trust values are averaged.

[0133] In some implementations, an aggregate trust value is calculated for multiple trust paths by using a probabilistic combination algorithm. Such an algorithm is AggregateTrustLevel = \((1-\cdot\cdot\cdot p_n)^{-1}\) where \(p_1\) are trust levels from a plurality of different possible trust paths. For example if one path gives a trust value of 0.8 and another gives a trust value of 0.63, the aggregate trust level would equal \((1-(0.8))^{-1}\). This function can be used for an arbitrary number of different trust paths.

[0134] In some implementations a trust indication from a first user to a second user indicates a specific category in which the indicated trust level applies. For example, Bob could trust Alice at a high level in one category of product, such as consumer electronics, but trust Alice at a low level for a second type of product, such as men’s shoes. Thus, when the first user submits a trust indication of a trust level for the second user, the trust indication includes the category of product or service for which the trust level applies. In this way the trust graph can include different trust levels for different categories of products. In some implementations, the different categories are arranged in a hierarchical format. In some arrangements the highest hierarchical level is a general level and there are a plurality of sub levels underneath the top general level such as “fashion,” “electronics,” and “media” among others. Each sublevel has more specific sub levels under them. For example, the sublevel “fashion” could include the sublevels of “women’s wear,” “men’s wear” and “kid’s wear.”

[0135] In some implementations, the server system can propagate trust levels from a sub-level to hierarchical levels higher, more general categories in the hierarchy of categories.
and to lower, more specific categories in the hierarchy of categories. In some implementations, trust scores are passed to lower, more specific categories without alteration, but trust levels are discounted when passed to higher, more general categories in the hierarchy. For example, Jim trusts Andy 0.5 in the category “Shoes.” That score is propagated to lower, more specific categories, such as “Athletic shoes” or “Formal Shoes,” without discount. However, when the trust level is propagated to higher, more general category, such as “Fashion,” the trust level is discounted by 10%. Thus, the graph would determine that Jim trusts Andy in “Fashion” at the level of 0.45.

[0136] In some implementations, the server system uses the gathered trust information to improve a user’s experience. The server system allows users to request recommendations for goods and services that the user is interested in purchasing. The server system uses the stored trust graphs to identify trusted potential recommenders. The server system first receives a request for a recommendation from a first user. The request for recommendation includes a category of good or service that defines the type of recommendation the first user wants. The server system uses the trust graph associated with the first user, among other factors, to determine a second user from whom to request a recommendation.

[0137] In some implementations, when a user first registers with the server system to participate in the online commerce system, the user has little or no trust information regarding other users. The server system can collect user information from other social networks (Facebook, Twitter, LinkedIn, etc.) or from webmail address books to identify other users of the server system that the new user might wish to trust. The server system then displays trust recommendations to the user. In some implementations, the server system only accesses the other social network information with the express permission of the user. In some implementations, the server system can gather demographic information from the new user and uses that demographic information (for example, user age and location) to determine recommended products and other users the potential new user may wish to trust.

[0138] In some implementations, the server system uses the social network and demographic information gathered from external sources, as described above, to determine recommendations from the user’s social network prior to receiving accurate trust level information from the user. In some implementations, the server system uses general product popularity data to find recommendations for users without sufficient trust information. General product popularity data is data that describes how popular particular products are within their product category. In other implementations, the server system uses direct editorial control (an editor selecting specific recommendations) to supply recommendations prior to receiving adequate trust level information from the user.

[0139] In some implementations, the server system will identify one or more users not already trusted by a first user (User A) that the server system has determined are likely to be trusted by the first user. In some implementations the server system identifies the one or more users by finding connections or similarities between the one or more users and the first user. For example, if multiple users trusted by the first user all trust a second user the server system determines that the first user will be likely to trust the second user. In other implementations, the server system will identify one or more users that have similar identified interests as the first user or have recommended similar products to those recommended by the first user. In yet other implementations, the server system identifies one or more users likely to be trusted by the first user using social connection information from third party websites or services, interactions that the users have had with the server system, and any other information the server system has with regard to the users.

[0140] In some implementations, the server system identifies a particular category in which the first user is likely to trust the one or more users. In some implementations, the server system identifies one or more users likely to be trusted by the first user in response to a request from the first user for recommendations of users to trust. In some implementations, the server system transmits the identified one or more users to the first user for display. In some implementation the server system also transmits information describing the reasons why the server system selected a particular user that can be displayed either with the recommendation itself or at the request of the first user. For example, the server system transmits a recommendation for Bob to trust Ernie and includes the fact that Ernie was identified because three people trusted by Bob (Alice, Carol, and Dan) also trust Ernie.

[0141] In some implementations, the one or more identified users are recommended only in a particular category. In response to receiving a recommendation to trust a second user in a specific category, the first user can choose to trust the user only in the recommended category, trust the user in a narrower category than the recommended category, or trust the user in a broader category than the recommended category. For example, the server system recommends that Bob trust Ernie in Men’s Formal Clothes. Bob has the option of trusting Ernie in only Men’s Formal Clothes, in a narrow category such as Men’s Belts, or a broader category such as Menswear. In some implementation, the first user can request to view public information about the identified on or more users in determining whether to trust the user or not. In some implementations, the first user can only see information that the identified one or more users have explicitly marked as public. For example, Ernie may have made his recommendations in Menswear public, and Bob can view the recommendations as part of his decision whether or not to trust Ernie.

[0142] In accordance with some implementations in response to receiving a request for a recommendation, the server system determines a list of users in the first user’s trust graph that have trust scores in the category requested by the user. The determined list has one or more other users. The server system then ranks the one or more users in order of trust level. In some implementations, the client system calculates transitive trust values for users with whom the first user does not have a direct trust level and includes at least some of these users in the list. Transitive trust values may also be pre-calculated and stored in the server system. When a server system receives a request for a recommendation, the server system determines whether additional transitive trust values need to be calculated. In some implementations, only some transitive trust values are pre-calculated and others are calculated as needed. For example, only first-order transitive trust values are pre-calculated and all other transitive trust values are calculated as needed. First order transitive trust values that only have one indirect trust link. For example, if Bob directly trusts Alice and Cody, then the first order transitive trust values are calculated for the users directly trusted by Alice and Cody.

[0143] The foregoing description, for purpose of explanation, has been described with reference to specific implemen-
tations. However, the illustrative discussions above are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The implementations were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best use the invention and various implementations with various modifications as are suited to the particular use contemplated.

[0144] It will also be understood that, although the terms first, second, etc., may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first contact could be termed a second contact, and, similarly, a second contact could be termed a first contact, without departing from the scope of the present implementations. The first contact and the second contact are both contacts, but they are not the same contact.

[0145] The terminology used in the description of the implementations herein is for the purpose of describing particular implementations only and is not intended to be limiting. As used in the description of the implementations and the appended claims, the singular forms “a,” “an,” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will also be understood that the term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will be further understood that the terms “comprises” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0146] As used herein, the term “if” may be construed to mean “when” or “upon” or “in response to determining” or “in response to detecting,” depending on the context. Similarly, the phrase “if it is determined” or “if (a stated condition or event) is detected” may be construed to mean “upon determining” or “in response to determining” or “upon detecting (the stated condition or event)” or “in response to detecting (the stated condition or event),” depending on the context.

What is claimed is:

1. A method, comprising:
at a computer system having one or more processors and memory storing one or more programs for execution by the one or more processors:
storining one or more vendor profiles, wherein each vendor profile is associated with a particular vendor and includes one or more vendor category scores;
receiving a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased;
determining a vendor to supply the requested product based on the stored vendor profiles;
purchasing the requested product from the determined vendor; and
arranging for delivery of the purchased product or service.

2. The method of claim 1, wherein determining a vendor to supply the requested product based on the stored vendor profiles includes:
for each vendor in the plurality of vendors:
generating an overall vendor score based on the one or more vendor category scores wherein the one or more vendor category scores are weighted based on the priority associated with each category;
ranking the plurality of vendors based on the generated overall vendor score.

3. The method of claim 2, wherein the priorities associated with the one or more vendor categories are predetermined by the computer system.

4. The method of claim 2, further including:
storing user profiles for each user of the computer system;
wherein user profiles include priorities associated with vendor categories.

5. The method of claim 1, further including:
creating vendor profiles, wherein creating vendor profiles includes:
determining one or more vendor scoring categories for one or more determined vendor categories, generating a vendor category score based on past vendor performance.

6. The method of claim 5, wherein past vendor performance is based on user supplied feedback.

7. The method of claim 5, wherein past vendor performance is determined based on data directly measured by the computer system.

8. The method of claim 7, wherein the data measured by the computer system includes product delivery time, product price, and number of products in stock.

9. The method of claim 1, includes:
receiving feedback from the user regarding the determined vendor; and
updating the vendor profiles based on the received feedback.

10. The method of claim 1, wherein the computer system includes a product database, and the method further includes:
receiving a request to add a product to the product database, wherein the request includes a vendor associated with the product.

11. The method of claim 10, wherein the request to add a product to the product database is a recommendation from a user.

12. The method of claim 10, wherein the request to add a product to the product database is included in the purchase request.

13. The method of claim 1, further including:
prior to purchasing the requested product, sending a request to the user to confirm the determined vendor; and
receiving a response from the user confirming the determined vendor.

14. A server system for rewarding users in a social referral system, the server system comprising:
one or more processors and
memory storing one or more programs to be executed by the one or more processors;
the one or more programs comprising instructions for:
stoiring one or more vendor profiles, wherein each vendor profile is associated with a particular vendor and includes one or more vendor category scores;
receiving a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased;
determining a vendor to supply the requested product based on the stored vendor profiles;
purchasing the requested product from the determined vendor; and
arranging for delivery of the purchased product or service.
15. The server system of claim 14, further including instructions for:
creating vendor profiles, wherein creating vendor profiles includes:
determining one or more vendor scoring categories
for one or more determined vendor categories, generating a vendor category score based on past vendor performance.

16. The server system of claim 14, further including instructions for:
receiving feedback from the user regarding the determined vendor; and
updating the vendor profiles based on the received feedback.

17. A non-transitory computer readable storage medium storing one or more programs configured for execution by an electronic device, the one or more programs comprising instructions for:
creating vendor profiles, wherein each vendor profile is associated with a particular vendor and includes one or more vendor category scores;
receiving a purchase request from a user of the server system, wherein the purchase request includes a product or service to be purchased;
determining a vendor to supply the requested product based on the stored vendor profiles;
purchasing the requested product from the determined vendor; and
arranging for delivery of the purchased product or service.

18. The non-transitory computer readable storage medium of claim 14, further including instructions for:
creating vendor profiles, wherein creating vendor profiles includes:
determining one or more vendor scoring categories for one or more determined vendor categories, generating a vendor category score based on past vendor performance.

19. The non-transitory computer readable storage medium of claim 14, further including instructions for:
receiving feedback from the user regarding the determined vendor; and
updating the vendor profiles based on the received feedback.

* * * * *