

(12) STANDARD PATENT APPLICATION (11) Application No. AU 2017210638 A1
(19) AUSTRALIAN PATENT OFFICE

(54) Title
SOCIAL MATCH PLATFORM APPARATUSES, METHODS AND SYSTEMS

(51) International Patent Classification(s)
G06Q 30/06 (2012.01) **G06Q 30/02** (2012.01)
G06F 17/30 (2006.01) **G06Q 50/30** (2012.01)

(21) Application No: **2017210638** (22) Date of Filing: **2017.08.04**

(43) Publication Date: **2017.08.24**

(43) Publication Journal Date: **2017.08.24**

(62) Divisional of:
2012272664

(71) Applicant(s)
MONSTER WORLDWIDE, INC

(72) Inventor(s)
CHEVALIER, Thomas;MUND, Matthew;INCE, Kristi;DELLOVO, Dee

(74) Agent / Attorney
Griffith Hack, GPO Box 4164, Sydney, NSW, 2001, AU

ABSTRACT

The SOCIAL MATCH PLATFORM APPARATUSES, METHODS AND SYSTEMS ("SMP") transforms platform join requests, social network info, and SMP network info
5 inputs via SMP components NJ, JIP, CIP, OP, CN-SGU and CN-UPSOG into job info, candidate info, offer info, and social meetup info outputs. A job information request for a candidate may be obtained. Social data associated with the candidate may be determined. A social job relevancy rating for various jobs may be calculated using the social data. A job may be selected using the social job relevancy rating for the job, and
10 information regarding the selected job may be provided.

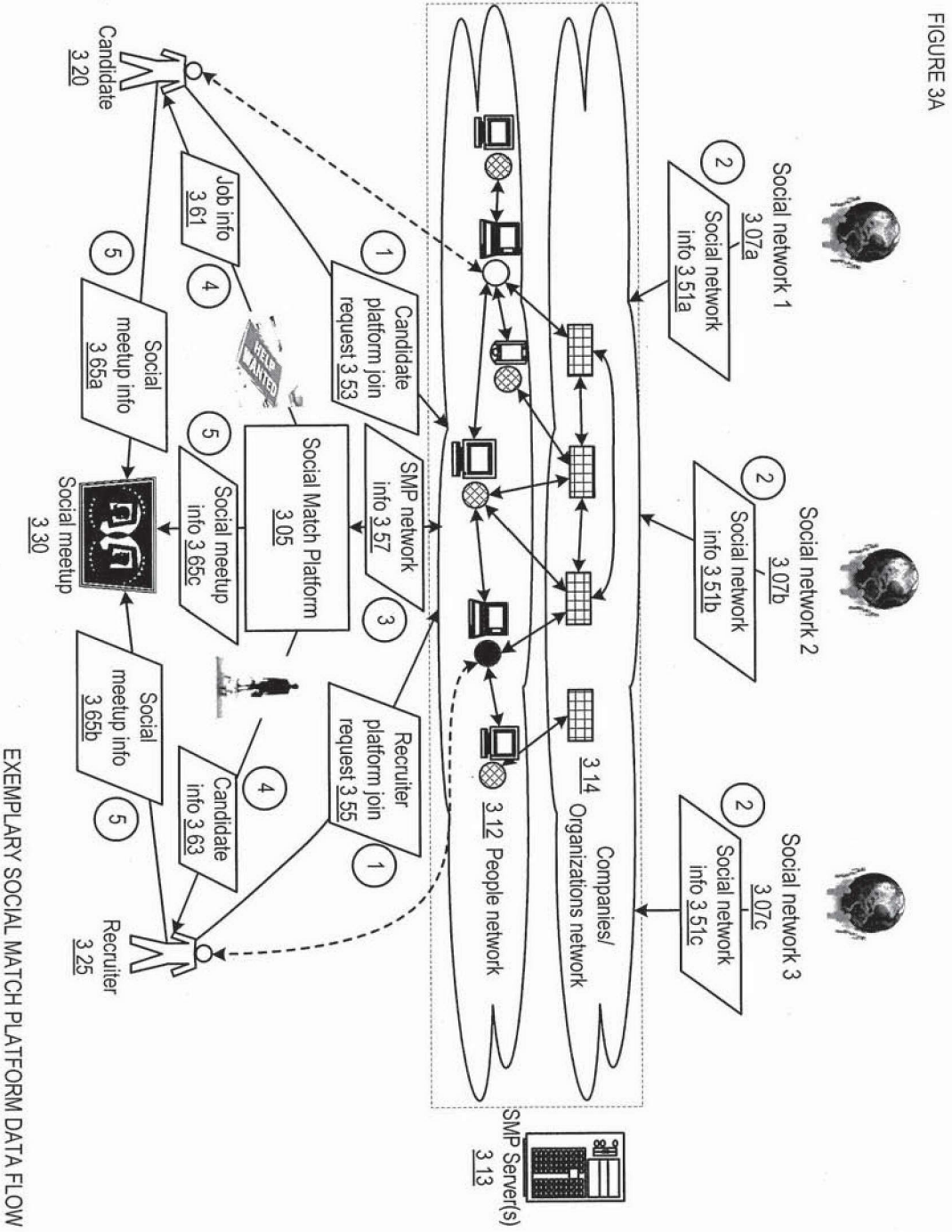


FIGURE 3A

EXEMPLARY SOCIAL MATCH PLATFORM DATA FLOW

1 **SOCIAL MATCH PLATFORM APPARATUSES, METHODS AND**
2 **SYSTEMS**

3 **[0001]** This application for letters patent disclosure document describes inventive
4 aspects directed at various novel innovations (hereinafter “disclosure”) and contains
5 material that is subject to copyright, mask work, and/or other intellectual property
6 protection. The respective owners of such intellectual property have no objection to the
7 facsimile reproduction of the disclosure by anyone as it appears in published Patent
8 Office file/records, but otherwise reserve all rights.

9 **PRIORITY CLAIM**

10 **[0002]** This application claims priority under 35 USC § 119 to United States
11 provisional patent application serial no. 61/501,095 filed June 24, 2011, entitled “Social
12 Match Platform Apparatuses, Methods and Systems,” attorney docket no. 17199-047PV.

13 **[0003]** The entire contents of the aforementioned application(s) are expressly
14 incorporated by reference herein.

15 **FIELD**

16 **[0004]** The present innovations are directed generally to matching people,
17 companies, organizations, and/or the like that may benefit from being connected (e.g.,
18 job candidates and recruiters, donors and charitable organizations, advertisers and
19 target audiences, self forming groups, and/or the like) using a social platform, and more
20 particularly, to SOCIAL MATCH PLATFORM APPARATUSES, METHODS AND
21 SYSTEMS (hereinafter "SMP").

22 **BACKGROUND**

23 **[0005]** Multiple social networking websites have been created over the last few
24 years. Some examples include Facebook, LinkedIn, MySpace, Orkut, Friendster, and

1 Twitter. Some social networking websites provide Application Programming Interfaces
2 (APIs) to allow others to programmatically access data collected by these social
3 networking websites.

4 **BRIEF DESCRIPTION OF THE DRAWINGS**

5 **[0006]** The accompanying appendices and/or drawings illustrate various non-
6 limiting, example, innovative aspects in accordance with the present descriptions:

7 **[0007]** FIGURE 1 shows a block diagram illustrating an exemplary SMP usage
8 scenario in one embodiment of the SMP;

9 **[0008]** FIGURE 2A shows a block diagram illustrating another exemplary SMP
10 usage scenario in one embodiment of the SMP;

11 **[0009]** FIGURE 2B shows a block diagram illustrating another exemplary SMP
12 usage scenario in one embodiment of the SMP;

13 **[0010]** FIGURE 3A shows a data flow diagram in one embodiment of the SMP;

14 **[0011]** FIGURE 3B-3F show various screen shots of implementations of the SMP;

15 **[0012]** FIGURE 4 shows a logic flow diagram illustrating a Network Join (NJ)
16 component in one embodiment of the SMP;

17 **[0013]** FIGURE 5 shows a logic flow diagram illustrating a Job Info Providing
18 (JIP) component in one embodiment of the SMP;

19 **[0014]** FIGURE 5 shows a logic flow diagram illustrating a Candidate Info
20 Providing (CIP) component in one embodiment of the SMP;

21 **[0015]** FIGURE 6 shows a logic flow diagram illustrating an Offer Providing (OP)
22 component in one embodiment of the SMP;

23 **[0016]** FIGURE 7 illustrates relationships between an embodiment of an
24 application for user interaction and various other modules or data stores, such as may
25 be embodied in a medium or in media;

1 **[0017]** FIGURE 8 illustrates an embodiment of a user interaction application as it
2 may be embodied in a medium or in media;

3 **[0018]** FIGURE 9 illustrates an embodiment of an application for data analysis
4 and various other modules or data stores, such as may be embodied in a medium or in
5 media;

6 **[0019]** FIGURE 10 illustrates an embodiment of an analysis application as it may
7 be embodied in a medium or in media;

8 **[0020]** FIGURE 11 illustrates an embodiment of a method of progressively
9 administering tests, providing information and selling merchandise;

10 **[0021]** FIGURE 12 illustrates relationships between an individual profile and
11 various sources of data or data stores in one embodiment;

12 **[0022]** FIGURE 13 illustrates relationships between a connector and other users
13 in one embodiment;

14 **[0023]** FIGURE 14 illustrates relationships between users in one embodiment;

15 **[0024]** FIGURE 15 illustrates an embodiment of a process of interaction between
16 a group and a vendor;

17 **[0025]** FIGURE 16 illustrates relationships between users with coupons or offers
18 in one embodiment;

19 **[0026]** FIGURE 17 illustrates a high-level data flow diagram associated with an
20 embodiment of the invention;

21 **[0027]** FIGURES 18A, 18B, and 18C illustrate flow diagrams of resume data
22 registration and user profile creation processes associated with embodiments of the
23 invention;

24 **[0028]** FIGURES 19A and 19B illustrate flow diagrams associated with resume
25 data submission processes;

26 **[0029]** FIGURES 20A and 20B illustrate flow diagrams associated with form
27 population and resume/cover letter generation processes;

1 **[0030]** FIGURES 21A, 21B, and 21C illustrate examples of invocation of the
2 AODSA tool according to embodiments of the invention;

3 **[0031]** FIGURE 22 illustrates an example of AODSA tool based on data served by
4 an ad server protocol;

5 **[0032]** FIGURE 23 illustrates additional aspects of the ad server AODSA tool
6 illustrated in Fig. 22;

7 **[0033]** FIGURE 24 shows an overview of entities and data flow in one
8 embodiment of CSE operation;

9 **[0034]** FIGURE 25 shows an implementation of application modules and
10 databases communicatively coupled to the CSE in one embodiment of CSE operation;

11 **[0035]** FIGURE 26A shows an implementation of combined logic and data flow
12 for acquiring and processing career data inputs in one embodiment of CSE operation;

13 **[0036]** FIGURE 26B shows an implementation of combined logic and data flow
14 for processing career data inputs in one embodiment of CSE operation;

15 **[0037]** FIGURE 27A shows a schematic illustration of resume data record
16 generation in one embodiment of CSE operation;

17 **[0038]** FIGURE 27B shows a schematic illustration of experience to state
18 conversion in one embodiment of CSE operation;

19 **[0039]** FIGURE 27C shows an implementation of logic flow for experience to state
20 conversion in one embodiment of CSE operation;

21 **[0040]** FIGURE 27D shows an implementation of a raw resume data record and a
22 state converted resume data record in one embodiment of CSE operation;

23 **[0041]** FIGURE 28 shows an implementation of combined logic and data flow for
24 building a state data record in one embodiment of CSE operation;

25 **[0042]** FIGURE 29A shows an implementation of combined logic and data flow
26 for processing state data to develop the statistical model in one embodiment of CSE
27 operation;

1 **[0043]** FIGURE 29B shows an implementation of combined logic and data flow
2 for processing state data to develop the statistical model in another embodiment of CSE
3 operation;

4 **[0044]** FIGURE 30 shows an implementation of logic flow for development of a
5 path-independent statistical model in one embodiment of CSE operation;

6 **[0045]** FIGURE 31 shows an implementation of a path-independent state model
7 data record in one embodiment of CSE operation;

8 **[0046]** FIGURE 32 shows an implementation of logic flow for development of a
9 path-independent statistical model with attributes in one embodiment of CSE
10 operation;

11 **[0047]** FIGURE 33 shows an implementation of a path-independent model with
12 attributes data record in one embodiment of CSE operation;

13 **[0048]** FIGURE 34 shows an illustration of career path modeling using path-
14 independent and path-dependent statistical models in one embodiment of CSE
15 operation;

16 **[0049]** FIGURE 35 shows an implementation of logic flow for development of a
17 path-dependent statistical model in one embodiment of CSE operation;

18 **[0050]** FIGURE 36 shows an implementation of a path-dependent statistical
19 model data record in one embodiment of CSE operation;

20 **[0051]** FIGURES 37A-B show an implementation of logic flow for development
21 and of a path-dependent statistical model in another embodiment of CSE operation; and

22 **[0052]** FIGURE 38 is of a mixed block, data and logic flow diagram illustrating
23 embodiments of the APPARATUSES, METHODS AND SYSTEMS FOR
24 ADVANCEMENT PATH TAXONOMY (hereinafter “APT”);

25 **[0053]** FIGURE 39 is of a logic flow diagram illustrating embodiments of the APT;

26 **[0054]** FIGURE 40 is of a logic flow diagram illustrating path-independent (i.e.,
27 targeted) path construction embodiments of the APT;

- 1 **[0055]** FIGURE 41 is of a logic flow diagram illustrating iteration-wise path-
2 independent path construction embodiments of the APT; and
- 3 **[0056]** FIGURE 42 is of a logic flow diagram illustrating iteration-wise path-
4 dependent path construction embodiments of the APT; and
- 5 **[0057]** FIGURE 43 is of a logic flow diagram illustrating N-part path-independent
6 path construction embodiments of the APT; and
- 7 **[0058]** FIGURE 44 is of a logic flow diagram illustrating N-part path-dependent
8 path construction embodiments of the APT; and
- 9 **[0059]** FIGUREs 45 and 46 is of a logic flow diagram illustrating gap analysis
10 embodiments of the APT; and
- 11 **[0060]** FIGUREs 47, 48, 49 are of a screen shot diagram illustrating embodiments
12 of the APT;
- 13 **[0061]** FIGURE 50 is a block diagram illustrating job carousel embodiments of
14 the APT;
- 15 **[0062]** FIGURE 51 is a logic flow diagram illustrating embodiments for invoking
16 and displaying a APT;
- 17 **[0063]** FIGURE 52is a logic flow diagram illustrating embodiments for tracking
18 seeker interactions with a APT;
- 19 **[0064]** FIGURE 53 is a block diagram illustrating feedback interactions with a
20 APT; and
- 21 **[0065]** FIGURE 54 is of a logic flow diagram illustrating benchmarking
22 embodiments for the APT;
- 23 **[0066]** FIGURE 55 is of a block diagram illustrating benchmarking interface
24 embodiments for the APT;
- 25 **[0067]** FIGURE 56 is of a mixed logic and block diagram illustrating path cloning
26 embodiments for the APT;
- 27 **[0068]** FIGURE 57 is of a mixed block and data flow diagram illustrating
28 advancement taxonomy embodiments for the APT;

1 **[0069]** FIGURE 58 is of a block diagram illustrating advancement taxonomy
2 relationships and embodiments for the APT;

3 **[0070]** FIGURE 59 is a diagram illustrating the entities that interact with the
4 system according to an embodiment of the invention;

5 **[0071]** FIGUREs 60A-60B are flow diagrams illustrating aspects of the
6 interaction between a system user and the system according to an embodiment of the
7 invention;

8 **[0072]** FIGURE 61A-61B illustrate aspects of a system user ad management tool,
9 according to an embodiment of the invention;

10 **[0073]** FIGURE 62A illustrates aspects of a base data entry, according to an
11 embodiment of the invention;

12 **[0074]** FIGURE 62B illustrates aspects of the base data entry conversion process
13 for creating an advertisement, according to an embodiment of the invention;

14 **[0075]** FIGURE 63 illustrates aspects of examples advertisement generated by the
15 system, according to an embodiment of the invention;

16 **[0076]** FIGURE 64 illustrates aspects of ad evolution, according to an
17 embodiment of the invention;

18 **[0077]** FIGURE 65 illustrates aspects of ad generation and distribution, according
19 to an embodiment of the invention;

20 **[0078]** FIGURE 66 illustrates aspects of advertisement generation and
21 distribution according to an embodiment of the invention;

22 **[0079]** FIGURE 67 illustrates aspects of a system database module;

23 **[0080]** FIGURE 68 illustrates an ad distribution module, according to an
24 embodiment of the invention;

25 **[0081]** FIGURE 69 illustrates another ad distribution module, according to an
26 embodiment of the invention;

27 **[0082]** FIGURE 70 illustrate aspects of advertisement performance analysis
28 according to an embodiment of the invention;

1 **[0083]** FIGUREs 71A-71C illustrate various configurations of the system
2 according to an embodiment of the invention;

3 **[0084]** FIGURE 72 is a diagram illustrating the entities that interact with the
4 system, according to an embodiment of the system;

5 **[0085]** FIGURE 73 is an overview flow diagram illustrating aspects of the
6 advertisement generation process, according to an embodiment of the system;

7 **[0086]** FIGUREs 74A-74E illustrate a flow diagram of the advertisement template
8 retrieval process, as well as examples of advertisement templates, according to an
9 embodiment of the system;

10 **[0087]** FIGURE 75A illustrates a flow diagram of the base data entry data
11 extraction process according to an embodiment of the system;

12 **[0088]** FIGURE 75B illustrates an example of a base data entry according to an
13 embodiment of the system;

14 **[0089]** FIGURE 75C illustrates examples of generated advertisements, according
15 to an embodiment of the system;

16 **[0090]** FIGURE 76A is a flow diagram illustrating aspects of the landing page
17 generation process, according to another embodiment of the system;

18 **[0091]** FIGURE 76B illustrates an example of a landing page generated by an
19 embodiment of the system;

20 **[0092]** FIGURE 77 is an ad creation management dashboard, according an
21 embodiment of the system;

22 **[0093]** FIGURE 78 is an overview of various entities that may interact with the
23 Engine at various points during system utilization;

24 **[0094]** FIGURE 79 is a high-level diagram illustrating aspects of the
25 advertisement targeting/distribution process;

26 **[0095]** FIGURE 80 illustrates aspects of an affiliate registration process according
27 to an implementation of the system;

- 1 **[0096]** FIGUREs 81A-81C illustrate aspects of sponsor system interaction
2 associated with establishing sponsor advertisement targeting/distribution parameters;
- 3 **[0097]** FIGUREs 82A-82B illustrates aspects of web user identification;
- 4 **[0098]** FIGURE 83 is a flow diagram illustrating aspects of a process flow
5 associated with advertisement targeting according to an implementation of the system;
- 6 **[0099]** FIGUREs 84A-84B illustrates aspects of creating advertisement request
7 message and processing the advertisement request message, respectively, according to
8 an implementation of the system;
- 9 **[00100]** FIGURE 85 illustrates a flow diagram of the targeting and distribution
10 process of another implementation of the system;
- 11 **[00101]** FIGURE 86 illustrates aspects of the advertisement distribution process
12 according to an implementation of the system;
- 13 **[00102]** FIGURE 87 provides a conceptual illustration of Ad evolution within an
14 embodiment of the present invention;
- 15 **[00103]** FIGURE 88 provides an overview of various entities that may interact with
16 the Engine at various points during system utilization;
- 17 **[00104]** FIGURE 89 exhibits a high-level flow diagram illustrating aspects of the
18 advertisement evolution process in one embodiment;
- 19 **[00105]** FIGURE 90 illustrates Ad generation in according to a system
20 embodiment;
- 21 **[00106]** FIGURE 91 illustrates one embodiment of the incorporation of the same
22 underlying BDE into different Ads using different Ad generation templates;
- 23 **[00107]** FIGUREs 92a-d show embodiments of Ads administering passive and
24 active performance metric registration;
- 25 **[00108]** FIGUREs 93a-c show logic flow for system driven registration of passive
26 performance metrics and web user drive registration of active performance metrics;
- 27 **[00109]** FIGURE 94 provides an illustrative example of the logic flow in one
28 embodiment of Ad evolution;

1 **[00110]** FIGURE 95 exhibits a schematic illustration of a three generation
2 evolution of Ad generation templates in one implementation of Ad duplication;

3 **[00111]** FIGURE 96 provides an illustrative example of the logic flow in another
4 embodiment of Ad evolution;

5 **[00112]** FIGURE 97 shows an illustrative example of Ad recombination in one
6 embodiment;

7 **[00113]** FIGURE 98 provides a visualization of three successive Ad generation
8 template generations in an Ad recombination embodiment;

9 **[00114]** FIGURE 99 shows a block diagram illustrating embodiments of a SMP
10 controller;

11 **[00115]** FIGURE 100 discloses an embodiment of a process tracking and
12 management environment in accordance with the present disclosure;

13 **[00116]** FIGURE 101 discloses an embodiment of a job application tracking and
14 management environment in accordance with the present disclosure;

15 **[00117]** FIGURE 102A discloses a flow diagram for a job listing search aspect of
16 one embodiment of the present disclosure;

17 **[00118]** FIGURE 102B discloses a flow diagram for a further embodiment of the
18 job listing search aspect of the present disclosure;

19 **[00119]** FIGURE 103A discloses a flow diagram for a cross correlation search
20 aspect of one embodiment of the present disclosure;

21 **[00120]** FIGURE 103B discloses a flow diagram for the cross correlation search
22 aspect of a particular implementation of the present disclosure;

23 **[00121]** FIGURE 103C discloses a flow diagram for a further implementation of the
24 correlation search aspect of the present disclosure;

25 **[00122]** FIGURE 104A discloses an embodiment of a job tracking extension in
26 accordance with the present disclosure;

27 **[00123]** FIGURE 104B discloses an overview of a scoring process for one
28 embodiment of the disclosed disclosure;

1 **[00124]** FIGURE 104C discloses a particular implementation of a job tracking
2 extension toolbar;

3 **[00125]** FIGURE 105A discloses a flow diagram for the listing match search of one
4 embodiment of the present disclosure;

5 **[00126]** FIGURE 105B discloses an overview of an analysis aspect of one
6 embodiment of the disclosed disclosure;

7 **[00127]** FIGURE 106A, 106B and 106C disclose logic flow diagrams of the results
8 processing match aspect for some embodiments of the present disclosure;

9 **[00128]** FIGURE 107A discloses an embodiment of an application history screen in
10 accordance with the present disclosure;

11 **[00129]** FIGURE 107B discloses a particular implementation of an application
12 history screen;

13 **[00130]** FIGURE 107C discloses a further embodiment of an application history
14 screen in accordance with the present disclosure;

15 **[00131]** FIGURE 108A discloses an embodiment of a job view screen in accordance
16 with the present disclosure;

17 **[00132]** FIGURE 108B discloses a particular implementation of a job view screen;

18 **[00133]** FIGURE 108C discloses a further embodiment of a job view screen in
19 accordance with the present disclosure;

20 **[00134]** FIGURE 109A discloses an embodiment of a company metrics screen in
21 accordance with the present disclosure;

22 **[00135]** FIGURE 109B discloses a particular implementation of a company metrics
23 screen;

24 **[00136]** FIGURE 110 A-B disclose a logic flow diagram for cross-network social
25 graph updating in an embodiment of the SMP;

26 **[00137]** FIGURE 111 discloses a logic flow diagram for cross-network user profile
27 sensitive query generation in one embodiment of the SMP;

1 **[00138]** FIGURES 112-124 disclose additional examples of SMP embodiments.

2 **[00139]** The leading number of each reference number within the drawings
3 indicates the figure in which that reference number is introduced and/or detailed. As
4 such, a detailed discussion of reference number 101 would be found and/or introduced
5 in Figure 1. Reference number 201 is introduced in Figure 2, etc.

6

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

DETAILED DESCRIPTION

Introduction

[00140] The SMP facilitates matching of people, companies, organizations, and/or the like that may benefit from being connected using information such as who you are, what you do, where you are located, what you are interested in, and/or the like using information sources such as social network data, location data, news and social media data, and/or the like. For example, a job candidate seeking a position at a company may benefit from having a contact at the company. The contact may be able to put the candidate in touch with a recruiter, recommend the candidate, help expedite processing of the candidate's job application, and/or the like. The SMP may facilitate these actions utilizing information regarding the candidate's social network and/or affiliation with companies and/or organizations, location, profile preferences, skills, experiences, education, and/or the like, and may also utilize such information to recommend relevant jobs to the candidate. Similarly, a recruiter seeking job candidates may benefit from having access to relevant contacts. Such contacts may facilitate finding a candidate, verifying the candidate's suitability and/or abilities, putting the recruiter in touch with the candidate, and/or the like. The SMP may facilitate these actions utilizing information regarding the recruiter's social network and/or affiliation with companies and/or organizations, location, profile preferences, skills, experiences, education, and/or the like, and may also utilize such information to recommend relevant candidates to the recruiter. Furthermore, a candidate and a recruiter may benefit from interacting with each other, and the SMP may facilitate such interaction via social meetup components (e.g., provided as part of the SMP and/or via a third party).

[00141] The SMP may be integrated into a social network or may be a stand alone application. In one embodiment, the SMP may be integrated into Facebook and be used as a Facebook application. In this instance, a user may be able to keep contacts on the SMP separate from their social contacts on the social network, thereby creating separation between personal or social contacts and professional contacts. Additionally,

1 by integrating the SMP with a social networking site such as Facebook, users are able to
2 build professional relationships in a convenient location - in this case, a website they
3 likely visit frequently. This may also be integrated into a mobile application for the
4 social networking site and/or it may be a standalone mobile application capable of
5 sourcing information from one or more social networking sites.

6 **[00142]** In some implementations, the SMP may have access to a user's social
7 network sites (regardless of whether or not the SMP is integrated as an application into
8 a particular social networking site. This allows the SMP to use the information posted
9 on these sites and contacts with whom the user communicates to enhance the SMP
10 content. For example, the SMP may be able to incorporate information in a user's post
11 about moving to serve ads to the user in a new city. Additionally, if a user is
12 communicating frequently with users in a different city than that in which the user
13 resides, the SMP may recognize that the user may want to move in an area closer to
14 his/her friends and therefore may incorporate this information and present the user
15 jobs in that city.

16 **[00143]** Because the SMP has access to a user's information on a social networking
17 site, the SMP may be able to provide ads for jobs within a user's social networking site.
18 For example, if a user is browsing Facebook, the user may see ads for jobs in which they
19 may be interested. These may be actively updated based on a user's recent searches,
20 recent conversations with other users, and/or the like. This is described in further detail
21 below in the section titled "Automated Online Data Submission." In some
22 implementations, the SMP ads may be hidden from the social network servers so
23 as to prevent integration between the social and professional networks.

24 **[00144]** Some implementations of the SMP may track job postings in which a user
25 shows interest. In some embodiments, the SMP may track the jobs viewed by a user,
26 and other implementations may allow a user to indicate that (s)he is interested in a
27 particular type of job (e.g., by hitting an "I'm interested" button, by saving the job to a
28 saved job list of jobs that the user may want to apply to at a later time, and/or the like).
29 This interest may be used by the SMP to compile a list of similar jobs to present to the
30 candidate. The SMP may also create a search profile for the user based on various types

1 of jobs in which a user has shown interest. The search profile may be used as a
2 secondary tool when a user searches for jobs. That is, if a user enters certain search
3 terms, the SMP may also use the profile as a way to prioritize certain job listings over
4 others when returning the search results to the user.

5 **[00145]** The SMP may prioritize certain listings over others for other reasons, as
6 well. For example, if a user has contacts at a particular company, these job listings may
7 be displayed above the postings by other companies where the user has no contacts. In
8 some implementations, the these contacts may be only the user's SMP contacts, but in
9 other implementations, the SMP may determine that a user has a social network contact
10 at a certain company with whom the user is not contacts on the SMP. In this scenario,
11 the SMP may suggest that the user connect with this contact on the SMP and/or
12 prioritize the listing into a second tier (below a first tier of companies where a user has
13 an SMP contact) and indicate to the user that (s)he has a contact through a social
14 networking site that (s)he may wish to contact if (s)he may consider and/or apply to the
15 job.

16 **[00146]** In some implemenations of the SMP, the user may post a job listing to the
17 SMP. The SMP may automatically push this job listing to the user's first degree contacts
18 (that is, contacts with whom the user is connected directly). Other implementations
19 may allow pushing the job to the user's second degree contacts (that is, the user's
20 contacts' contacts). In this implemenation, the second degree contacts who are
21 interested in the posted job listing may request an introduction to the user who posted
22 the job through their direct contact (the user's first degree contact). In additional
23 implementations, the SMP may display all jobs posted at contacts' employers to a user,
24 so a user would see all jobs posted by companies where their contacts are employed.
25 Some embodiments of this implementation may filter these employer jobs to those in
26 which a user may be interested. That is, if a user is a computer programmer and a
27 contact is employed by Microsoft, the user would see all jobs posted by Microsoft. In a
28 scenario in which these jobs are filtered, entry level positions at Microsoft may not be
29 displayed to a senior programmer.

30 **[00147]** Additional implemntations of the SMP may serve a user job postings in
31 which his/her contacts may have shown interest. In some scenarios, a programmer may

1 have many contacts who are also programmers. If one of the programmers in the user's
2 network shows interest in a particular job, this may be an indication that that particular
3 job may be of interest to his/her contacts who are also programmers. In this
4 implementation, it may be hidden from the user that the jobs displayed have been seen
5 or applied to by his/her contacts, or it may be anonymous as to which contact viewed
6 that job, but the user may know that one of his/her contacts viewed that job.

7 **[00148]** In some implementations, a user may view a home page when (s)he logs
8 into the SMP. This home page may display jobs a user might be interested in and/or
9 jobs that his/her contacts may have posted. These may be displayed in the order of
10 potential relevancy for the user, the most recent postings and/or the like.

11 **[00149]** In some scenarios, the user may be able to view a page of jobs in which the
12 user's contacts may be interested. In some embodiments of the SMP, the user may be
13 rewarded for recommending jobs to his/her contacts. For example, if a user has a
14 contact who is an electrician and a company posts a job for an electrician, the user may
15 recommend the job to his/her electrician contact. In some implementations, the reward
16 is monetary and is awarded if the connection is hired by the company. In alternative
17 embodiments, the user may receive an award if the contact shows interest in the
18 recommended job, if the user receives an interview, and/or the like. In some
19 implementations, the monetary award is paid by the company who posted the job and
20 the SMP facilitates contact and payment of the reward. Alternative embodiments may
21 reward the user within the SMP, for example, with a badge. Badges may be awarded to
22 users in several scenarios, such as when a user reaches a certain threshold of contacts,
23 has posted a certain number of jobs, has successfully recommended a certain number of
24 contacts, and/or the like.

25

SMP

26 **[00150]** FIGURE 1 shows a block diagram illustrating an exemplary SMP usage
27 scenario in one embodiment of the SMP. Figure 1 illustrates how a candidate 110 may
28 utilize the SMP to find a job based on contacts 120, 125. In Figure 1, a recruiter 115 may
29 post a job "Job 2" to the SMP. The candidate 110 may utilize the SMP to search for a job,

1 and, if there were no contact information, would be presented with jobs "Job 1", "Job 2",
2 and "Job 3", in that order (e.g., based on relevance matching of candidate skills to job
3 prerequisites). However, the SMP may utilize social information (e.g., including social
4 network data, location data, news and social media data, and/or the like) regarding the
5 candidate and the recruiter to adjust relevance ranking of the presented jobs. In this
6 example, the candidate and the recruiter have a contact 125 in common, and the SMP
7 may utilize this information to increase the relevance of the recruiter's posted job. That
8 is, the common contact between the recruiter and candidate elevates "Job 2" to be the
9 most relevant for the candidate. Accordingly, the candidate would be presented with
10 jobs "Job 2", "Job 1", and "Job 3", in that order. In some implementations, the SMP
11 may determine that a common contact exists by using a social networking site.

12 **[00151]** FIGURE 2A shows a block diagram illustrating another exemplary SMP
13 usage scenario in one embodiment of the SMP. Figure 2A illustrates how a candidate
14 210 and a recruiter 215 may utilize the SMP 205 based on social network contacts 220,
15 225 to find a job and a job candidate, respectively. In Figure 2A, the candidate 210 may
16 be looking for a job. The candidate may have a contact 220 that may know of a job that
17 would be of interest to the candidate. For example, the contact 220 may be connected to
18 another contact 225, who may be connected to a recruiter 215, who may be looking for a
19 job candidate. In other examples, the contact 220 may be connected to a recruiter
20 directly or separated from the recruiter by some number of degrees of separation, may
21 be a recruiter himself, may be connected to a recruiter through multiple connections,
22 and/or the like. The SMP 205 may analyze information regarding the social networks of
23 the various parties 210, 215, 220, 225 and other information (e.g., match between
24 candidate's skills and job prerequisites, affiliations with companies and/or
25 organizations, location, profile preferences, experiences, education, and/or the like) and
26 determine that a job offered by the recruiter 215 may be relevant to the candidate 210.
27 Accordingly, the SMP 205 may recommend the job to the candidate and/or facilitate
28 social meetup with the recruiter (e.g., via a chat or messaging application).

29 **[00152]** Similarly, the recruiter 215 may be looking for a job candidate. The
30 recruiter may have a contact 225 that may know of a candidate that would be of interest
31 to the recruiter. For example, the contact 225 may be connected to another contact 220,

1 who may be connected to the candidate 210, who may be looking for a job. The SMP 205
2 may analyze information regarding the social networks of the various parties 210, 215,
3 220, 225 and other information and determine that the candidate 210 may be relevant
4 to the recruiter 215. Accordingly, the SMP 205 may recommend the job seeker to the
5 recruiter and/or facilitate social meetup with the job seeker.

6 **[00153]** In some embodiments, the contact 220 is connected to both the recruiter
7 215 and the candidate 210. Similar to the above examples, in this implementation, the
8 SMP 205 may recommend the job to the candidate and/or facilitate a social meetup with
9 the job seeker, but alternatively or additionally, the SMP 205 may suggest to common
10 contact 220 that (s)he recommend the recruiter's job listing to the candidate 210. Some
11 implementations may provide a reward to the contact 220 for creating contact between
12 the candidate 210 and the recruiter 215. The SMP may provide the link between the job
13 seeker and the recruiter and may facilitate social meetup with the job seeker.

14 **[00154]** FIGURE 2B shows a block diagram illustrating another exemplary SMP
15 usage scenario in one embodiment of the SMP. Figure 2B illustrates how an applicant
16 240 to a college 230 may utilize the SMP 235 to interact with other SMP members 245,
17 250, 255 to determine whether this college would be a good place to attend. In Figure
18 2B, the applicant 240 may be trying to apply to various colleges and may wish to
19 determine whether the college 230 would be a good place to apply. The SMP 235 may
20 determine that a student 245, a professor 250, and an alum 255 may be helpful to the
21 applicant 240 in making this determination. For example, the SMP 235 may determine
22 that the student 245 is connected with both the college 230 and with the applicant 240,
23 which may make the student's description of the college (e.g., condition of the
24 dormitories) helpful to the applicant. In another example, the SMP 235 may determine
25 that the professor 245, who recently published a well renowned research article, teaches
26 a class at the college 230, which may make the professor's description of the college
27 (e.g., classes, research environment) helpful to the applicant. In another example, the
28 SMP 235 may determine that the alum 255 had the same major in college as the likely
29 major for the applicant 240 (e.g., based on the analysis of the applicant's interests),
30 which may make the alum's description of the college 230 (e.g., quality of classes)
31 helpful to the applicant. Accordingly, the SMP 235 may facilitate social meetup between

1 the applicant 240 and other SMP members 245, 250, 255 (e.g., between the applicant
2 and one other member, as a group with all four members, and/or the like).

3 **[00155]** FIGURE 3 shows a data flow diagram in one embodiment of the SMP. In
4 Figure 3, various entities (e.g., people, companies, organizations, groups, and/or the
5 like) may wish to become SMP members. For example, a candidate 320 may wish to
6 join the SMP 305 and become part of a people network 312. The candidate may be
7 represented in the people network by a white circle. The candidate may request to join
8 the SMP 305 via a candidate platform join request 353. In one embodiment, such a
9 candidate platform join request may be received via a website (e.g., via the Monster.com
10 website, such as when a user clicks a "Sign Up" button). In another embodiment, the
11 SMP may be integrated into a social networking site, such as Facebook, as an SMP app.
12 In this scenario, a candidate platform join request may be received via the SMP app
13 (e.g., in the case of a Facebook app, when a user requests to use the app). The candidate
14 platform join request 353 may include various pieces of information regarding the
15 candidate including personal information, contact information, previous and/or current
16 professional experience, previous and/or current education information, login
17 information for social networking websites and/or applications, social meetup
18 preference information, and/or the like. For example, the candidate platform join
19 request 353 may be in XML format substantially in the following form:

```
20 <XML>
21   <CandidatePlatformJoinRequest>
22     <PersonalInfo>
23       <Name>John Doe</Name>
24       <ScreenName>ScreenName1</ScreenName>
25     </PersonalInfo>
26     <ContactInfo>
27       <StreetAddress>123 Main Street, New York, NY</StreetAddress>
28       <Phone>(333)333-3333</Phone>
29     </ContactInfo>
30     <Experience>
31       <Position1>Job Title1, Experience1</Position1>
32       <Position2>Job Title2, Experience2</Position2>
33     </Experience>
34     <Education>
35       <Education1>University1, Degree1</Education1>
36       <Education2>University2, Degree2</Education2>
37     </Education>
38     <SocialNetworks>
39       <SocialNetwork1>LoginInfo1</SocialNetwork1>
```

```
1         <SocialNetwork2>LoginInfo2</SocialNetwork2>
2     </SocialNetworks>
3     <SocialMeetupPreferences>
4         <Preference>Allow all to request social meetup</Preference>
5     </SocialMeetupPreferences>
6 </CandidatePlatformJoinRequest>
7 </XML>
```

8
9 **[00156]** In one embodiment, the candidate may provide such information via the
10 candidate platform join request 353 upon signing up to use the SMP 305. In another
11 embodiment, the candidate may provide such information via a series of screens and/or
12 over time and the candidate platform join request 353 may comprise a plurality of
13 candidate platform join requests. In yet another embodiment, this information may be
14 sourced from a candidate profile that already exists, for example, from an existing
15 candidate profile on the Monster.com website, from a candidate's social networking
16 website profile, and/or the like.

17 **[00157]** In the example of an SMP Facebook app, this information may be used to
18 build and/or populate a user profile (example screen shot shown in Fig. 3B). The app
19 may be able to incorporate information that the user has already provided in his/her
20 Facebook profile to save the user time and avoid redundancy. In alternative SMP
21 Facebook app implementations, the user may opt not to include any information from
22 his/her Facebook page, and may instead opt to enter the information directly into the
23 app. Similarly, in some implementations, a user may opt to select friends with whom
24 (s)he wishes to connect professionally (i.e., on the SMP Facebook app), thereby keeping
25 the user's Facebook friends and social contacts separate from his/her professional
26 network. In some embodiments, the app may suggest Facebook friends and/or other
27 known user contacts with whom the user may wish to connect professionally to assist
28 the user in growing his/her professional network. An example screenshot may be seen
29 in Fig. 3C. In some embodiments, as a user builds his/her professional network, (s)he
30 may receive badges. In one embodiment, badges may be awarded for reaching a certain
31 specified number of connections (e.g., "First-Class Connector" badge for reaching 25
32 contacts, "Graduate Connector" for reaching 100 contacts, "Super Connector" for
33 reaching 250 contacts, and "Epic Connector" for reaching 500 contacts). In some
34 implementations, other badges may be awarded for various achievements, including, for
35 example an "Established" badge for working at one place for over 5 years, a "Loyal"

1 badge for working at one place for over 2 years, a badge for receiving a master's degree,
2 and/or the like. Sample badges are shown in the screenshot in Fig. 3D.

3 **[00158]** In another example, a recruiter 325 may wish to join the SMP 305 and
4 become part of the people network 312. The recruiter may be represented in the people
5 network by a black circle. The recruiter may request to join the SMP 305 via a recruiter
6 platform join request 355. In one embodiment, such a recruiter platform join request
7 may be received via a website. In another embodiment, such a recruiter platform join
8 request may be received via an app. The recruiter platform join request 355 may include
9 various pieces of information regarding the recruiter including personal information,
10 contact information, jobs info (i.e., info regarding available positions at the recruiter's
11 company), previous and/or current professional experience, previous and/or current
12 education information, login information for social networking websites and/or
13 applications, social meetup preference information, and/or the like. For example, the
14 recruiter platform join request 355 may be in XML format substantially in the following
15 form:

```

16 <XML>
17   <CandidatePlatformJoinRequest>
18     <PersonalInfo>
19       <Name>John Doe</Name>
20       <ScreenName>ScreenName1</ScreenName>
21     </PersonalInfo>
22     <ContactInfo>
23       <StreetAddress>123 Main Street, New York, NY</StreetAddress>
24       <Phone>(333)333-3333</Phone>
25     </ContactInfo>
26     <JobsInfo>
27       <Job1>Title1, prerequisites1, salary info1</Job1>
28       <Job2>Title2, prerequisites2, salary info2</Job2>
29     </JobsInfo>
30     <Experience>
31       <Position1>Job Title1, Experience1</Position1>
32       <Position2>Job Title2, Experience2</Position2>
33     </Experience>
34     <Education>
35       <Education1>University1, Degree1</Education1>
36       <Education2>University2, Degree2</Education2>
37     </Education>
38     <SocialNetworks>
39       <SocialNetwork1>LoginInfo1</SocialNetwork1>
40       <SocialNetwork2>LoginInfo2</SocialNetwork2>
41     </SocialNetworks>
42     <SocialMeetupPreferences>

```

```

1         <Preference>
2             Allow friends (1st degree) and friends of friends
3             (2nd degree) to request social meetup
4         </Preference>
5     </SocialMeetupPreferences>
6 </CandidatePlatformJoinRequest>
7 </XML>
8

```

9 **[00159]** In one embodiment, the recruiter may provide such information via the
10 recruiter platform join request 355 upon signing up to use the SMP 305. In another
11 embodiment, the recruiter may provide such information via a series of screens and/or
12 over time and the recruiter platform join request 355 may comprise a plurality of
13 recruiter platform join requests. Alternative embodiments may allow the recruiter to
14 import job details into the SMP from a job listing previously posted on a jobs website,
15 e.g. Monster.com. In other implementations, the information posted on the SMP may
16 be used to populate job listings on a jobs website.

17 **[00160]** In one embodiment, the information received from the recruiter may be
18 used to create a recruiter profile, which may include various job postings the recruiter is
19 looking to fill. In some implementations, the recruiter may be looking to fill job
20 postings for a particular company and may create a company profile. Company profiles
21 may include a summary about the company and/or a listing of available jobs at the
22 company. Some implementations may also indicate if a user viewing the company
23 profile has any connections who work (or, in some embodiments, have worked) for the
24 company. A sample screen shot may be seen in Fig. 3E.

25 **[00161]** In other examples, donors, charitable organizations, advertisers, target
26 audiences, self forming groups, education providers, and/or the like may wish to join
27 the people network 312 and/or the companies/organizations network 314. In one
28 implementation, a user may provide a platform join request associated with the user's
29 classification. For example, a user classified as a candidate may provide the candidate
30 platform join request 353 and a user that is classified as a recruiter may provide the
31 recruiter platform join request 355. In another implementation, a user may provide a
32 platform join request that is independent of the user's classification. For example, the
33 candidate platform join request 353 and the recruiter platform join request 355 may be
34 the same type of request, and whether a user is a candidate, a recruiter, or both, may

1 depend on the provided information (e.g., if the user is currently seeking employment
2 the user may be classified as a candidate, if the user posts any jobs and/or if the user's
3 company has available positions the user may be classified as a recruiter).

4 **[00162]** The SMP 305 may obtain social network information 351a-351c from
5 social networks 307a-307c. Social network information may include data regarding a
6 user's contacts at various social networking websites (e.g., Facebook, LinkedIn,
7 MySpace, Orkut, Friendster, Twitter, and/or the like), data regarding a user's contacts at
8 various messaging application (e.g., AIM, Skype, Yahoo! Messenger, Google Talk,
9 Facetime, and/or the like), data regarding a user's email contacts, data regarding a
10 user's phone contacts, and/or the like. Social network information may also include data
11 regarding a user's affiliations with companies (e.g., previous and/or current employers,
12 company groups, and/or the like) and/or organizations (e.g., previous and/or current
13 universities, clubs, charities, and/or the like), a user's geographic location, a user's likes
14 and/or dislikes, endorsements of a user and/or of the user's work, languages a user
15 knows, news and/or social media information regarding a user and/or a company
16 and/or organization associated with the user, and/or the like. For example, social
17 network information 351a-351c may be in XML format substantially in the following
18 form:

```
19 <XML>
20   <Contacts>List of user's contacts</Contacts>
21   <Affiliations>
22     <Affiliation1>Company1</Affiliation1>
23     <Affiliation2>University1</Affiliation2>
24     <Affiliation3>Club1</Affiliation3>
25   </Affiliations>
26   <Location>New York, NY</Location>
27   <Endorsements>
28     <Endorsement1>Endorsement of user</Endorsement1>
29     <Endorsement2>Endorsement of user's work</Endorsement2>
30   </Endorsements>
31   <Languages>
32     <Language1>Language1 - fluent</Language1>
33     <Language2>Language2 - proficient</Language2>
34   </Languages>
35 </XML>
```

36
37 **[00163]** In one embodiment, information regarding users and/or regarding the
38 users' social networks, may be obtained via API calls to social networks 307a-307c and

1 used as a people network 312 and/or a companies/organizations network 314. In
2 another embodiment, the SMP 305 may use information regarding the users and/or
3 regarding the users' social networks to generate a people network 312 and/or a
4 companies/organizations network 314. Information regarding the users and/or
5 regarding the users' social networks may include information regarding how people are
6 connected to each other and/or to companies and/or organizations, profile information
7 associated with the users, profile information associated with companies and/or
8 organizations, and/or the like. In some implementations, this information may be used
9 to gather additional information about candidates and suggest profile updates for users
10 if the user has not updated his or her information or if additional information indicates
11 that some information is missing from his/her profile. For example, information
12 regarding the people network 312 and/or the companies/organizations network 314,
13 SMP network info 357, may include consolidated explicit and/or implicit information
14 from a variety of sources (e.g., user provided information, information from various
15 social networks, and/or the like). Explicit information may include profile information,
16 location information, preference information, and/or the like. Implicit information may
17 include user classification (e.g., a college applicant) inferred from user provided
18 information, skill level rating inferred from news and/or social media information
19 regarding the user, group membership inferred from provided data (e.g., NY Doctors
20 group for a user who finished medical school and is located in NY), and/or the like. For
21 example, the SMP network info 357 may be in XML format substantially in the following
22 form:

```
23 <UserProfile>
24   <PersonalInfo>
25     <Name>John Doe</Name>
26     <ScreenName>ScreenName1</ScreenName>
27   </PersonalInfo>
28   <ContactInfo>
29     <StreetAddress>123 Main Street, New York, NY</StreetAddress>
30     <Phone>(333)333-3333</Phone>
31   </ContactInfo>
32   <JobsInfo>
33     <Job1>Title1, prerequisites1, salary info1</Job1>
34     <Job2>Title2, prerequisites2, salary info2</Job2>
35   </JobsInfo>
36   <Experience>
37     <Position1>Job Title1, Experience1</Position1>
38     <Position2>Job Title2, Experience2</Position2>
```

```

1         <Position3>Job Title3, Experience3</Position3>
2     </Experience>
3     <Education>
4         <Education1>University1, Degree1</Education1>
5         <Education2>University2, Degree2</Education2>
6     </Education>
7     <SocialMeetupPreferences>
8         <Preference>Allow all to request social meetup</Preference>
9     </SocialMeetupPreferences>
10    <Contacts>
11        Consolidated list of user's contacts from various social networks
12    </Contacts>
13    <Affiliations>
14        <Affiliation1>Company1</Affiliation1>
15        <Affiliation2>University1</Affiliation2>
16        <Affiliation3>Club1</Affiliation3>
17    </Affiliations>
18    <Location>New York, NY</Location>
19    <Endorsements>
20        <Endorsement1>Endorsement of user</Endorsement1>
21        <Endorsement2>Endorsement of user's work</Endorsement2>
22    </Endorsements>
23    <Languages>
24        <Language1>Language1 - fluent</Language1>
25        <Language2>Language2 - proficient</Language2>
26    </Languages>
27 </UserProfile>
28
29
30 <CompanyOrganizationProfile>
31     <Name>Company1</Name>
32     <Location>New York, NY</Location>
33     <Industry>Software</Industry> - e.g., companies may be related by industry
34     <Members>List of members</Members>
35     <Media>Photos and videos regarding the company</Media>
36 </CompanyOrganizationProfile>

```

37

38 **[00164]** The SMP 305 may provide various types of information including job
39 information, candidate information, education information, offer information,
40 information regarding companies, organizations and/or groups, relevant contacts
41 information, and/or the like to a user. For example, the SMP 305 may provide job
42 information 361 (e.g., as a result of a search, as an advertisement, and/or the like) to the
43 candidate 320. Such information may include a job recommendation, identification of
44 contacts that may be helpful to the candidate in obtaining the job, and/or the like, and
45 may be based on the SMP network info 357. In one embodiment, a recruiter may limit
46 who can see a job posted by the recruiter (e.g., friends (1st degree) and friends of friends
47 (2nd degree)), and job recommendations regarding the job may be limited accordingly.

1 In another embodiment, a job may be recommended to any candidate determined by
2 the SMP 305 to be relevant. For example, the job information 361 may be in XML
3 format substantially in the following form:

```
4 <XML>  
5   <PositionInfo>  
6     <Title>JobTitle1</Title>  
7     <Company>company name</Company>  
8     <Prerequisites>Skills, Education</Prerequisites>  
9     <Location>job location</Location>  
10    <Link>job page link</Link>  
11    <Contacts>List of relevant contacts</Contacts>  
12  </PositionInfo>  
13 </XML>
```

14
15 **[00165]** In another example, the SMP 305 may provide candidate information 363
16 to the recruiter 325. Such information may include a candidate recommendation,
17 identification of contacts that may be helpful to the recruiter in obtaining additional
18 information regarding the candidate, and/or the like, and may be based on the SMP
19 network info 357. In one embodiment, a candidate may limit who (e.g., only 1st degree
20 contacts) and/or what kinds of information (e.g., experience and education, but not
21 current location) may be viewed by a recruiter, and candidate recommendations and/or
22 candidate information may be limited accordingly. In some implementations, the
23 candidate may specify how (s)he would like to be contacted by a recruiter, and this may
24 be based on degrees of separation from a recruiter. For example, a first degree contact
25 may be able to ping the candidate via a instant messaging service over the SMP, whereas
26 instant messaging may be disabled for a second (or higher) degree contact. In another
27 embodiment, any candidate may be recommended if determined by the SMP 305 to be
28 relevant. For example, the candidate information 363 may be in XML format
29 substantially in the following form:

```
30 <XML>  
31   <CandidateInfo>  
32     <Name>candidate name</Name>  
33     <Title>current job title</Title>  
34     <Experience>Job Title1, Experience1</Experience>  
35     <Education>University1, Degree1</Education>  
36     <Contacts>List of relevant contacts</Contacts>  
37   </CandidateInfo>  
38 </XML>
```

39

1 **[00166]** If SMP members may be helpful to each other, and if their social meetup
2 preference settings allow this, the SMP 305 may facilitate a social meetup 330 via social
3 meetup info 365a-c. For example, a representative of a company that provides
4 ergonomic keyboards may use a social meetup to interact with a target group of software
5 developers (e.g., database engineers of company X) interested in ergonomic products
6 and/or other software developer groups (e.g., application engineers of company X)
7 associated (e.g., both groups are at company X) with the target group. In another
8 example, a recruiter that has a job for which a candidate may be a good match may
9 interact with the candidate. Another example may provide a recruiter with information
10 of a common contact for a candidate of interest. That is, if the recruiter finds a second
11 degree contact (s)he believes to be a good candidate for a job posting, the SMP may
12 provide the recruiter with the contact information for the recruiter's first degree contact
13 such that the first degree contact assists in connecting the recruiter with the candidate.

14 **[00167]** In some SMP embodiments, professionally connected users may be able to
15 assist other users by recommending a job posting to a candidate or a candidate to a job
16 posting. For example, if a first user, who is professionally connected to a second user
17 via the SMP, sees a job posting that the second user may be interested in, the first user
18 may recommend the job posting to the second user. In some implementations, the job
19 posting may be for a job at the first user's company. In one embodiment, the first user
20 may receive a reward (e.g., award money) for referring the second user if the second
21 user applies and/or gets an offer for the recommended job listing. In some
22 embodiments, the SMP may provide a link comprising jobs for which the first user's
23 friends may wish to apply. This list of jobs for friends may list the job posting, its
24 location, suggested connections for a particular job, and the award amount if the
25 suggested user gets the job. An example screenshot is shown in Fig. 3F.

26 **[00168]** In one embodiment, social meetup info 365a-c may include names (e.g., of
27 the candidate and/or of the recruiter), venue preferences (e.g., face-to-face, phone
28 conference, chat room, and/or the like), length preferences (e.g., meet for half an hour),
29 offer details, product description, candidate resume, additional job description, and/or
30 the like.

1 **[00169]** FIGURE 3G shows an exemplary join request data flow in one
2 embodiment of the SMP. A candidate 320, recruiter 325 or a company 330 may request
3 331 to join the SMP. In some embodiments, this may be a request to join the SMP
4 Facebook app and may be instantiated by clicking a "join now" button on Facebook. The
5 join request may include a user name and password for the SMP, as well as user names
6 and passwords for each social network the user wishes to associate with his/her SMP
7 account, and/or the like. The join request may also include the user's user name and
8 password for various social networking websites. This information is received by the
9 Social Match Platform 305, which then may send a social network data graph request
10 333 to the user's social networking sites 307a, 307b and 307c. The data graph response
11 335 may then be sent back to the SMP, which in turn may use the response to create a
12 profile or user page for the user 337. This profile may be stored in the disk 399. In
13 some embodiments, the SMP may create a graph path for the user 339. This is
14 explained in greater detail in the "Advancement Path Taxonomy" section below. This
15 graph may also be stored 341 in the disk 399. Some implementations may use the graph
16 path to determine next potential career moves for a candidate and this information may
17 be used to serve the candidate potential jobs. For a recruiter or a company seeking
18 candidates, this graph path may be generated to determine what type of jobs potential
19 candidates may be coming from, and therefore the SMP may suggest candidates most
20 likely to be seeking the jobs the recruiter and/or company are offering.

21 **[00170]** FIGURE 3H shows an exemplary search data flow embodiment of the
22 SMP. In one example, a candidate 320 may submit a job search request 345 to the SMP
23 305, for example, by searching for "programmer" "NYC." The SMP may then generate a
24 job query 347 and search jobs 349. The query results are analyzed 350 to determine
25 which jobs may be most relevant to the user. This determination may include
26 information such as whether the user has contacts at the company offering the job, and
27 this may be weighted in determining the list of search results and/or the order of the
28 search results to display to the user. This is explained in more detail below in the
29 description for Fig. 5A, as a Job Relevancy value may be calculated. The query results
30 analysis 350 may also utilize information such as the location of the user's contacts to
31 determine where a user may be interested in looking for jobs. For example, a user living

1 in Washington, D.C., may have several contacts living in San Francisco, CA, and the
2 query result analysis may determine that the user may be interested in jobs located in
3 San Francisco, CA, even if the user has not specified this in the query. In other
4 implementations, the SMP may determine that job query results included jobs at a of a
5 particular sort or at a particular company (for example, a programmer at Microsoft),
6 and while the user may not have contacts at that particular company, there are other job
7 postings for similar jobs (e.g., a programmer at IBM) where the user does have contacts.
8 Once the SMP performs the analysis 350, the SMP may generate an updated query 352
9 based on social factors that may have been analyzed at 350. In the last example, for
10 instance, the SMP may update the query results to include the jobs at IBM.

11 **[00171]** During this process, the SMP may also determine other users that the
12 contact may know. In one implementation, the user may be connected to a second user
13 on a social networking site and the user may be seeking a job where the second user is
14 employed or the second user may be employed in a similar field as the user. In some
15 implementations, the SMP may be able to determine how much interaction the user has
16 with the second user on the other social networking sites. For example, if the user is
17 connected to the second user on the social networking site and communicates with the
18 second user regularly, this may be a strong recommendation, whereas if the user is
19 connected to the second user but there has been limited communication between them,
20 the SMP may provide a weaker recommendation. In some embodiments, this may be
21 determined by examining the contact between the users on several social networking
22 sites.

23 **[00172]** The user may determine that (s)he wishes to add and/or follow a new
24 contact. In some implementations, this may be a contact the SMP recommended; in
25 other implementations, the user may have found the contact via other means, e.g., a
26 name search. The user may request to add and/or follow the new contact 354. The SMP
27 may receive this request and send a group selection confirmation request 356 to the
28 contact with whom the user is looking to connect. The contact may then determine
29 whether (s)he wishes to connect with the user and send a group selection response 358
30 to the SMP. In some implementations, in order for the connection to be made between
31 the user and the contact, the contact may need to approve the connection. In alternative

1 embodiments, the user may be able to view the actions of the contact without contact
2 approval, essentially creating a one-way connection.

3 **[00173]** FIGURE 3I shows a data flow diagram illustrating various updating
4 embodiments of the SMP. The SMP may receive various inputs 370 from users of the
5 SMP. For example, a candidate 320 may provide a resume to the SMP and/or may post
6 information on his/her profile page. A recruiter or a company may post jobs to the
7 SMP. Yet another source of input to the SMP may be the Career Advertisement Network
8 ("CAN") or Career Pathing server(s) 315. The Career Advertisement Network is
9 described in more detail below in the section titled "Advertisement Generation,
10 Selection And Distribution System Registration" and Career Pathing is discussed in
11 more detail in the section titled "Advancement Path Taxonomy." These servers may
12 provide input relevant to the relationship between the candidates and the job postings.
13 The SMP also may receive input 372 from user postings and user activity on various
14 social networks 307a-c. The SMP may use these inputs to perform an analysis of the
15 updates 375 and may determine that based on the new information, the SMP may need
16 to utilize the CAN/Pathing Server(s) 315 to update job postings being served to a
17 particular user. In one example, a candidate may post on a social networking site that
18 (s)he is moving to a new city. The SMP may then use this information to serve the
19 candidate job postings in the new city. In another example, a user may post that (s)he is
20 being promoted. This may be used by the Pathing server to determine a potential new
21 career path for the candidate, and the SMP may then serve different job postings to the
22 candidate based on the new information. This new information may be updated and
23 stored in the SMP 377.

24 **[00174]** FIGURES 110A-B show logic flow diagrams illustrating examples of
25 transforming user identification and profile data via a Cross-Network Social Graph
26 Updating (CN-SGU) component into cross-network user social graph data. With
27 reference to FIGURE 110A, in some embodiments, a server within the SMP may obtain a
28 trigger for cross-network social graph generation, e.g., 11001. The server may parse the
29 trigger, and extract an identifier of a user for whom the cross-network social graph is to
30 be generated, e.g., 11002. The server may query a database for a profile of the user,
31 based on the extracted user ID, e.g., 11003. For example, the server may issue PHP/SQL

1 commands to query a database table (such as FIGURE 99, Users 9919a) for user profile
 2 data. An example user profile data query 11003, substantially in the form of PHP/SQL
 3 commands, is provided below:

```

4  <?PHP
5  header('Content-Type: text/plain');
6  mysql_connect("254.93.179.112", $DBserver, $password); // access database server
7  mysql_select_db("SMP_DB.SQL"); // select database table to search
8  //create query
9  $query = "SELECT network_id_list network_name_list login_secure_list FROM
10 UsersTable WHERE userID LIKE '%" $user_id";
11 $result = mysql_query($query); // perform the search query
12 mysql_close("SMP_DB.SQL"); // close database access
13 ?>
14

```

15 **[0001]** In some embodiments, the server may parse the results of the query, and
 16 may extract the identity of the social networks of which the user is a member; and may
 17 obtain the user's credential(s) for those social network(s), e.g., 11004. For example, the
 18 server may utilize parsers such as the example parser discussed below in the description
 19 with respect to computer systemization FIGURE 99. The server may select a social
 20 network for information mining, e.g., 11005. The server may generate an application
 21 programming interface ("API") call to the social networking server, e.g., 11006. In some
 22 embodiments, where the server does not have access to the user's login credentials, the
 23 server may submit a request to a user of the SMP to login to the social networking
 24 service to provide the server access to the user's social data. For example, the server
 25 may provide an HTML page to a client of the user including authentication commands
 26 similar to the exemplary illustrative listing provided below:

```

27  <html>
28  <div id="fb-root"></div>
29  <script src="http://connect.facebook.net/en_US/all.js"></script>
30  <script>
31    FB.init({appId: 'A3BFE5', status: true, cookie: true, xfbml: true});
32    FB.Event.subscribe('auth.sessionChange', function(response) {
33      if (response.session) {
34        // A user has logged in, and a new cookie has been saved
35      } else {
36        // The user has logged out, and the cookie has been cleared
37      }
38    });
39  </script>
40  </html>
41

```

1 **[0002]** The server may then generate and provide a request for social data
 2 including, but not limited to: user ID, friend ID(s), friend relationship strength(s), social
 3 activity timestamp(s), message ID(s), message(s), and/or the like. For example, the
 4 load balancing server may execute PHP commands similar to those in the exemplary
 5 illustrative listing provided below:

```

6 <?PHP
7 header('Content-Type: text/plain');
8 // Obtain user ID(s) of friends of the logged-in user
9 $friends = json_decode(file_get_contents(
10     'https://graph.facebook.com/me/friends?access_token=' .
11     $cookie['oauth_access_token']), true);
12 $friend_ids = array_keys($friends);
13
14 // Obtain message feed associated with the profile of the logged-in user
15 $feed = json_decode(file_get_contents(
16     'https://graph.facebook.com/me/feed?access_token=' .
17     $cookie['oauth_access_token']), true);
18
19 // Obtain messages by the logged-in user's friends
20 $result = mysql_query('SELECT * FROM content WHERE uid IN (' .
21     implode($friend_ids, ',') . ')');
22 $friend_content = array();
23 while ($row = mysql_fetch_assoc($result)) {
24     $friend_content[] = $row;
25 }
26

```

27 **[0003]** If the API call is successful, e.g., 11007, option "Yes," in response, a social
 28 networking server may provide the requested information, e.g., 11010. For example, the
 29 social networking server may provide a JavaScript Object Notation format ("JSON")-
 30 encoded data structure embodying the requested information. An exemplary JSON-
 31 encoded data structure embodying social data (e.g., user ID(s) of friends of the logged-in
 32 user) is provided below:

```

33 { "data": [
34     { "name": "Tabatha Orloff",
35       "id": "483722"},
36     { "name": "Darren Kinnaman",
37       "id": "865743"},
38     { "name": "Sharron Jutras",
39       "id": "091274"}
40 ]}
41

```

42 **[00175]** If the API call was not successful (e.g., 11007, option "No"), and the API
 43 call has been tried a timeout threshold number of times (see 11008), the server may

1 cease attempting to obtain user social data from the selected social network (see 11009).
 2 In some embodiments, the server may aggregate social data from each accessible social
 3 networking service into an aggregated cross-network user social data set, 11011, (e.g.,
 4 including the user's original user profile from which the user's social networks were
 5 identified at 11004).

6 **[00176]** In some embodiments, the server may parse the aggregated user social
 7 data set, and extract the data fields (and, e.g., their associated data values), from the
 8 cross-network aggregated social data set, e.g., 11012. For example, the server may
 9 utilize parsers such as the example parser discussed below in the description with
 10 respect to computer systemization FIGURE 99. The server may query a database for
 11 social graph weight selection rules, e.g., 11013, and relationship type identification rules,
 12 e.g., 11014, for user cross-network social graph generation/updating. For example, the
 13 server may issue PHP/SQL commands to query a database table (such as FIGURE 99,
 14 Social Graph Generation Rules Table 9919z) for social graph weight selection and
 15 relationship type identification rules ("social graph generation rules"). An example
 16 social graph generation rules query, substantially in the form of PHP/SQL commands, is
 17 provided below:

```
18 <?PHP
19 header('Content-Type: text/plain');
20 mysql_connect("254.93.179.112", $DBserver, $password); // access database server
21 mysql_select_db("SMP_DB.SQL"); // select database table to search
22 //create query
23 $query = "SELECT rule_id rule_listing rule_expiry rule_input_params_list
24 rule_output_params_list, API_call_list FROM Social Graph GenRulesTable WHERE
25 rule_type = "Social Graph Gen";
26 $result = mysql_query($query); // perform the search query
27 mysql_close("SMP_DB.SQL"); // close database access
28 ?>
```

29
 30 **[00177]** In some embodiments, the database may provide the selected rules.
 31 Examples of such rules are illustrated below in XML form:

```
32 <social_graph_weight_selection_rule>
33 <IF>field_source = OpenSocial graph JSON</IF>
34 <THEN> weight = 1 </THEN>
35 <ELSE> continue </ELSE>
36 </social_graph_weight_selection_rule>
37
38 <relationship_type_identification_rule>
39 <IF>field_source = OpenSocial GRAPH JSON</IF>
```



```

1     <THEN> type = OpenSocial TYPE </THEN>
2     <ELSE> continue </ELSE>
3     </relationship_type_identification_rule>
4

```

5 **[00178]** With reference to FIGURE 110B, in some embodiments, the server may
6 select a data field from the aggregated social data set, e.g., 11015. The server may
7 determine a user-related entity associated with the selected data field (e.g., by searching
8 for user identities among other data fields associated with the selected data field), e.g.,
9 11016. The server may identify a relationship type using the relationship type
10 identification rules, e.g., 11017. For example, the server may apply each rule, and for
11 each rule that is satisfied, a new entity may be created as being related to the user, and
12 provided with an entity ID. For each identified entity ID, the server may calculate a
13 relationship weight using the social graph weight selection rules, e.g., 11018. The server
14 may generate (an updated) relationship score for each user-entity relationship as a
15 branch weight for a branch in a cross-network user social graph, e.g., 11019. For
16 example, if via two separate rules, scores are generated for the same user-entity branch
17 in the cross-network user social graph, those scores may be added to provide a total
18 branch strength. The server may perform such a branch identification, relationship type
19 identification and score calculation for each branch in the cross-network user social
20 graph (see 11020). The server may then provide the user-entity branch listing, and
21 relationship scores as cross-network user social graph data, e.g., 11021.

22 **[00179]** FIGURE 111 shows a logic flow diagram illustrating examples of
23 transforming user identification and profile data via a Cross-Network User Profile
24 Sensitive Query Generation (CN-UPSQG) component into cross-network user profile
25 sensitive search results. In some embodiments, a server within the SMP may obtain a
26 trigger for search query generation, e.g., 11101. The server may parse the trigger, and
27 extract an identifier of a user for whom to generate the query, e.g., 11102. The server
28 may query a database for a profile of the user, based on the extracted user ID, e.g.,
29 11103. For example, the server may issue PHP/SQL commands to query a database
30 table (such as FIGURE 99, Users 9919a) for user profile data. An example user profile
31 data query 11103, substantially in the form of PHP/SQL commands, is provided below:

```

32     <?PHP
33     header('Content-Type: text/plain');

```

```

1  mysql_connect("254.93.179.112",$DBserver,$password); // access database server
2  mysql_select_db("SMP_DB.SQL"); // select database table to search
3  //create query
4  $query = "SELECT network_id_list network_name_list login_secure_list FROM
5  UsersTable WHERE userID LIKE '%' $user_id";
6  $result = mysql_query($query); // perform the search query
7  mysql_close("SMP_DB.SQL"); // close database access
8  ?>
9

```

10 **[0004]** In some embodiments, the server may parse the results of the query, and
11 may extract the identity of the social networks of which the user is a member; and may
12 obtain the user's credential(s) for those social network(s), e.g., 11104. For example, the
13 server may utilize parsers such as the example parser discussed below in the description
14 with respect to computer systemization FIGURE 99. The server may select a social
15 network for information mining, e.g., 11105. The server may generate an application
16 programming interface ("API") call to the social networking server, e.g., 11106. In some
17 embodiments, where the server does not have access to the user's login credentials, the
18 server may submit a request to a user of the SMP to login to the social networking
19 service to provide the server access to the user's social data. For example, the server
20 may provide an HTML page to a client of the user including authentication commands
21 similar to the exemplary illustrative listing provided below:

```

22  <html>
23  <div id="fb-root"></div>
24  <script src="http://connect.facebook.net/en_US/all.js"></script>
25  <script>
26    FB.init({appId: 'A3BFE5', status: true, cookie: true, xfbml: true});
27    FB.Event.subscribe('auth.sessionChange', function(response) {
28      if (response.session) {
29        // A user has logged in, and a new cookie has been saved
30      } else {
31        // The user has logged out, and the cookie has been cleared
32      }
33    });
34  </script>
35  </html>
36

```

37 **[0005]** The server may then generate and provide a request for social data
38 including, but not limited to: user ID, friend ID(s), friend relationship strength(s), social
39 activity timestamp(s), message ID(s), message(s), and/or the like. For example, the
40 load balancing server may execute PHP commands similar to those in the exemplary
41 illustrative listing provided below:

```

1 <?PHP
2 header('Content-Type: text/plain');
3 // Obtain user ID(s) of friends of the logged-in user
4 $friends = json_decode(file_get_contents(
5     'https://graph.facebook.com/me/friends?access_token=' .
6     $cookie['oauth_access_token']), true);
7 $friend_ids = array_keys($friends);
8
9 // Obtain message feed associated with the profile of the logged-in user
10 $feed = json_decode(file_get_contents(
11     'https://graph.facebook.com/me/feed?access_token=' .
12     $cookie['oauth_access_token']), true);
13
14 // Obtain messages by the logged-in user's friends
15 $result = mysql_query('SELECT * FROM content WHERE uid IN (' .
16     implode($friend_ids, ',') . ')');
17 $friend_content = array();
18 while ($row = mysql_fetch_assoc($result)) {
19     $friend_content[] = $row;
20 }
21

```

22 **[0006]** If the API call is successful, e.g., 11107, option "Yes," in response, a social
23 networking server may provide the requested information, e.g., 11110. For example, the
24 social networking server may provide a JavaScript Object Notation format ("JSON")-
25 encoded data structure embodying the requested information. An exemplary JSON-
26 encoded data structure embodying social data (e.g., user ID(s) of friends of the logged-in
27 user) is provided below:

```

28 { "data": [
29     { "name": "Tabatha Orloff",
30       "id": "483722"},
31     { "name": "Darren Kinnaman",
32       "id": "865743"},
33     { "name": "Sharron Jutras",
34       "id": "091274"}
35 ]}
36

```

37 **[00180]** If the API call was not successful (e.g., 11107, option "No"), and the API
38 call has been tried a timeout threshold number of times (see 11108), the server may
39 cease attempting to obtain user social data from the selected social network (see 11109).
40 In some embodiments, the server may aggregate social data from each accessible social
41 networking service into an aggregated cross-network user social data set, 11111, (e.g.,
42 including the user's original user profile from which the user's social networks were
43 identified at 11104).

1 **[00181]** In some embodiments, the server may parse the aggregated user social
 2 data set, and extract the data fields (and, e.g., their associated data values), from the
 3 cross-network aggregated social data set, e.g., 11112. For example, the server may utilize
 4 parsers such as the example parser discussed below in the description with respect to
 5 computer systemization FIGURE 99. The server may query a database for data value
 6 filtration rules, e.g., 11113, for query keyword set generation. For example, the server
 7 may issue PHP/SQL commands to query a database table (such as FIGURE 99,
 8 Keyword Generation Rules Table 9919z) for data value filtration rules. An example data
 9 value filtration rules query, substantially in the form of PHP/SQL commands, is
 10 provided below:

```
11 <?PHP
12 header('Content-Type: text/plain');
13 mysql_connect("254.93.179.112", $DBserver, $password); // access database server
14 mysql_select_db("SMP_DB.SQL"); // select database table to search
15 //create query
16 $query = "SELECT rule_id rule_listing rule_expiry rule_input_params_list
17 rule_output_params_list, API_call_list FROM KeywordGenRulesTable WHERE
18 rule_type = "KeywordGen";
19 $result = mysql_query($query); // perform the search query
20 mysql_close("SMP_DB.SQL"); // close database access
21 ?>
```

22 **[00182]** In some embodiments, the database may provide the selected rules.
 24 Examples of such rules are illustrated below in XML form:

```
25 <data_value_filtration_rule>
26 <IF>field_source = OpenSocial graph JSON</IF>
27 <THEN> include; stop other rules </THEN>
28 <ELSE> continue </ELSE>
29 </data_value_filtration_rule>
```

30
 31 **[00183]** The server may filter the extracted data fields using filtration rules to
 32 generate the query keyword set, e.g., 11114. The server may then generate a database
 33 query (or queries, e.g., depending on the number and/or relatedness of keywords), using
 34 the generated query keyword set, e.g., 11115. The server may provide the generated
 35 database queries as cross-network user profile sensitive queries for submission to
 36 database(s) for cross-network user social data-relevant search results.

37 **[00184]** FIGURE 4 shows a logic flow diagram illustrating a Network Join (NJ)
 38 component in one embodiment of the SMP. In Figure 4, a request to join the SMP may

1 be received at 405. As described with regard to Figure 3, such a request 353, 355 may be
2 received from a user (e.g., from a candidate, a recruiter, an applicant, a company, an
3 organization, a group, and/or the like).

4 **[00185]** A determination may be made at 410 whether there are social data sources
5 from which the SMP should obtain social data, location data, news and social media
6 data, and/or the like. For example, upon joining a user may specify that the SMP should
7 obtain information from one or more social networking sites, such as Facebook and
8 LinkedIn. If there is a target social data source, the SMP may obtain access to such
9 social data source at 415. In one embodiment, the user may provide login credentials to
10 the SMP that may be used to access the social data source (e.g., LinkedIn username and
11 password). In another embodiment, installing an application on a social network (e.g., a
12 Facebook application) may also result in the user authorizing the SMP to access their
13 data on the social network (e.g., Facebook data).

14 **[00186]** The SMP may obtain social data, location data, news and social media
15 data, and/or the like from the social data source at 420. In one embodiment, such data
16 may be obtained (e.g., via a Facebook API call) upon use. In another embodiment, such
17 data may be obtained (e.g., via a Facebook API call) and stored by the SMP. In yet
18 another embodiment, the SMP may obtain such data and consolidate it with data from
19 other social sources. For example, the SMP may obtain education data from Facebook,
20 work experience data from LinkedIn, and create a profile for the user that includes both
21 types of data (e.g., as described with regard to SMP network info 357). In some
22 implementations, this may be used to compile a SMP profile for the candidate.

23 **[00187]** The user's affiliations with companies/organizations and/or people may
24 be determined at 425. Affiliations may include companies, organizations, universities,
25 groups, clubs, people, and/or the like with which the user was previously and/or is
26 currently associated. In one implementation, affiliations may be determined by
27 examining the Affiliations field of the UserProfile data structure described with regard
28 to SMP network info 357. In another implementation, affiliates may be inferred based
29 on analysis of information regarding the user. For example, affiliates may include other
30 users that are members of a group in which the user is a member, users that have

1 similar education, skills, location, preferences, career paths, and/or the like as the user,
2 and/or the like.

3 **[00188]** In some implementations, affiliations may be determined 425 by
4 examining relationships between users. For example, a first user may indicate that
5 (s)he went to school with (or worked with, etc.) a second user. For instances where the
6 second user education information is not available, the SMP may associate the
7 education information from the first user with that of the second user. The second user
8 may then approve, claim or edit the education information provided by the first user,
9 thereby incorporating the information into their (the second user's) SMP profile.

10 **[00189]** A determination may be made at 430 whether the user's affiliations
11 include company/organization affiliates that have not previously been presented to the
12 SMP (e.g., based on the names of the affiliates). If such new affiliates are included,
13 information regarding these affiliates may be stored at 435 (e.g., as described with
14 regard to the CompanyOrganizationProfile data structure described with regard to SMP
15 network info 357), and company/organization pages may be created for the new
16 affiliates. For example, if the user works at Microsoft and is the first user to indicate an
17 affiliation with Microsoft, a new page for Microsoft may be created by the SMP. Such
18 pages may provide information regarding the company/organization (e.g., name,
19 number of affiliated users on the SMP), media information (e.g., user provided photos,
20 videos, and/or the like), a listing of affiliated users on the SMP, and/or the like.
21 Furthermore, such pages may be linked to other pages (e.g., groups within a company
22 may be linked to the company) and/or consolidated (e.g., Microsoft and Microsoft Inc.
23 may be consolidated under one page) by a company/organization representative and/or
24 by the SMP (e.g., based on a similarity analysis of names).

25 **[00190]** The user's social information (e.g., including social network data, location
26 data, news and social media data, and/or the like) and/or affiliations data may be
27 analyzed at 440. For example, social data of the user's contacts may be analyzed to
28 discern other users that are contacts of a significant number (e.g., 5 contacts, 20% of
29 contacts) of the user's contacts and that share similar location, education, skills, and/or
30 the like. Such other users may be suggested to the user at 445 as potential new contacts.
31 In another example, the user's affiliations may be analyzed to discern other users that

1 share a significant number (e.g., 5 affiliations, 20% of affiliations) of the user's
2 affiliations. In some embodiments, users may classify their contacts by, for example,
3 providing an indication of how they know other users of the SMP. For example, a first
4 user may indicate that (s)he went to school with a second user at a certain educational
5 institution, and the second user has indicated that (s)he attended school with a third
6 user, the third user may be suggested to the first user as a potential new contact. As
7 such, a list of "people you may know" may be provided to each user. In this light, other
8 users (e.g., people, companies/organizations, and/or the like) may be suggested to the
9 user at 445 as potential new contacts.

10 **[00191]** FIGURE 5A shows a logic flow diagram illustrating a Job Info Providing
11 (JIP) component in one embodiment of the SMP. The JIP component may be used to
12 provide job information to a candidate and/or to facilitate social meetup with the
13 candidate. In Figure 5A, a request to provide job information may be received at 505. In
14 one embodiment, such a job information request may be received as a result of a job
15 search query initiated by a candidate. In another embodiment, such a job information
16 request may be a result of determining which job to present to a candidate in an
17 advertisement. For example, the job information request may include information such
18 as the candidate's unique ID, a job title of a desired job (e.g., based on the candidate's
19 last job title), job location, and/or the like, and may be in XML format substantially in
20 the following form:

```
21 <XML>  
22   <JobInfoRequest>  
23     <UserID>Candidate's unique ID</UserID>  
24     <Title>JobTitle1</Title>  
25     <Location>New York, NY</Location>  
26   </JobInfoRequest>  
27 </XML>
```

28
29 **[00192]** The candidate's job skills, affiliations and contacts may be determined at
30 510, 515, and 520, respectively. In one embodiment, such information may be retrieved
31 based on the candidate's unique ID (UID) via an API call to a social data source (e.g., via
32 Facebook API calls). For example, an API call to determine the candidate's contacts
33 from Facebook may be written in JavaScript substantially in the following form:

```
34 FB.Data.query('SELECT flid FROM friendlist WHERE owner=FacebookUniqueID');  
35
```

1 **[00193]** In another embodiment, such information may be retrieved from the data
 2 stored by the SMP. For example, the candidate's affiliations may be determined via a
 3 SQL query substantially in the following form:

```
4 SELECT affiliations FROM UserInfo WHERE uid='Candidate's Unique ID'
```

5
 6 **[00194]** The candidate's job skills, affiliations, contacts, endorsements,
 7 recommendations, and/or the like may be analyzed to determine relevant jobs at 525. In
 8 one embodiment, relevant jobs may be determined based on the number of the
 9 candidate's connections (e.g., 1st degree contacts, 2nd degree contacts, and/or the like)
 10 with the recruiter posting the job and/or other users affiliated with the company
 11 offering the job, and/or based on the candidate's skills. For example, job relevancy may
 12 determined according to a formula substantially in the following form:

```
13 Job Relevancy = (10 * # of recruiters) + (3 * # of 1st degree contacts at
14 company) + (0.5 * # of 2nd degree contacts) + (10 * % of
15 desired job skills that the candidate has)
```

16
 17 Accordingly, the job relevancy in this example is determined as the sum of: (1) ten
 18 multiplied by the number of recruiters who posted the job who are 1st degree contacts of
 19 the candidate, (2) three multiplied by the number of first degree contacts affiliated with
 20 the company offering the job (3) one half multiplied by the number of second degree
 21 contacts who are affiliated with the company, and (4) ten multiplied by the percentage
 22 of desired skills listed for the job that the candidate has. In another embodiment,
 23 relevant jobs may be determined based on the match between candidate's skills and/or
 24 education and/or location and job prerequisites, and ordered for presentation to the
 25 candidate based on the job relevancy determined using social data (e.g., using the job
 26 relevancy formula above). In additional embodiments, social interactions and opinion
 27 makers, as described in the Connecting Internet Users section, may affect job relevancy.
 28 For example, contacts who are identified as connectors may be weighted higher than
 29 other contacts when calculating the job relevancy value. In alternative embodiments,
 30 social data may affect the choice of paths determined by CSE and/or APT, as described
 31 in the Advancement Path Taxonomy section. In yet other embodiments, the candidate's
 32 career path to date may affect job relevancy.

1 **[00195]** Information regarding relevant jobs may be provided to the candidate at
2 530. For example, in response to a search query, the candidate may be presented with
3 ten most relevant jobs as determined by the job relevancy formula. In another example,
4 a job advertisement presented to the candidate may be the most relevant job as
5 determined by the job relevancy formula (e.g., social data may affect the choice of
6 advertisements to present to the candidate using the Career Advertisement Network, as
7 described in Advertisement Generation, Selection And Distribution System Registration
8 section, Advertisement Generation section, Advertisement Targeting/Distribution
9 section, and Advertisement Evolution section). In some embodiments, such information
10 may be presented to the candidate using an interactive display box 2300 illustrated in
11 Fig. 23, as described in the Automated Online Data Submission section.

12 **[00196]** A determination may be made at 532 whether relevant contacts (e.g., a
13 recruiter, a recruiter's direct or indirect contacts, employees of the company posting the
14 job, and/or the like) for a relevant job are available for social meetup. For example,
15 interacting with the recruiter may improve the candidate's chances of getting the job
16 and may provide the recruiter with additional information regarding the candidate. In
17 another example, interacting with company employees may help the candidate assess
18 the culture of the company. In one implementation, such determination may be made by
19 examining the SocialMeetupPreferences field of the UserProfile data structure described
20 with regard to SMP network info 357. If one or more relevant contacts are available for
21 social meetup and the candidate desires to do so, the SMP may facilitate social meetup
22 with such relevant contacts at 534, as described in further detail in 535-542.

23 **[00197]** The SMP may obtain a meetup type from the candidate at 535. For
24 example, the candidate may select a meetup type (e.g., in-person, text chat, audio chat,
25 video chat, and/or the like) via a select box. If the candidate selects in-person meetup, a
26 suggested venue (e.g., a restaurant, a coffee shop) may be determined at 537. In one
27 embodiment, the suggested venue may be determined based on geographic proximity to
28 meetup participants (e.g., if the meetup participants are located in Manhattan, the
29 venue may be a restaurant in Manhattan). In another embodiment, the suggested venue
30 may be based on sponsorship (e.g., a venue may pay a fee to be on the list of venues that
31 may be suggested to the meetup participants). If the venue is not approved by the

1 meetup participants at 538, the next preferred venue may be selected and suggested to
2 the user. If the venue is approved, invites to schedule a meetup at a specified time (e.g.,
3 after 3 days) may be sent to the meetup participants at 539. For example, invites may be
4 sent via email, via Facebook events, via Google calendar, and/or the like. If the
5 candidate selects electronic meetup, a meetup time may be suggested at 540. For
6 example, the SMP may suggest a meetup right away (e.g., if the participants are
7 available), after a predetermined time (e.g., in half an hour or the following day), and/or
8 the like. If a determination is made at 541 that the meetup is not going to happen right
9 away, invites to schedule a meetup at a specified time may be sent to the meetup
10 participants at 539. If the meetup is going to happen right away (or the scheduled
11 meetup time has arrived), a chat client (e.g., text, audio, video and/or the like) may be
12 instantiated at 542 (as illustrated with regard to 542). In one embodiment, the chat
13 client may be instantiated by the SMP (e.g., on a webpage via a SMP chat application
14 written in Flash). In another embodiment, the chat client may be a third party
15 application (e.g., Skype, AIM, Facebook Chat, and/or the like) launched by the SMP.

16 **[00198]** In some implementations, if the candidate's preferred meetup type is an
17 in-person meetup, the SMP may determine the feasibility of an in-person meetup by, for
18 example, determining if the candidate and recruiter live in the same city or within a
19 certain mile radius. If the SMP detects that the candidate and recruiter are located in
20 distant cities, the SMP may suggest that an electronic meetup may be more feasible than
21 an in-person meetup. In this scenario, an option may be presented to both the
22 candidate and the recruiter and one or both parties may select to proceed either with the
23 in-person meet up or the suggested electronic meetup.

24 **[00199]** FIGURE 5B shows a logic flow diagram illustrating a Candidate Info
25 Providing (CIP) component in one embodiment of the SMP. The CIP component may be
26 used to provide candidate information to a recruiter and/or to facilitate social meetup
27 with the recruiter. In Figure 5B, a request to provide candidate information may be
28 received at 550. In one embodiment, such a candidate information request may be
29 received as a result of a candidate search query initiated by a recruiter. In another
30 embodiment, such a candidate information request may be a result of determining
31 which candidate to present to a recruiter in an advertisement. For example, the

1 candidate information request may include information such as the recruiter's unique
 2 ID, a job title (e.g., based on the job title of a job posted by the recruiter), job location,
 3 and/or the like, and may be in XML format substantially in the following form:

```

4 <XML>
5   <CandidateInfoRequest>
6     <UserID>Recruiter's unique ID</UserID>
7     <Title>JobTitle1</Title>
8     <Location>New York, NY</Location>
9   </CandidateInfoRequest>
10  </XML>

```

11
 12 **[00200]** The recruiter's posted jobs, affiliations and contacts may be determined at
 13 555, 560, and 565, respectively. In one embodiment, such information may be retrieved
 14 based on the recruiter's unique ID (UID) via an API call to a social data source (e.g., via
 15 Facebook API calls). For example, an API call to determine the recruiter's contacts from
 16 Facebook may be written in JavaScript substantially in the following form:

```

17 FB.Data.query('SELECT flid FROM friendlist WHERE owner=FacebookUniqueID');
18

```

19 **[00201]** In another embodiment, the recruiter's posted jobs may be retrieved via an
 20 API call to a social data source (e.g., via an API call to a data source associated with the
 21 Monster.com website).

22 **[00202]** In yet another embodiment, such information may be retrieved from the
 23 data stored by the SMP. For example, the recruiter's affiliations may be determined via a
 24 SQL query substantially in the following form:

```

25 SELECT affiliations FROM UserInfo WHERE uid='Recruiter's Unique ID'
26

```

27 **[00203]** The recruiter's posted jobs, affiliations, contacts, information regarding
 28 members of a group at a company seeking an employee, and/or the like may be analyzed
 29 to determine relevant candidates at 570. In one embodiment, relevant candidates may
 30 be determined based on the closeness of the recruiter's connection (e.g., 1st degree
 31 contact, 2nd degree contact, and/or the like) with a candidate and/or the number of
 32 connections that a candidate has with members of the group seeking an employee,
 33 and/or based on the candidate's skills. For example, candidate relevancy may be
 34 determined according to a formula substantially in the following form:

```

35 Candidate Relevancy = (5 / closeness of connection) + (10 * # of 1st degree
36                   contacts at the group) + (2 * # of 1st degree contacts

```

1 at the company) + (0.5 * # of 2nd degree contacts) +
2 (10 * % of desired job skills that the candidate has)
3

4 Accordingly, the candidate relevancy in this example is determined as the sum of: (1)
5 five divided by the closeness of the candidate's connection with the recruiter (1st degree
6 contact: 1, 2nd degree contact: 2, and/or the like), (2) ten multiplied by the number of
7 the candidate's first degree contacts affiliated with the group at the company offering
8 the job, (3) two multiplied by the number of first degree contacts affiliated with the
9 recruiter's company, (4) one half multiplied by the number of second degree contacts
10 who are affiliated with the company, and (5) ten multiplied by the percentage of desired
11 skills listed for the job that the candidate has. In another embodiment, relevant
12 candidates may be determined based on the match between candidate's skills and/or
13 education and/or location and job prerequisites, and ordered for presentation to the
14 recruiter based on the candidate relevancy determined using social data (e.g., using the
15 candidate relevancy formula above). In additional embodiments, social interactions and
16 opinion makers, as described in the Connecting Internet Users section, may affect
17 candidate relevancy. For example, contacts who are identified as connectors may be
18 weighted higher than other contacts when calculating the candidate relevancy value.

19 **[00204]** Information regarding relevant candidates may be provided to the
20 recruiter at 575. For example, in response to a search query, the recruiter may be
21 presented with ten most relevant candidates as determined by the candidate relevancy
22 formula. In another example, a candidate advertisement presented to the recruiter may
23 be the most relevant candidate as determined by the candidate relevancy formula (e.g.,
24 social data may affect the choice of advertisements to present to the recruiter using the
25 Career Advertisement Network, as described in Advertisement Generation, Selection
26 And Distribution System Registration section, Advertisement Generation section,
27 Advertisement Targeting/Distribution section, and Advertisement Evolution section). In
28 some embodiments, such information may be presented to the recruiter using an
29 interactive display box 2300 illustrated in Fig. 23, as described in the Automated Online
30 Data Submission section.

31 **[00205]** A determination may be made at 580 whether relevant contacts (e.g., a
32 relevant candidate, the relevant candidate's direct or indirect contacts, employees of the

1 company where the relevant candidate currently works, alums of an educational
2 institution where the relevant candidate studied, and/or the like) are available for social
3 meetup. For example, interacting with a candidate may provide the recruiter with
4 additional information regarding the candidate. In another example, interacting with
5 alums of an educational institution where a candidate studied may help the recruiter
6 assess the quality of education provided by the educational institution. In one
7 implementation, such determination may be made by examining the
8 SocialMeetupPreferences field of the UserProfile data structure described with regard to
9 SMP network info 357. If one or more relevant contacts are available for social meetup
10 and the recruiter desires to do so, the SMP may facilitate social meetup with such
11 relevant contacts at 585. In one embodiment, the SMP may provide names (e.g., of the
12 candidate, of the recruiter, of alums, and/or the like), venue preferences (e.g., face-to-
13 face, group meeting, phone conference, chat room, and/or the like), length preferences
14 (e.g., meet for half an hour), candidate resume, additional job description, common
15 interests, and/or the like to the recruiter and/or relevant contacts. In another
16 embodiment, the SMP may also manage the venue for the social meetup (e.g., the SMP
17 may provide video, audio, text and/or the like connectivity applications for the social
18 meetup participants).

19 **[00206]** FIGURE 6 shows a logic flow diagram illustrating an Offer Providing (OP)
20 component in one embodiment of the SMP. The OP component may be used to provide
21 offer information to a group and/or to facilitate social meetup with the offer provider. In
22 Figure 6, a request to provide an offer to a group may be received at 605. Such an offer
23 request may be received as a result of a group member providing information regarding
24 the group's interests with regard to various offers. For example, a group member (e.g.,
25 the group leader, any authorized group member, and/or the like) may provide
26 information regarding offer types, locations, and/or the like that interest the group via
27 the SMP user interface (e.g., via an input box of a web page). In one implementation, the
28 offer request may include information such as a group ID, an offer type, an offer
29 location, and/or the like, and may be in XML format substantially in the following form:

```
30 <XML>  
31   <OfferRequest>  
32     <GroupID>Group's unique ID</GroupID>
```

```

1         <OfferType>Ergonomic Products</OfferType>
2         <OfferLocation>New York, NY</OfferLocation>
3     </OfferRequest>
4 </XML>
5

```

6 **[00207]** The group's members may be determined at 610 (e.g., by examining a
7 members database table associated with the group ID), and common characteristics of
8 the group's members may be determined at 615. For example, the group's members may
9 be software developers (e.g., database engineers of company X) working in New York.
10 The group's affiliations may be determined at 620. For example, such affiliations may
11 include the company X, another group within the company (e.g., application engineers
12 of company X), companies in the same industry, and/or the like. In one embodiment,
13 such information may be determined via an API call to a social data source (e.g., via
14 Facebook API calls). In another embodiment, such information may be retrieved from
15 the data stored by the SMP.

16 **[00208]** Data regarding the group may be analyzed to determine a relevant offer
17 at 625. For example, the SMP may determine that software developer groups interested
18 in ergonomic products tend (e.g., based on statistics collected over time, based on data
19 provided by a third party, and/or the like) to be interested in ergonomic office furniture
20 (e.g., ergonomic chairs). In another example, the SMP may determine that similar
21 groups in other companies in the same industry tend to be interested in ergonomic
22 office furniture. In another example, the SMP may determine that a representative of a
23 company that provides ergonomic office furniture has contacts in the group (e.g., first
24 degree contacts, second degree contacts, and/or the like). For example, offer relevancy
25 score may be determined according to a formula substantially in the following form:

```

26 Offer Relevancy = (10 * % of groups in the industry interested in this offer) +
27                   (2 * # of 1st degree contacts in the group) +
28                   (0.5 * # of 2nd degree contacts in the group)
29

```

30 **[00209]** Accordingly, the offer relevancy in this example is determined as the sum
31 of: (1) ten multiplied by the percentage of the groups in this industry that showed
32 interest in this offer, (2) two multiplied by the number of the first degree contacts that a
33 representative of a company providing the offer has in the group, and (3) one half
34 multiplied by the number of the second degree contacts that a representative of a
35 company providing the offer has in the group. In additional embodiments, social

1 interactions and opinion makers, as described in the Connecting Internet Users section,
2 may affect offer relevancy. For example, contacts identified as connectors may be
3 weighted higher than other contacts when calculating the offer relevancy value.

4 **[00210]** Information regarding a relevant offer (e.g., an offer having the highest
5 value calculated using the offer relevancy formula above), such as an offer to purchase
6 ergonomic office furniture, may be provided to the group (e.g., posted to the wall of the
7 group) and/or to group members (e.g., via an email message) at 630. For example, the
8 offer may be provided via a Facebook API call.

9

Connecting Internet Users

10 **[00211]** In the following detailed description of embodiments of the invention,
11 reference is made to the accompanying drawings in which like references indicate
12 similar elements, and in which is shown by way of illustration specific embodiments in
13 which the invention may be practiced. These embodiments are described in sufficient
14 detail to enable those skilled in the art to practice the invention, and it is to be
15 understood that other embodiments may be utilized and that logical, mechanical,
16 electrical, functional, and other changes may be made without departing from the scope
17 of the present invention. The following detailed description is, therefore, not to be
18 taken in a limiting sense, and the scope of the present invention is defined only by the
19 appended claims.

20 **[00212]** Fig. 7 illustrates relationships between an embodiment of an application
21 and various other modules or data stores, such as may be embodied in a medium
22 or in media. The application communicates with users through a user interface and
23 accesses data in various databases or data stores. Note that the term databases is used
24 for any collection of information, whether organized as a relational database (for
25 example) or stored in some other manner.

26 **[00213]** Within system 750, application 700 communicates with users through
27 user interface 710, which may be a website user interface or other graphical user
28 interface for example. Based on its communication with a user, application 700
29 accesses data in each of user database 725, test database 735, information database 745

1 and merchandise database 755. The databases described may be well defined or may
2 represent a collection of data such as may be found in a directory structure accessible
3 through a file system for example.

4 **[00214]** In one embodiment, the user database 725 includes user profiles which
5 encompass user login information, user history with the application 700, personal user
6 information, and results of user interaction with test database 735, information
7 database 745 and merchandise database 755 for example. In such an embodiment,
8 test database 735 includes tests of various types which may be administered, along
9 with information indicating how to analyze results of responses to the tests and
10 information indicating relationships between the various tests.

11 **[00215]** Moreover, in such an embodiment, information database 745 includes
12 reference or instructional information on a variety of topics, such as how to
13 interpret test results, self-help information, relationship information, career
14 information, or other information topics which may be of interest to a user or
15 users. Also, in such an embodiment, merchandise database 755 includes
16 information about goods or services for sale, and about merchants offering goods
17 or services for example. Merchandise refers to that which may be offered (such as
18 services or goods for example), rather than strictly to material goods in this context.

19 **[00216]** In response to a user query through user interface 710, application 700
20 may present information from information database 745, administer a test from test
21 database 735, lookup user information in user database 725 or lookup desired goods
22 and/or services in merchandise database 755. Similarly, application 700 may update
23 profile information in user database 725 responsive to user requests or actions, or
24 inventory information in merchandise database 755 for example. Other modifications
25 may be appropriate, depending on the form and availability of the various databases.

26 **[00217]** The application of Fig. 7 may be embodied in a variety of ways, either as
27 an apparatus or as a method. Fig. 8 illustrates an embodiment of an application as it
28 may be embodied in a medium or in media. Machine-readable media may embody
29 instructions, which, when executed by a processor, cause the processor to act,
30 thereby carrying out a method or uniquely configuring an apparatus. In one

1 embodiment, a medium includes an application and a variety of interfaces allowing
2 the application to communicate with a user and access (for both read and write
3 purposes) data in various data stores.

4 **[00218]** As illustrated, medium 800 includes application 810, the application
5 which receives commands or otherwise interacts with a user and manipulates data
6 responsive to interaction with the user and a host processor or system. Medium 800
7 also includes a user interface 820, which may be executed or operated to communicate
8 with the user. Note that medium 800 is depicted as a single, integrated or unitary
9 medium. However, medium 800 may actually be a collection of media of the same
10 or different types as appropriate in a particular embodiment or implementation.
11 Medium 800 further includes user information interface 830, test information
12 interface 840, information interface 850, and merchandise interface 860.

13 **[00219]** User information interface 830 may be executed or operated to obtain
14 information (such as personal information or login information for example) from a
15 user information database such as user database 725. Test information interface
16 840 may be executed or operated to obtain information from a test database such as
17 test database 735 (such as tests or relationships between tests for example).
18 Information interface 850 may be executed or operated to obtain information from an
19 information database such as information database 745 (such as web pages or
20 documents from a directory structure for example). Merchandise interface 860 may be
21 executed or operated to obtain data (such as merchandise characteristics or
22 availability for example) from a merchandise information database such as
23 merchandise database 755.

24 **[00220]** Tests used by an application may have a variety of relationships and a
25 variety of storage methodologies. All of these relationships may be maintained within a
26 relational database of test for example, along with numerous other relationships not
27 shown. Alternatively, tests may be stored as individual files within a file system,
28 and the file system may include various forms of links to other tests (files) within the
29 file system. The links between various tests may be used to determine a progression of
30 tests, or select among various progressions of tests. Similarly, when several tests are
31 linked, the various links may provide a basis for presenting a set of tests, one of which

1 is administered as a progressive test. Moreover, tests that are not directly linked
2 may be correlated through use of links to intermediate tests, potentially resulting
3 in a suggestion to administer an intermediate test to collect additional information.

4 **[00221]** Similarly, users of an application may be related in a variety of ways. Fig.
5 15 illustrates an embodiment of a process of interaction between a group and a vendor.
6 This interaction depends on the relationships between members of the group. The
7 method illustrated includes a number of modules that may be executed or operated in a
8 serial or parallel fashion, for example. At module 1510, an offer is provided to a
9 connector or set of connectors. These connectors are expected to propagate the
10 offer through the Internet. At module 1520, the connectors receive the offer. At
11 module 1530, the connectors propagate the offer to directly and indirectly
12 connected or coupled people through the Internet or a similar network, such as a
13 network designed around a progressive testing facility. At module 1540, responses
14 to the offer arrive with the connector(s). At module 1550, the connector(s) pass the
15 responses along to the vendor. At module 1560, the vendor completes the
16 transaction.

17 **[00222]** In some embodiments, variations of the method may be used. For
18 example, responses need not flow through the connectors, they may be sent directly
19 to the vendor. Similarly, offers of various forms may be used, such as circulating
20 coupons, group offers, progressive coupons, sponsored offers (wherein a connector or
21 third-party contributes to the vendor's offer), or other forms of offers. In some such
22 circumstances, a threshold level of responses (acceptances) may need to be met,
23 either as a percentage of responses or as a predetermined quantity of responses for
24 example.

25 **[00223]** A number of different mechanisms connecting buyers to sellers may be
26 understood with reference to Fig. 16. For example, various embodiments may be
27 understood from Fig. 16, including embodiments using circulating coupons, group
28 offers, progressive coupons, connectors circulating coupons, and connectors forming
29 groups. Each of these embodiments, along with other embodiments, may use a
30 structure as illustrated in Fig. 16 under various circumstances.

1 **[00224]** As illustrated in Fig. 16, user A1 has associated therewith data A2. User A1
2 propagates data A2 or data derived from data A2 to user B1, user C1 and user D1.
3 Such propagation may occur simultaneously, concurrently, serially, sequentially, or
4 otherwise. Consequently, user B1 has associated therewith data B2, user C1 has
5 associated therewith data C2, and user D1 has associated therewith data D2.
6 Additionally, user C1 may further propagate data, such that user E1 has associated
7 therewith data E2 and user F1 has associated therewith data F2. This basic model
8 may apply to a variety of embodiments.

9 **[00225]** In a first embodiment, each of data A2 - F2 represent an electronic
10 coupon. Such an electronic coupon may take the form of a code to be entered at a
11 website, data that may be provided as a paper coupon, or a pointer to a restricted-access
12 website, among other forms. Such a coupon may not be active until a predetermined
13 number of people have received it (a group coupon), it has been sent to a
14 predetermined number of people (a circulating coupon), or it has been used by a
15 predetermined number of people (a progressive coupon). Alternatively, the coupon may
16 have an initial value that may increase based on usage threshold(s) or reception
17 threshold(s). Note that reception/circulation/forwarding may be tracked using
18 embedded code or by tracking access to an URL where data referenced by the coupon
19 may be found. Moreover, rather than increasing value, reaching thresholds may result in
20 initial or increasing rewards (e.g., points, miles, rebates, etc.) related to the coupon. In
21 the connector context, user A1 may be a connector circulating a coupon A2.
22 Users B1, C1 and D1 may be directly connected to or coupled to connector A1 Users
23 B1 and F1 may automatically receive instances of the coupon or may receive them
24 responsive to action by user C1. Again, these instances of coupons may be
25 progressive, circulating, or result in group discounts and/or benefits. Moreover,
26 the coupons may also be stable, with a predetermined and unchanging value or
27 benefit. Additionally, other rewards or benefits may accrue to the connector because of
28 coupon use attributable to the connector.

29 **[00226]** In the context of group offers, users A1 - F1 may be previously
30 identified as sharing a common interest. Data A2-F2 may represent an offer to the
31 group to purchase a good or service. The offer may provide a static or progressive

1 discount or special price. Alternatively, the offer may require a predetermined
2 number of acceptances, percentage of acceptances, or volume of orders to activate
3 the offered discount or price. Acceptances may be conditioned on activation of the
4 offer. Note that an offer in this context may be an offer to bargain rather than a
5 legally binding offer.

6 **[00227]** In yet another embodiment, a connector may be provided an
7 opportunity to form a purchasing group. The connector A1 may distribute instances of
8 an offer A2 to potential members of a group of purchasers B1-F1 as instances B2-F2.
9 Offer A2 may provide for a static or dynamic discount on purchase of a good or service.
10 Alternatively, offer A2 may relate to a reward such as points, miles, rebates or other
11 inducements for purchase of a good or service. The inducement or discount may
12 require a predetermined number of purchases to reach an activation threshold or
13 may have tiers of acceptances which correspond to progressively higher levels of
14 inducements or discounts. Moreover, acceptance of offers within the group and/or
15 distribution of the offer to the group may reward the connector.

16 **[00228]** Research has shown that nearly 80% of Internet messages are generated
17 by approximately 5% of Internet users. Identifying these 5%, who may be
18 referred to as connectors, provides an opportunity for a form of direct or viral
19 marketing. By convincing these connectors to disseminate advertising and offers
20 to purchase or for discounts, a new marketing avenue may be opened. Using
21 rebates, commissions, points, miles or other rewards, some of these connectors may be
22 enticed to participate.

23 **[00229]** Other relationships between users may also be present. Fig. 14
24 illustrates relationships between users in one embodiment. Users 1400 are a collection
25 of related users. Alternatively, users 1400 is a database including information related
26 to a number of different users, and representations of similarities or affirmative
27 links between individual users. The database may also be used in an anonymous
28 manner for extracting common data or statistical models based on a statistically
29 significant sample of a population. Correlation within the model may be instructive as
30 to what products or services users are interested in, what common interests various

1 users have, what underlying relationships between users exist, and many other
2 correlated variables.

3 **[00230]** User 1410 is the user in question (such as a user currently logged in for
4 example). Logically related to user 1410 are users 1420, 1430 and 1440. The illustrated
5 links between users 1410, 1420, 1430 and 1440 are links deduced by the system which
6 have not been affirmed or requested by user 1410 or by some combination of users
7 1410, 1420, 1430 and 1440. For example, user 1410 may be a sibling of user 1430
8 and a child of users 1420 and 1440, with users 1420 and 1440 being either spouses or
9 ex-spouses for example. Such relationships within the set of users may be formalized or
10 recognized as necessary or as requested. However, the relationships may also be
11 utilized for their effects on a profile of user 1410 without formal recognition.
12 Similarly, user 1490 may be a co-worker of user 1410, thus allowing for a deduced link
13 which need not be formalized.

14 **[00231]** In contrast, users 1460 and 1470 are positively linked to user 1410,
15 such as by request of user 1410 for example. This may be a result of friendship, other
16 social bonds, or a result of referral to the system by one of the users. User 1460 is also
17 positively linked to each of users 1450 and 1465, each being linked to each other.
18 Moreover, user 1460 is positively linked to each of users 1470 and 1485, each
19 positively linked to user 1480.

20 **[00232]** In one embodiment, these positive links allow for communication
21 between users within the system (for example, email may only be sent to users
22 accessible through a direct positive link, or through traversal of a series of positive
23 links). Thus, user 1410 may be able to send email only to 1460 and 1470, or only to
24 1450, 1460, 1465, 1470, 1480 and 1485 for example. Additionally, user 1410 may
25 have well-defined connections only to users 1460 and 1470 within the system
26 1400.

27 **[00233]** Also illustrated are common characteristics, shown by symbols within the
28 users. For example, users 1410, 1450 and 1470 each have a 'X' indicating a common
29 characteristic. This may be due to an affirmative indication from each of the users in
30 question, or from deduction through statistical profiling of the users in question.

1 For example, this common characteristic may be interest in a form of art of a specific
2 athletic endeavor. Similarly, users 1410, 1430, 1460, 1480 and 1485 each have a P*¹
3 indicating a different common characteristic. Again, the common characteristic may
4 be deduced or affirmatively selected, and may have a variety of different forms.

5 **[00234]** Note that the user database 1400 may be correlated to the test database
6 1300, such that test results from tests of test database 1300 may be analyzed based on
7 relationships between users in user database 1400. Moreover, history of individual test-
8 taking by a user 1410 may be analyzed by reference to test database 1300, with such
9 analysis updated based on changes in test database 1300 (such as revised
10 relationships for example). Additionally, correlation of test analysis based on test
11 database 1300 and user profile analysis or statistical analysis from user database 1400
12 may lead to suggestions about purchases, information to browse, or groups to join for
13 example.

14 **[00235]** Applications may implement a variety of methods. In one embodiment,
15 only tests are progressively administered. Fig. 9 illustrates an embodiment of a
16 method of progressively administering tests 900. The process, in one
17 embodiment, involves administering a first test to a user, processing the results of a
18 test, providing the results (feedback) to the user, and suggesting either a next test,
19 a next set of tests, or a set of potential next tests, whereupon the user may then take
20 the next test and repeat the process of processing and feedback.

21 **[00236]** As illustrated with respect to Fig. 13, the tests may be progressive in
22 nature, such that a second test, next test, or first progressive test may drill-down or
23 otherwise focus on a single subject from a variety of subjects in a first test or general
24 test. The progressive nature allows for more refined, sophisticated or nuanced
25 analysis of information from the tests collectively, and historical tracking of test
26 results allows for gradual and progressive updating of an analytical picture or
27 profile of a user.

28 **[00237]** At module 910, a query arrives from a user, indicating interest from
29 the user in taking a first test. This query may represent a more involved process of
30 enrolling a user into a membership group for example, and/or obtaining financial

1 information allowing for monetization of the transaction in which the first test (or
2 future tests) is administered. The query may also be as simple as clicking on a link on a
3 website which leads to administration of the first test.

4 **[00238]** At module 920, the first test is administered. The first test may be a
5 predetermined test, or a test selected from a predetermined set of tests. In some
6 embodiments, the first test is a personality test designed to provide information
7 about the preferences and personality of the user, assuming a reasonable
8 attempt to faithfully take the test (rather than answering questions outrageously
9 for example).

10 **[00239]** At module 930, the results of the test (such as the first test) are processed,
11 thereby analyzing some aspect of the user (such as IQ or intelligence quotient,
12 personality, knowledge of financial principles, for example). At module 940,
13 feedback is provided to the user in the form of test results, either as a score, an
14 analysis of correct and incorrect answers, an analysis of trends in answers, some
15 combination of all of these, or some other form of feedback. The feedback may be
16 separate from other interactions with the user, or may be combined with the
17 suggestion of module 950 for example.

18 **[00240]** At module 950, a next test is suggested, based in part on the results of
19 the test(s) already administered and processed for the user. The next test may be a
20 single identified test, a set of tests from which a choice may be made, or a series of
21 several tests (some or all of which may ultimately be administered). At module 960,
22 the next test is administered, and the process then moves to module 930 for
23 processing of the results of the next test. In this manner, the loop may be iterated
24 several times, to refine results of tests and explore additional facets of a person.
25 Furthermore, the tests may each be monetized, and feedback may be monetized,
26 allowing for fees for administering the tests or for providing either any analysis, or
27 in-depth analysis beyond free superficial analysis.

28 **[00241]** A more complex embodiment of a method may progressively administer
29 tests and sell goods or services. Fig. 10 illustrates an embodiment of a method of
30 progressively administering tests and selling merchandise 1000. The process, in one

1 embodiment, involves administering a first test to a user, processing the results of
2 a test, providing the results or feedback to the user, suggesting a next test or
3 assortment of tests. Then, the process includes either administering the next test
4 or monitoring shopping of the user, and revising the proposed next test based on
5 user activity.

6 **[00242]** At module 1010, a user query is received, initiating a session. At
7 module 1020, a first test is administered. At module 1030, results of a test are
8 processed, analyzing the answers given by the user. At module 1040, results of the test
9 are provided as feedback in one form or another to the user. At module 1050, a next
10 test is suggested. Note that the modules 1010, 1020, 1030, 1040 and 1050 may be
11 implemented in a manner similar to that of modules 910, 920, 930, 940 and 950 of
12 Fig. 9 for example.

13 **[00243]** At module 1060, a decision is made as to whether the user will take
14 further tests at this time. If yes, the next test is administered at module 1055, and the
15 process returns to module 1030. If no (at module 1060), a decision is made at module
16 1070 as to whether the user will shop for goods and/or services.

17 **[00244]** If, at module 1070, the decision is yes (to shop), then the process moves to
18 module 1065, and merchandise (goods and/or services) is displayed for perusal by the
19 user. As the user browses, and as goods or services are chosen for purchase, this is
20 monitored at module 1075. Upon termination of shopping, at module 1085, purchases
21 are processed (such as checking out and arranging payment for example), and information
22 from browsing and purchasing is fed back into a profile for the user. Based on the user
23 profile, at module 1050, a next test is suggested. This next test may be the same test
24 suggested after analysis of the first test, or may be a different test, reflecting new information
25 gleaned from shopping activities.

26 **[00245]** If, at module 1070, the decision is no (not to shop), the process moves to module
27 1080, and a decision is made by the user as to whether to log out or not. If the user does
28 not log out, the process moves to module 1050 and a next test is suggested. If the user
29 logs out, the process ends the session at module 1090. The user may then log back in at
30 module 1015, avoiding the original first test. At module 1035, data that the system has

1 received since the last session for the user is processed. This data may be statistical
2 modeling data, or information from other users, either linked to the user through a
3 network, or related to the user in some way, such as by family relation, social or
4 professional relationship. The process then moves to module 1050, and a next test is
5 suggested based on this updated profile of the user.

6 **[00246]** Note that administering tests and providing results may be monetized as
7 discussed with respect to Fig. 9. Moreover, this monetization may be linked to
8 shopping; leading to discount programs, free tests in response to shopping at
9 affiliated merchants, and running accounts of credits or debits in a user's
10 account. The initial user query may involve arranging for payment for services, either
11 through credit or debit processes, or through periodic billing for example. Such credit
12 or debit processes may involve interaction with credit accounts, deposit accounts
13 of a bank, credit union or similar financial institution, cash deposits or other
14 deposits at a point-of-sale terminal, or maintenance of an internal balance within
15 a system using the methods and/or apparatuses described herein. Furthermore,
16 note that monetization may permeate the system, as an option in every module of
17 a process for example, and may relate to payments by a sponsor to allow a user to
18 utilize the system. When a sponsor is involved, identification of the sponsor in the
19 process may occur, or perception of an advertisement from the sponsor may be
20 involved, with perception including visual, audio, or other forms of perception.

21 **[00247]** Yet another complex embodiment may be used to progressively
22 administer tests, sell goods and services, and provide information. Fig. 11 illustrates an
23 embodiment of a method of progressively administering tests, providing information
24 and selling merchandise 1100. The process, in one embodiment, includes
25 administering a first test, processing and providing results of a test, and suggesting a
26 next test or assortment of tests. Then, the user may either take the next test, shop, or
27 browse for information (or logout and come back later). Based on the user's
28 activities, the proposed next test is revised, and content (information, possibly
29 merchandise) presented to the user is adjusted. At module 1110, a user query is
30 received, initiating a session. At module 1115, a first test is administered. At module
31 1120, results of a test are processed, analyzing the answers given by the user. At

1 module 1125, results of the test are provided as feedback in one form or another to
2 the user. At module 1130, a next test is suggested. Note that the modules 1110, 1115,
3 1120, 1125 and 1130 may be implemented in a manner similar to that of modules
4 910, 920, 930, 940 and 950 of Fig. 9 for example.

5 **[00248]** At module 1135, a determination is made as to what the user will do
6 next. The options include taking a test, shopping, browsing information, and logging
7 out. If the decision is to take a test, then at module 1140, the next test is
8 administered, and the process returns to module 1120 for processing.

9 **[00249]** If the decision is to shop, at module 1145, merchandise is displayed,
10 potentially in a manner responsive to aspects of the user profile already established. At
11 module 1150, browsing and choices for purchases are monitored, both for accounting
12 purposes and for profile building. At module 1155, purchases are processed, such as
13 by finalizing a transaction and arranging for delivery or appointments for example.
14 The process then proceeds to module 1130, at which point a next test is suggested
15 based on the current contents of the user profile, including information from the
16 shopping session.

17 **[00250]** If the decision is to browse information, information access is provided
18 at module 1160. Such information access may be shaped based on profile contents,
19 either by emphasizing potential topics, or by presenting a group of topics expected to be
20 of interest to the user for example. At module 1165, actual browsing by the user is
21 monitored, adding further information to the user profile. When browsing terminates,
22 the process moves to module 1130, at which point a next test is suggested based on
23 the current contents of the user profile, including information from the information
24 browsing session.

25 **[00251]** If the decision is to log out, at module 1170 the session is ended. At
26 module 1175, the process awaits a login by the user. At module 1180, the user logs back
27 in. At module 1185, changes to a user profile occurring while the user was not logged in
28 are processed, such as information submitted by people solicited to evaluate the
29 user or information from evolving statistical models for example. The process then
30 moves to module 1130, at which point a next test is suggested based on the current

1 contents of the user profile, including information from the changes or data
2 processed.

3 **[00252]** Note that opportunities for monetization abound within the method of
4 Fig. 11. For example, a test may be sponsored by a sponsor, with the test subject matter
5 related to the sponsor's products and/or services. Similarly, browsing information may
6 be monetized based on what information is accessed or the time and/or bandwidth
7 taken to access the information. Such monetization may be based on a user paying a
8 subscription fee or on a user viewing or otherwise perceiving an advertisement or
9 information from a sponsor, with the sponsor paying the sum involved in
10 monetization. Monetization in each of the methods or embodiments may also be
11 related to levels of feedback for results from a test or set of tests. Moreover,
12 monetization may be involved in soliciting input from external sources of data such
13 as friends, relatives or co-workers. Additionally, joining groups or setting up matches
14 between users may be accomplished based on apparent compatibility within respective
15 profiles, and monetization may be involved therein.

16 **[00253]** A progressive system for managing an interface with a user may utilize a
17 variety of relationships between the user and various sources of data. Fig. 12 illustrates
18 relationships between an individual profile and various sources of data or data stores
19 in one embodiment. The relationships with a user (and conduits for the user
20 profile) fall into four broad categories (user activated, related user activated,
21 statistical modeling, and presentation to the user) in one embodiment. User profile
22 1210 represents a collection of data about a user. This may include attribute-value pairs,
23 histories of browsing and testing, pointers to related members in the network, and
24 other representations of data. This data may correspond to test results, personal
25 characteristics, histories as mentioned, connections to people related by family, social,
26 occupational and other links, and other forms of information.

27 **[00254]** Sources of information can be grouped into three categories: user-
28 created information, related-user information, and statistical information. User-created
29 information includes frequency of use 1230 (how often a user logs in, how long
30 sessions last for example), purchase monitoring 1235, network relationships 1240 (who
31 the user is connected to within a network of users), information monitoring 1245

1 (indications of what a user browses within a website for example), and test results
2 1250. Each of these sources of information comes directly from user actions, and
3 relates specifically to the user in question.

4 **[00255]** Related-user information includes references or testimonials 1260, testing
5 of relatives 1265, and testing of friends 1270. References or testimonials 1260 may be
6 solicited directly by the user or indirectly through a website implementing a network
7 of members or users. Such references or testimonials 1260 may include standardized
8 evaluations of users or freeform evaluations of users, and may be monetized, such as
9 by a fee for obtaining and/or processing the information. Relatives and friends may
10 be determined based on membership within a network of members or users, in which
11 the friends or relatives are pointed to by the user profile 1210. Such relatives or friends
12 may be identified by the user, or may be deduced by user activity (email exchanges
13 for example) and characteristics (same address for example). Results of relatives
14 testing 1265 and friends testing 1270 may reflect on the user based on whom the
15 user is related to or associates with.

16 **[00256]** Statistical modeling 1220 involves aggregating information from a
17 (preferably) statistically significant number of users and extracting trends or
18 correlations between variables therein. Data from user-created information and
19 related-user information sources in a user profile 1210 may be aggregated,
20 allowing for prediction of unknown information within a user profile 1210. For example,
21 if most people in San Francisco with library cards tend to buy books, a user in San
22 Francisco with a library card may be predicted to be interested in books.

23 **[00257]** Such predictions may be used in conjunction with the process of selecting
24 tests to present, information to present, merchandise to present and potential
25 relationships to present. Relationship presentation 1280 may be a module of an
26 application (such as application 700 for example) which suggests people who
27 would be good matches as friends, potential spouses, or other forms of acquaintances.
28 Test selection 1285 may be a module of an application (such as application 700 for
29 example) which selects a next test based on information in the user profile 1210,
30 both to help the user become more self-aware and to collect more data for the user
31 profile 1210.

1 **[00258]** Information presentation 1290 may be a module of an application (such
2 as application 700 for example) which determines what topics of information should be
3 presented or emphasized when a user browses an informational website, based on
4 user profile information and potentially statistical modeling information. Merchandise
5 presentation 1295 may be a module of an application (such as application 700 for
6 example) which indicates what merchandise (goods and/or services) should be
7 presented or emphasized when a user browses a sales portion of a website.

8 **Automated Online Data Submission**

9 **[00259]** Fig. 17 illustrates a high-level flow diagram of an embodiment of the
10 present invention. The flow diagram illustrates the entities involved with managing,
11 storing, configuring and transmitting the data exchanged by the system between entities
12 using the AODSA tool. By way of example only, the system includes a system processor
13 1700 and a system database 1710 configured to store and manage user data (e.g., job
14 applicant's data including resumes). As illustrated in Fig. 17, the system processor 1700
15 and the system database 1710 are situated remotely (e.g., on a remote server). However,
16 it is to be understood that although these elements are implemented remotely, alternate
17 embodiments may utilize a user's local resources (e.g., desktop CPU and/or hard drive)
18 in coordination with a locally stored system application 1730.

19 **[00260]** It is to be understood, that the invention will be discussed in the job
20 application data submission context and that the invention may be configured for other
21 implementations such as mortgage applications, bidding on real estate or other goods or
22 services, applying for admission to schools or organizations, applying for internships or
23 volunteer positions, applying for scholarships or grants, and/or the like. As illustrated
24 in Fig. 17, the systemization 1700/1710 may include server-side functionality/processing
25 that is accessed by a system user through system application 1730 (e.g., a java-enabled
26 applet running locally on a user's desktop). Alternately, the applet may run as a
27 background task and is accessed when a system user (e.g., a job applicant) wants to
28 submit application information in response to a job posting.

1 **[00261]** Alternately, system application 1730 may be bundled as a software
2 application that is situated locally and utilizes a computer's central processing unit as
3 system processor 1700 and a computer's hard drive as the system database 1710. As will
4 be described in greater detail below, online data content 1740 may be viewable on a
5 system user's computer through the use of a system application 1730, such as a web
6 browser. Such online data content 1740 may, in one implementation, be presented to a
7 user within the context of a content provider 1725 website, such as in the form of a
8 banner ad situated on a content provider's news web site.

9 **[00262]** On a high level, a user interacting with system application 1730 browses
10 online content 1740 via communications network 1750. When the user wants to submit
11 job application data, system application 1730 interacts with the system processor 1700
12 and system database 1710 over communications network 1750 to access and forward the
13 requested job application data associated with the system user.

14 **[00263]** Fig. 18A, 18B, and 18C illustrate flow diagrams of the user data
15 registration and profile creation processes. According to an embodiment of the
16 invention, represented in Fig. 18A, the system user initiates the system application in
17 step 1810. The user selects either a manual registration 1815 procedure or an automated
18 registration 1820 procedure. In the manual registration 1815, the user manually enters
19 job application data in step 1825 including name, contact information, employment
20 history and/or other identifying information. In step 1845, the user may be presented
21 with an option for assistance in creating a resume based on the entered data. According
22 to one implementation, the system user may select a system resume template and an
23 interactive data entry module. As part of the interactive data entry module, the system
24 data entry application presents the user with a series a questions designed to extract
25 certain user information that would appear on a resume or could be used to populate
26 online employment application forms.

27 **[00264]** In some implementations, the system may present the user with an option
28 to upload an electronic copy of a resume in step 1845. The system application uploads
29 and stores the information in the centralized system database in step 1850. In some
30 implementations, the system may be configured to transmit an acknowledgment
31 message indicating the AODSA tool is ready for use, as in step 1855.

1 **[00265]** Alternately, the user may select the automated registration process 1820.
2 The automated process starts with the user uploading an electronic copy of a resume
3 and/or submission cover letter and indicating the corresponding file format in step
4 1830. The system parses the resume and extracts data corresponding to database fields
5 such as contact information, employment history, or education history in step 1840.
6 The system uploads the data to the system database in step 1850 and in some
7 implementations transmits an acknowledgment message in step 1855 that the AODSA
8 tool is ready for use.

9 **[00266]** Fig. 18B shows a logic flow in one embodiment of resume parsing and
10 profile creation. At 1860, the system receives a user resume, which is parsed at 1865 for
11 recognizable resume elements, such as but not limited to name, social security number,
12 e-mail address, postal address, education, work experience, honors and awards, skills,
13 and/or the like. In one implementation, the system may employ optical character
14 recognition techniques in order to convert a resume submitted in an image format into a
15 text format that may be manipulated and/or analyzed more conveniently. In an
16 implementation, the converted resume may serve as the basis for creating a user
17 portfolio of one or more customized resumes.

18 **[00267]** The system provides a great deal of flexibility and may be tailored to meet
19 the needs of any number of system users. For example, the resume element recognition
20 process may be implemented in a variety of different ways. In an implementation,
21 terms extracted from the user resume may be compared against a database of known
22 resume terms in order to identify resume elements or data field identifiers. In another
23 implementation, only those terms from the user resume that appear in a special font
24 (e.g., bold, underlined, italics, large font, etc.) are considered as possible resume field
25 names. When a resume field name is detected, the system extracts field data associated
26 with and/or proximate to that field name at 1870. In one implementation, the system
27 may detect a special character (e.g., a colon) after the field name and extract as field data
28 any text after that character and before a carriage return, the next field name, and/or
29 the like. Each detected field name is stored with its associated field data in a user profile
30 record at 1875. At 1880, the system determines whether there are additional resume
31 field names to consider and, if so, the flow returns to 1865. Otherwise, the system

1 proceeds to 1885 where the user profile record is displayed to the user for approval
2 1890. If the user is not satisfied, he or she is given the opportunity to edit user profile
3 record fields at 1895. Otherwise, the user profile record is persisted in a system
4 database at 18100 for future use.

5 **[00268]** Fig. 18C shows detailed logic flow in another embodiment of profile and
6 resume creation. At 18105, the system presents a user with a registration web form that
7 may contain a plurality of questions and/or blank fields by which the user may enter
8 personal information. At 18110, the system receives the user responses entered into the
9 web form. A choice is presented to the user at 18115 as to whether or not he or she
10 would like to generate a resume based on the information submitted at 18110. If not,
11 the flow proceeds directly to 18150, wherein the entered user web form responses are
12 persisted in a user profile record stored in a system database.

13 **[00269]** If a resume is desired, on the other hand, then the system may present the
14 user with a plurality of resume template choices at 18120. These may, in one
15 implementation, be in the form of example resumes and/or contain descriptions of the
16 resume styles along with recommendations for appropriate situations in which to
17 employ the various templates. The system receives a user selection of a particular
18 resume template at 18125, populates resume fields in the template with user web form
19 responses at 18130, and presents the resume for user inspection at 18135. The user
20 indicates at 18140 whether or not he or she is satisfied with the resume in its current
21 form and, if not, may be given the opportunity to edit resume fields at 18145. The
22 completed resume is persisted as part of the user profile record at 18148, and the user is
23 given the option to create new and/or alternate resumes at 18149. The user web form
24 responses may be separately incorporated into the user profile record at 18150.

25 **[00270]** Fig. 19A illustrates a high level flow diagram of an autonomous automated
26 data submission process associated with an embodiment of the invention. The system
27 user browses online generic job listings in step 1910. The user identifies a particular job
28 listing that they want to pursue in step 1920. The system user can then access the
29 AODSA tool in step 1930. Depending on the particular implementation, the AODSA tool
30 may present the system user with a range of application data submission options (as
31 discussed in greater detail in Figs. 20A, 20B, 21A-21C and 22) in step 1940. In step

1 1950, after the system user selects a AODSA data submission component is selected, the
2 AODSA tool accesses the user data on the centralized system and transmits the user
3 data to the corresponding posting entity.

4 **[00271]** Fig. 19B illustrates a high level flow diagram of an embedded automated
5 data submission process associated with an embodiment of the invention. The system
6 user accesses a content provider website at 1958. The content provider may be a system
7 affiliated entity or otherwise provider with an agreement to display system tools to
8 appropriate users. At 1960, the content provider checks the user's computer for a cookie
9 or other indication of user identity and/or system affiliation, based on which the content
10 provider may determine eligibility or appropriateness of system tool distribution and/or
11 display.

12 **[00272]** A determination is made at 1970 whether or not an appropriate cookie
13 exists and, if not, the automated data submission process may offer the user an
14 opportunity to register for system participation 1975. Should the user decide to do so,
15 the system proceeds to a registration process such as that outlined in Fig. 18A.
16 Otherwise, the system exits at 1980 and no system tool is provided to the user.
17 Otherwise, the content provider queries cookie contents at 1990, such as user
18 identifying information, user system identifying information, and/or the like. At 19100,
19 the content provider forwards extracted cookie information to a system server, which
20 processes that information in order to select system data for inclusion in a system web
21 module. Depending on the implementation, the system may be configured to provides a
22 wide variety of content/functionality to an identified system user. For example, the
23 content provider may act as a gateway and provide access to a system user's full user
24 account/functionality on the system (discussed in greater detail below in Figs. 22 and
25 23). The content provider retrieves the system web module from the system 19110 and
26 displays it to the user at 19120.

27 **[00273]** Figs. 20A and 20B illustrate flow diagrams associated with form
28 population and resume/cover letter generation processes, respectively. The system
29 undertakes the steps shown in Fig. 20A when a user initiates application submission
30 2001 involving an online data entry form. At 2005, the system queries the name of the
31 next empty web form field (e.g., name, social security number, work experience,

1 education, etc.) and subsequently searches stored user profile information for a
2 matching field entry 2010. In one implementation, this is accomplished by scanning
3 user profile information for character strings matching web form field names that have
4 proximate, non-empty data entries.

5 **[00274]** At 2015, a determination is made as to whether the current web form field
6 exists in the user profile and, if not, that field is noted in a temporary record of empty
7 web form fields. Otherwise, the data entry from the user profile that corresponds to the
8 web form field is used to populate that field at 2025. A determination is made at 2030 if
9 there are remaining empty web form fields to be filled and, if so, the system returns to
10 2005. Otherwise, the system checks at 2035 whether any of the missing web form fields
11 noted at 2020 are required for form submission. If so, the system may prompt the user
12 for manual entry of data pertaining to those fields at 2040. Alternately, in some
13 implementations, the scan may include alternate field matching if a match is not
14 identified (e.g., searching and entering address information in a field titled, "residence",
15 if a field for "mailing address" is not matched). The completed form is submitted by the
16 system at 2045.

17 **[00275]** The system undertakes the steps shown in Fig. 20B when a user initiates
18 application submission 2050 involving resume and cover letter submission/generation.
19 The system determines at 2055 and 2065 whether multiple cover letter and/or resume
20 templates are available for the user to choose from and, if so, requests the user's
21 selections at 2060 and 2070. Once unique cover letter and resume templates are
22 selected, the system queries the name of an empty cover letter or resume field at 2075.
23 The system searches stored user profile information for a matching field entry 2080
24 and, a determination is made at 2090 whether a matching entry exists in the user
25 profile. If not, the missing field is noted at 2092, and if so, then the field is populated
26 with the corresponding user profile information at 2095.

27 **[00276]** The system determines whether additional empty resume/cover letter
28 fields exist at 20100 and, if so, the system returns to 2075. Otherwise, the system
29 determines at 20105 whether the missing fields are required for generation of the
30 resume or cover letter. If so, the system requests the user to enter data for those fields
31 at 20110. Finally, the system generates the cover letter and resume based on the

1 collected user profile information 20115, and submits them to the desired location at
2 20120. In an optional step 20118, the system may present the generated resume and/or
3 cover letter for display to the user, who may then decide whether one or both are
4 acceptable, or may choose to manually modify or supplement data included therein. At
5 any point during this process, the user may save a current/modified resume to user at a
6 future point as a template. Furthermore, the user may create a portfolio of these saved
7 resumes for future user. This may be useful in creating a variety of resumes each with
8 customized objectives (e.g., a general resume tailored for a software engineering
9 position, a more specific resume highlighting certain experiences for a Java
10 programming position, etc...).

11 **[00277]** In an alternative embodiment, instead of generating new cover letters
12 and/or resumes in response to a user request for application submission, the system
13 may store a selection of pre-made resumes and/or cover letters. The user may access,
14 customize and save the pre-made resumes/cover letters and incorporate them into an
15 application submission package.

16 **[00278]** Figs. 21A, 21B, and 21C illustrate examples of user invocations of the
17 AODSA tool according to implementations of system application 130 (from Fig. 17). Fig.
18 21A illustrates an example generic data posting. By way of example only, Fig. 21A
19 implements a generic job listing 2100 that lists a series of current software engineering
20 job opportunities 2100. The generic job listing may be configured as a listing on a
21 generic job listing repository, such as a web-based classified listing. Alternately, the
22 generic job listing may be hosted by a particular company, and detail the current
23 opportunities available within the company or a particular industry (e.g. jobs within
24 IBM or within the Computer Programming Industry).

25 **[00279]** In Fig. 21B, the job applicant selects an internet hyper-link corresponding
26 to a posted job 2105 from job listing 2100 in Fig. 21A. The user is then transferred to
27 the corresponding web page (Fig. 21B) associated with the particular job description and
28 can invoke the AODSA tool 2110. The AODSA tool 2110 provides a job applicant (or
29 other system user) with a wide range of application data submission options, including
30 an upload additional/redacted resume 2110; auto-fill a form with identifying
31 information option 2120; auto-forward an email requesting additional

1 information/forwarding a standardized job application cover letter with a resume
2 attached 2130; or an option to update/edit stored resume data 2140.

3 **[00280]** After reviewing the opportunities detailed on the web page, the user may
4 select the appropriate data submission and the user's data is retrieved from the AODSA
5 centralized system and forwarded accordingly. According to the implementation
6 illustrated in Fig. 21B, there are two primary user data transmission procedures (a) an
7 online job application form auto-fill procedure 2130; and (b) emailing a cover letter with
8 a resume to an email recipient extracted from the data posting 2140.

9 **[00281]** If AODSA component 2130 is selected, in coordination with the "click here
10 to apply" link, the AODSA tool may spawn a new browser window with the online form.
11 The AODSA tool may be configured to retrieve the user's identifying information and
12 attempt to auto-fill the elements of the form based on the user's data retrieved from the
13 system database.

14 **[00282]** If AODSA component 2140 is selected - the auto-email procedure - the
15 AODSA tool may be configured to automatically email a user-selected resume and cover
16 letter to a particular email address. Further, it is to be understood that in addition to
17 submitting/updating resume data in AODSA components 2120/2150, the AODSA tool
18 may be configured to assist the user in creating a number of stored cover letters to
19 accompany the resume. Alternately, the AODSA tool may create an email with
20 standardized employment application language with blanks that users can customize
21 before the cover letter sending to the posting entity.

22 **[00283]** An embodiment of the auto-email interface is exhibited in Fig. 21C,
23 wherein the user is requested to select from a portfolio of saved resumes and cover
24 letters or pre-configured resume/cover letter templates. In this example, the resume
25 selections are Software Engineering 2160, Java Programming 2165, combination 2170,
26 or custom 2175, and the cover letter selections are specific 2180, general 2185,
27 professional 2190, or custom 2195. Selection is made in this implementation by means
28 of checkbox widgets 21100, though a variety of other interactive interface widgets are
29 possible in other implementations. In one implementation, the user selects templates
30 that are to be populated on the fly to generate cover letters and/or resumes, while in

1 another embodiment, the user selects actual saved resumes and/or cover letters to be
2 directly incorporated into application packages.

3 **[00284]** Fig. 22 illustrates an embodiment of the invention directed to serving
4 AODSA functionality via an ad server as a portable web module embedded within a
5 browser application. As illustrated, the user may surf the internet and access a
6 particular website, for example a content providing 2200. The AODSA tool may be
7 incorporated into a partner's website, in an area of the website that has been set aside
8 for advertisements 2205.

9 **[00285]** In an implementation, the web module identifies the system user and
10 access their full user data profile on an affiliate web site. The system user may be
11 provided with full access to their user data profile and/or all of the functionality
12 associated with the affiliate web site while using the content provider as an
13 intermediary. For example, a web user registered with Monster.com accesses Content
14 Provider CNN.com. The web user is identified by CNN.com as a registered Monster user
15 and is provided access to their Monster.com account and/or Monster.com functionality
16 (e.g., conducting job searches) without leaving the content provider's web site.

17 **[00286]** The AODSA tool is a fully functional portable web module, in which
18 content can be served via as an online advertisement (e.g., via ad-tag). Accordingly, the
19 portable web module may be configured to recognize a system user through a matching
20 user data stored locally such as via a cookie. The system may then generate a
21 customized list of jobs for a particular system user, which are then displayed for the
22 system user as content within the portable web module. This process is illustrated in
23 greater detail in Fig. 19B. The portable web module may be configured with a control
24 bar 2215 to facilitate system user interaction with the AODSA tool set.

25 **[00287]** Depending on the particular implementation, the control bar 2215 may be
26 configured with additional job listing data presentation components. By way of example
27 only, the control bar 2215 may be configured to facilitate additional system user driven
28 keyword searching within a designated system database. In some implementations, the
29 user can change the geographic focus 2225 of a key word search. In such

1 implementation additional data entry windows 2220, 2225 may be spawned in order to
2 facilitate user interaction.

3 **[00288]** Although Fig. 22 illustrates an embodiment directed to presenting certain
4 job listings selected from a general jobs database, it is to be understood that this
5 discussion is simply for purposes of illustration. The actual implementation may be
6 further adapted to meet the needs of a particular application.

7 **[00289]** By way of example only, the portable web module AODSA implementation
8 may be configured to facilitate general job listing search functionality, based on key
9 words, search terms, company names, industries, geographical areas, experience and/or
10 educational levels, skills, salary range, and/or the like. Alternately, the displayed
11 content may be customized according to settings established by a particular system user
12 to display certain categories of jobs within a particular location, associated with a
13 particular industry/job segment, user-defined salary range or other user-defined display
14 parameter. It is to be understood that in additional embodiments of the invention, the
15 portable web module may be further customized to illustrate listings associated with a
16 co-brand and/or partner posting entity. Moreover, the portable web module may be
17 adapted for private labeled postings to conduct customer recruiting.

18 **[00290]** The portable web module AODSA tool 2300 also may be configured to
19 provide functionality similar to that described in Figs. 21A, 21B, and 21C. By way of
20 example only, Fig. 23 illustrates the AODSA tool portable web module 2300 adapted to
21 interact with the system user.

22 **[00291]** In an implementation, the user may select a particular listing 2210 from
23 Fig. 22. As illustrated, upon selection of a listing 2210, the portable web module 2300
24 retrieves and displays additional data associated with the listing 2210. Depending on
25 the amount of detail for the listing, the portable web module may be configured to
26 facilitate page browsing, wherein the user clicks an "advance" portion of the display
27 2305 to "turn the pages" of the displayed data associated with the posting 2210. The
28 portable web module may include a listing browsing functionality button 2310 that
29 enables a system user to navigate between detailed descriptions of the job listings 2210

1 at a granular level (i.e., where detailed listing data associated with a single listing is
2 displayed to a system user).

3 **[00292]** In some embodiments, the portable web module may also be configured
4 with auto resume submission 2315, listing auto-fill functionality (similar to the
5 functionality discussed above in Figs. 21B and 21C), and/or a listing bookmark feature
6 2320 that saves the selected job listing/company information/content to a system user
7 data profile for review at a later time.

8 **[00293]** In an embodiment, the portable web module is configured to facilitate
9 resume submission for a displayed job listing 2210. Depending on the implementation,
10 the user may simply drag and drop an electronic resume 2315 (e.g., resume formatted as
11 a Microsoft word document, a .PDF file, or any other number of formats of digital
12 resume data) from a desktop or a file folder to the portable web module in order to
13 facilitate the application process. Alternately, the portable web module may be adapted
14 for the data submission processes and/or resume/cover letter creation processes
15 associated with Figs. 21B and 21C and discussed above.

16

Advancement Path Taxonomy

17

Career Statistical Engine

18 **[00294]** Figures 24-37B detail a career statistical engine (hereinafter, “CSE”)
19 component of the APPARATUSES, METHODS AND SYSTEMS FOR ADVANCEMENT
20 PATH TAXONOMY (hereinafter “APT”), which is detailed in figures 38-58. The CSE
21 allows for the generation and statistical mapping of an advancement state structure,
22 which furthers analysis associated with job market analysis, job search strategies, career
23 counseling, educational advancement, financial advancement, and/or the like. It is to
24 be understood that depending on the particular needs and/or characteristics of a job
25 seeker, employer, career counselor, system operator, hardware implementation,
26 network environment, and/or the like, various embodiments of the APT may include
27 a career statistical engine component, which may include implementations allowing a
28 great deal of flexibility and customization. The instant disclosure discusses an
29 embodiment of the CSE within the context of job market analysis, career path

1 modeling, job search strategies/recommendations, and/or the like. However, it is to
2 be understood that the CSE may be readily configured/customized for a wide range
3 of other applications or implementations. For example, aspects of the CSE may be
4 adapted for operation within a single computer system or over a network, for use in
5 educational path modeling and/or recommendations, task management, skill
6 development; and/or the like. It is to be understood that the CSE may be further
7 adapted to other implementations or experience analysis and management applications.

8 **[00295]** FIGURE 24 shows an overview of entities and data flow in one
9 embodiment of CSE operation. The CSE 2401 may be configured to allow a plurality of
10 job seekers (Job Seeker 1, Job Seeker 2, . . . , Job Seeker N) 2405 to interact with the CSE
11 and/or engage CSE functionality. A job seeker may communicate with the CSE, such as
12 via a communications network 2410, and/or directly via a job seeker/network interface
13 2415. The job seeker/network interface 2415 may be coupled to a CSE controller 2420,
14 which may serve a central role in facilitating CSE functionality and mediating
15 communications and/or data exchanges between CSE modules, databases, and/or the
16 like. The CSE controller 2420 may be further coupled to a resume acquisition module
17 2425, configured to receive and process resumes from job seekers 2405. In alternative
18 embodiments, the CSE may be configured to receive and/or process one or more of a
19 variety of different experiential sequences, such as educational transcripts, task lists,
20 award histories, military records, and/or the like. The CSE Controller 2420 may further
21 be coupled to an analysis module 2435, configured to analyze received resumes and to
22 determine statistical relationships between and among experiences, job titles, education
23 levels, accomplishments, and/or the like listed therein. The CSE Controller 2420 may
24 further be coupled to a plurality of databases storing data received and/or processed by
25 the CSE. Such databases may include, for example, an attributes database 2438, storing
26 attributes data derived from submitted resumes; a state model database 2440, storing
27 elements of the state model; a resume database 2430, storing received resumes and/or
28 resume-derived information; and a user profiles database 2437, storing user accounts,
29 user information, and/or the like. The CSE controller 2420 may further be coupled to
30 an application interface 2445 configured to process for and/or relay CSE processed data

1 to one or more external applications (Application 1, Application 2, . . . , Application m)
2 2450.

3 **[00296]** FIGURE 25 shows an implementation of application modules and
4 databases communicatively coupled to the CSE 2501 in one embodiment of CSE
5 operation. The illustrated CSE Application overview includes Career Path Modeling
6 2530, as well as Career Path User Interface system 2540 features driven data processed,
7 analyzed and coordinated by the underlying CSE 2501 and/or associated Databases
8 2505. In various embodiments, Career Path Modeling 2530 may be based on path-
9 dependent 2532 and/or path independent 2534 state model implementations and/or
10 may further couple to a recommendation/recruiter engine 2536. Similarly, in various
11 embodiments, Career Path UI Modeling 2540 may be based on path-dependent 2542
12 and/or path independent 2544 state model implementations and/or may further couple
13 to a recommendation/recruiter engine 2546 The CSE 2501 may also coordinate Career
14 Data Structure Adaptation 2550 and Career Benchmarking 2555 features. The CSE
15 manages data associated with various system processes in CSE Databases 2505 that
16 include State Model database 2510, Taxonomy database 2520 and Attribute Database
17 2515 information, as well as the underlying Video 2522, People 2524, Ads 2526, and
18 other content 2528 that may be incorporated into various implementations of the
19 system. Further, in some implementations, the CSE Databases also coordinates the
20 relationships/associations between these modules, as well.

21 **[00297]** FIGURE 26A shows an implementation of combined logic and data flow
22 for acquiring and processing career data inputs in one embodiment of CSE operational.
23 A plurality of individual career data inputs 2601, such as resume data, profile data,
24 and/or the like may be input to a career data aggregation module 2605. In one
25 implementation, free-form resume data may be parsed by an automated resume parses
26 2610, such as may be based on resume templates. In another implementation, resume
27 data may be input as structured inputs in an online structured resume data entry
28 module 2615, such as a web form interface admitting experiential, educational, and/or
29 the like resume data inputs from users. In another implementation, future or
30 prospective employment information may be entered via an online future employment
31 data entry module 2620. In another implementation, user profile information may be

1 entered 2621, such as may be received from a user profile database. In one
2 implementation, a seed set of raw seeker data (e.g., of structured resume data) may be
3 processed initially by the CSE to yield an initial state model, topic model, and/or the
4 like. For each job seeker 2625, the CSE may read in raw seeker data 2630, such as
5 resume data, profile data, and/or the like. Elements of the raw seeker data, such as job
6 titles, start and/or end dates of employment experiences, and/or the like may then be
7 processed to discern a plurality of job state classifications, job states, states, and/or the
8 like 2635. In one implementation, statistical analysis of raw seeker job titles and/or
9 other work experience data may be undertaken by a statistical analysis toolkit, such as
10 by the Mallet toolkit available at <http://mallet.cs.umass.edu>, to discern job states
11 and/or other classifications. Elements of the raw seeker data, such as work experience
12 descriptions, may further be processed to discern a plurality of topics and associated
13 terms and/or phrases 2640. For example, in one implementation, job seeker work
14 experience description data may be processed by elements of the Mallet toolkit to
15 discern a plurality of topics comprising common terms and/o phases appearing in those
16 descriptions. Discerned states and/or topics may then be coalesced into a state model,
17 and the state model may be stored in a database 2645. A determination may then be
18 made as to whether there is additional seeker processing to undertake 2650. If so, then
19 the CSE may return to 2625. Otherwise, the CSE may proceed to determining and
20 assigning topic weights to states in the state model, as shown in one implementation in
21 Figure 26B.

22 **[00298]** FIGURE 26B shows an implementation of combined logic and data flow
23 for processing career data inputs, in one embodiment of CSE operation, to determine
24 and assign topic weights to states in a state model. For each state of the plurality of
25 states discerned by the statistical analysis toolkit in Fig. 26A, a weight may be assigned
26 to each topic of the plurality of topics also discerned by the toolkit in Figure 26A.
27 Weights may, in one implementation, be based on the frequency with which terms
28 associated with topics appear in descriptions for resume work experiences associated
29 with states. For each state in the state model 2655, the CSE may determine work
30 experiences, work experience data structures, and/or the like associated to the state
31 2660. In one implementation, such a determination may be made based on information

1 stored in or by the statistical analysis toolkit from Figure 26A, the information being
2 generated as part of the classification of work experiences and the discernment of states
3 at 2635. The CSE may then parse terms from descriptions associated with the work
4 descriptions 2665 in order to match those terms against terms associated with topics
5 2670. In this manner, the CSE may determine which work experiences corresponding
6 to a given state also correspond to a given topic or set of topics. The CSE may then
7 count the number of work experiences for a given state that match a given topic 2675
8 and divide by the total number of work experiences associated with the state to
9 determine the frequency, and accordingly the weight, to assign to that given topic in
10 association with that given state 2680. The determined weights may then be associated
11 with their corresponding topics within the state record for the given state 2685. The
12 CSE may then store the state model with states and topics, including weights assigned to
13 topics in association with each state, in a database 2690. A determination may then be
14 made as to whether additional processing of job seeker data is warranted 2695. If so,
15 the CSE may return to 2655 and update topic weights. Otherwise, the CSE may proceed
16 to building a state data record, such as shown in one implementation in Figure 28.

17 **[00299]** FIGURE 27A shows a schematic illustration of resume data record
18 generation in one embodiment of CSE operation, whereby a submitted resume may be
19 mapped to states, topics, and/or the like using the state model generated according to
20 Figs. 26A-26B. A submitted resume 2701 may contain a variety of information
21 describing experiences, attributes, and/or the like associated with a job seeker. The
22 resume 2701 in the illustrated implementation includes user contact information 2703
23 (e.g., postal address, e-mail address, phone numbers, and/or the like), a work
24 experience sequence 2706 comprising job titles 2709 and description 2712, a list of skills
25 2715, a list and/or description of education experiences (e.g., schools attended, degrees
26 received, grades, courses, and/or the like) 2718, a section listing and/or describing
27 languages spoken 2721, and/or the like. A state model 2724 may serve to process
28 resume 2701 data into one or more data records 2731 configured for analysis and/or
29 processing by CSE modules. In one implementation, the state model 2724 may process
30 resume 2701 information in conjunction with user profile information 2728 and/or
31 education information 2729 to generate the one or more data records 2731. The state

1 model 2724 may, for example, analyze job titles 2709 and/or descriptions 2712 in order
2 to map them to a pre-set listing of job “states”. The work experience listing 2706, thus,
3 may be converted into a state sequence 2736 comprising a plurality of states 2739
4 associated with the job titles 2709 and/or descriptions 2712 from the resume 2701.

5 **[00300]** Furthermore, an attributes model 2727 may receive and/or process other
6 resume information, such as that external to the work experience listing 2706, to
7 generate elements of a data record configured to analysis and/or use by other CSE
8 components. The attribute model 2727 may further be configured to consider education
9 2718 and/or relational taxonomy 2730 inputs, in addition to the other resume
10 information, in generating those elements. In one implementation, the attribute model
11 may map resume information to elements of a pre-set listing of attributes. Thus, the
12 skills 2715, education 2718, languages spoken 2721, and/or the like extracted from the
13 resume 2701 may be converted into an attributes listing 2742 comprising a plurality of
14 attributes 2745 corresponding to various elements of the resume information. Other
15 resume information may also be included in a resume data record 2731, such as may be
16 collected in an "Other" category 2748 for subsequent reference and/or use. The resume
17 data record 2731 may be associated with a unique resume identifier (ID) 2733, based on
18 which the record may be queried and/or otherwise targeted.

19 **[00301]** FIGURE 27B shows a schematic illustration of experience to state
20 conversion in one embodiment of CSE operation, whereby an input resume may be
21 converted and/or otherwise mapped to states, topics, and/or the like using the state
22 model generated according to Figs. 26A-26B. Experiences listed in a resume may be
23 processed by one or more CSE state models to convert those experiences to at least one
24 of a list of pre-defined states. In some cases, job seekers may use the same or similar job
25 titles and/or descriptions to refer to jobs that may be very different and/or that may
26 correspond to different states within the CSE state model. Figure 27B provides an
27 illustration of CSE state resolution for similar resume work experience listings.
28 Experience listings at 2751 and 2760 each comprise the job title "Operations Manager",
29 but have different job descriptions. The CSE state model 2754 may include a plurality of
30 states, each having a plurality of corresponding job titles, and the CSE may employ the
31 model to determine which, if any, of the states have titles matching the titles supplied at

1 2751 and/or 2760. In one implementation, different states may have common
 2 corresponding job titles. To resolve the appropriate state corresponding to each of the
 3 work experience listings 2751 and 2760, the CSE may analyze the listings' job
 4 description field for comparison with "topics" associated to each state. The job
 5 description in the listing at 2751 includes keywords "shipping" and "receiving" that
 6 match topics in the state model 2754 entry corresponding to the state "Manufacturing
 7 Operations Manager" with state number 23418, so the listing 2751 is mapped to this
 8 unique state 2757. The job description in the listing at 2760, on the other hand, includes
 9 the keywords "personnel" and "schedules", matching topics in the state model 2754
 10 entry for the state entitled "Staffing Operations Manager" with state number 52154, so
 11 the listing 2760 is mapped to this unique state 2766. In one embodiment, a state
 12 structure may be represented by way of database tables. In another embodiment, a state
 13 structure, or limited subset thereof, may be represented as XML information, which
 14 may be used for advancement pathing. In one embodiment, the XML structure may take
 15 the following form:

16

```

17 <states>
18   <state id="0" njobs="3712" ntokens="90708">
19     <title>cna, certified nursing assistant, caregiver</title>
20     <jobtitles>
21       <jobtitle count="260" pct="7.0">cna</jobtitle>
22       <jobtitle count="142" pct="3.8">certified nursing assistant</jobtitle>
23       <jobtitle count="104" pct="2.8">[no title]</jobtitle>
24       <jobtitle count="83" pct="2.2">caregiver</jobtitle>
25       <jobtitle count="67" pct="1.8">home health aide</jobtitle>
26     ...
27       <jobtitle count="15" pct="0.4">residential counselor</jobtitle>
28     </jobtitles>
29     <topics>
30       <topic id="494" n="32601" words="care residents home daily living
31 patients personal nursing aide activities " />
32       <topic id="696" n="1719" words="patients patient medical insurance
33 appointments charts doctors doctor procedures care " />
34     ...
35       <topic id="205" n="544" words="daily basis needed reports activity log
36 assist interacted complete schedule " />
37     </topics>
38     <next>
39       <state id="0" pct="10.5" titles="cna, certified nursing assistant,
40 caregiver" topics="care patients treatment career care unit medical activities
41 children daily" />

```

```

1      <state id="268" pct="4.6" titles="cna, certified nursing assistant,
2      caregiver" topics="care cleaning job job assist helped shift duties clean
3      food" />
4      ...
5      <state id="45" pct="1.1" titles="medical records clerk, medical
6      transcriptionist, file clerk" topics="medical records answered phones office
7      answer office patients data data" />
8      </next>
9      <prev>
10     <state id="999" pct="23.0" titles="[First job]" topics="[First job]" />
11     <state id="0" pct="10.2" titles="cna, certified nursing assistant,
12     caregiver" topics="cna, certified nursing assistant, caregiver" />
13     ...
14     <state id="243" pct="0.9" titles="administrator, executive director, director
15     of nursing" topics="administrator, executive director, director of nursing" />
16     </prev>
17     </state>
18     <state id="1" njobs="3570" ntokens="113569">
19     ...
20     </state>
21     <states>
22

```

23 **[00302]** The XML form including a title, other analogue job titles and related
24 frequency counts and likelihood percentages, topics, next states and previous states with
25 frequency occurrences, and/or the like.

26 **[00303]** Job listings with different job titles may also be mapped to the same state
27 by a CSE state model 2754. The listing at 2769 includes a job title of "Facilities
28 Manager", which matches one of the titles for the state "Manufacturing Operations
29 Manager" (though possibly other states as well) in the CSE state model 2754. The listing
30 2769 further includes a job description comprising the keywords "shipping" and
31 "receiving", which match topics associated with the state "Manufacturing Operations
32 Manager", so the listing 2769 is mapped to the unique state 2775, which is the same as
33 the state at 2757 despite the different job title in the original listing.

34 **[00304]** FIGURE 27C shows an implementation of logic flow for experience to state
35 conversion in one embodiment of CSE operation. The logic in Figure 27C may be
36 applied, for example, to a work experience listing extracted from a resume or curriculum
37 vitae (CV). In alternative implementations, the logic in Figure 27C could be applied to
38 job listings from other sources, such as career development resources, school and/or
39 corporate websites, and/or the like. A job title may be queried and/or extracted from
40 the listing 2776 and compared with a plurality of job titles corresponding to states in the

1 state model 2777 in order to determine whether there exist any states having matching
2 job titles 2778. If there are no matches, then the CSE may engage an error handling
3 procedure, try approximate matching of the job titles, and/or the like 2779. For
4 example, in one implementation, the CSE may perform a search based on a subset of the
5 complete job listing job title to find approximate matches. In another implementation,
6 the CSE may seek states having job titles with subsets matching the job title extracted
7 from the job listing (e.g., a state model job title of "Manufacturing Operations Manager"
8 may be deemed a match for an input job title of "Manufacturing Manager"). In still
9 another implementation, an error message may be returned for the input job title
10 and/or the job title may be set to a null state.

11 **[00305]** If one or more matches are established at 2778, a determination may be
12 made as to whether there are multiple matching states 2780. If there is only one
13 matching state, then the CSE may immediately map the input listing to the matching
14 state 2781. Otherwise, the CSE may query and/or extract a job description from the
15 input listing 2782 and parse key terms from that description 2783. Parsing of key terms
16 may be accomplished by a variety of different methods in different implementations
17 and/or embodiments of CSE operation. For example, in one implementation, the CSE
18 may parse all terms from the description having more than a minimum threshold
19 number of characters. In another implementation, the CSE may filter all words in the
20 description that match elements of a listing of common words/phrases and extract the
21 remaining words from the description. The parsed key terms may then be compared at
22 2784 to state model topics corresponding to the matching states determined at 2777-
23 2778. A determination is made as to whether there exist any matches between the job
24 description terms and the state topics 2785 and, if not, then one or more error handling
25 procedures may be undertaken to distinguish between the matching states 2786. For
26 example, in one implementation, the CSE may choose a state randomly from the
27 matching job states and map the input listing thereto. In another implementation, the
28 CSE may present a job seeker, system administrator, and/or the like with a message
29 providing a selectable option of the various matching states, to allow for the selection of
30 a desired match.

1 **[00306]** If a match exists at 2785 between description key terms and state topics in
2 the CSE state model, then a determination may be made as to whether there exists more
3 than one matching state 2787. In one implementation, this determination may only find
4 that multiple matches exist if the number of key terms matching state topics is the same
5 for more than one state (i.e., if one state has more matching topics than another, then
6 the former may be deemed the unique matching state). If there are not multiple
7 matching states, then the input listing may be matched to the unique matching state
8 2789. If, on the other hand, multiple matches still exist, then the CSE may, in various
9 implementations, undertake any of a variety of different methods of further discerning a
10 unique matching state for the input listing. For example, in one implementation, the
11 CSE may choose randomly between the remaining states. In another implementation,
12 the CSE may provide a list of remaining states in a message to a job seeker, system
13 administrator, and/or the like to permit selection of a desired, unique state. In another
14 implementation, the CSE may map the job listing to all of the multiple matching states.

15 **[00307]** In one implementation, logic flow similar to that described in Figure 27C
16 may be employed to map other resume information, such as education experiences,
17 skills, languages spoken, honors and/or awards, travel, and/or the like to one or more
18 attribute states stored in and/or managed by the CSE, a CSE state model, a CSE
19 attribute model, and/or the like.

20 **[00308]** FIGURE 27D shows an implementation of a raw resume data record and a
21 state converted resume data record in one embodiment of CSE operation. The raw
22 resume data record 2790 is indexed by a resume ID 2791, and includes a variety of
23 resume data, including contact information 2792, a job sequence listing 2793, and other
24 information 2793 such as education, skills, honors/awards, and/or the like. The
25 corresponding state converted resume data record is shown at 2795, and includes a state
26 sequence 2796 corresponding to the job sequence 2793, as well as a series of attributes
27 2797 that are based on the other resume information. The state converted resume data
28 record also may incorporate other resume data 2797.

29 **[00309]** FIGURE 28 shows an implementation of combined logic and data flow for
30 building a state data record in one embodiment of CSE operation. For each job seeker
31 data record 2801, such as may correspond to resume and/or profile data submitted by

1 the job seeker, the CSE may process the seeker record to create and/or update one or
2 more state models and/or data tables 2805. An example of such data processing in one
3 implementation is shown at 2810, wherein a unique state ID is created and state data is
4 mapped thereto. Associated with the state ID may be one or more job titles, topics
5 and/or topic IDs, skills, education information, salary information, experience
6 information, length and/or time at a job, and/or the like. The state record may further
7 include links to next state IDs, previous state IDs, and/or external database links, such
8 as to associated videos, people and/or profiles, ads, and/or other content. The state
9 record may be stored in and/or used to update the state model for storage in a state
10 model database 2815. A determination may then be made as to whether additional state
11 processing is to be undertaken 2820. If so, then the CSE may return to 2801 and draw
12 on the next seeker data record. Otherwise, the CSE may move to processing state data
13 to develop the statistical database and/or perform incremental state discovery, such as
14 by the embodiments shown in Figs. 29A-29B.

15 **[00310]** FIGURE 29A shows an implementation of combined logic and data flow
16 for processing state data to develop the statistical model in one embodiment of CSE
17 operation. For each state data record 2901, the CSE may update a career path and/or
18 state model topology and/or topology weights based on analysis of the data record 2905.
19 In one implementation, a career path and/or state model topology may comprise a
20 plurality of relationships between job states established based on the frequency of
21 occurrence of such relationships in the work experience sections of analyzed resumes.
22 The CSE may also be configured to add new nodes to the career path and/or state model
23 topologies as necessary 2910, such as if a newly analyzed resume introduces a
24 relationship between job states that had not been seen in previously analyzed resumes.
25 The updated career path and/or state model topology may be stored in a database 2915
26 and a determination made as to whether additional statistical analysis is required 2920.
27 If so, then the CSE returns to 2901 and proceeds with additional statistical analysis of
28 the state data record and/or moves to a next state data record. Otherwise, the career
29 path and/or state model topology may be provided for access and/or use by other career
30 path features and/or functions 2925.

1 **[00311]** FIGURE 29B shows an implementation of combined logic and data flow
2 for processing state data to develop the statistical model in another embodiment of CSE
3 operation.

4 **[00312]** For each state data record 2930, the CSE may analyze the record using any
5 of a variety of statistical analysis tools. Numerous methods of topic modeling may be
6 employed as discussed in: "Latent Dirichlet Allocation," D. Blei, A. Ng, M. Jordan, "The
7 Journal of Machine Learning Research", 4303. Markov models may also be employed as
8 discussed in: "A tutorial on hidden Markov models and selected applications in speech
9 recognition," L. Rabiner, Proceedings of the IEEE, 1989. In one embodiment, Mallet
10 Processing tools 2935 may also be employed, such as may be found at
11 <http://mallet.cs.umass.edu>. The analysis may include aggregation and/or analysis of
12 user individual state records 2940, aggregation and/or analysis of user state chain
13 records 2945, and/or aggregation and/or analysis of user historical parameter(s) 2950.
14 User historical parameters 2955 may, for example, comprise salary, location, state
15 experience duration, subjective experiences associated with job states, benefits, how the
16 job was obtained, other benchmarking and/or user generated content, and/or the like.
17 The statistics associated with the state record may be summed 2960 and added to the
18 state statistical records in one or more state models stored in a state model database
19 2965. A determination may be made as to whether additional statistical analysis of state
20 data records is to be undertaken 2970. If so, then the CSE may return to 2930 to
21 proceed with additional analysis of the state data record and/or to move to the next
22 state data record. Otherwise, the state model may be provided for path modeling 2975,
23 benchmarking 2980, and/or the like applications.

24 **[00313]** FIGURE 30 shows an implementation of logic flow for development of a
25 path-independent statistical model in one embodiment of CSE operation. In one
26 embodiment, a path-independent statistical model may comprise a collection of job
27 states, each having corresponding probabilities for most likely next and previous job
28 states, wherein the probabilities depend only on the job state itself and not on any prior
29 history of job states. A resume, profile data, and/or the like may be received at 3001,
30 such as from a resume database, and a job and/or other work experience sequence
31 extracted therefrom 3005. Jobs from that sequence may be mapped to corresponding

1 states in a CSE state model 3010, such as according to the embodiments described in
2 Figs. 27A-D. Then, for each job state in the sequence 3015, the CSE may query a next
3 job state (J_n) and a previous job state (J_p) in the sequence 3020, where a null state may
4 be set to J_p for the first state in the sequence and to J_n for the last state in the sequence.
5 A state record corresponding to the current state under consideration (3015) may be
6 retrieved 3025, such as from a CSE state model, and a determination may be made as to
7 whether J_n exists in the state record 3030. For example, the CSE may seek J_n in a listing
8 of common next job states corresponding to the given job state. If J_n does exist in the
9 record, then a number of occurrences, $N(J_n)$, of J_n as a next state for the state under
10 consideration may be incremented 3035. Otherwise, J_n may be appended to the listing
11 of next states for the state under consideration 3040 and a value for the number of
12 occurrences of J_n initialized 3045.

13 **[00314]** The CSE may also determine whether J_p exists in the state record, such as
14 in a listing of common previous job states corresponding to the state under
15 consideration 3050. If so, then a number of occurrences, $N(J_p)$, of J_p as a previous state
16 for the state under consideration may be incremented 3055. Otherwise, J_p may be
17 appended to the listing of previous states for the state under consideration 3060 and a
18 value for the number of occurrences of J_p initialized 3065. The CSE may then increment
19 a total number, N_{tot} , associated with the number of resumes used to update the
20 particular state entry of the path-independent statistical model 3070. The CSE may
21 then determine probabilities corresponding to J_p and J_n by dividing $N(J_p)$ and $N(J_n)$
22 each respectively by N_{tot} 3075. These probabilities may, for example, provide an
23 indication to job seekers of the likelihood of changing to or from a job from another job,
24 based on the accumulated resume records of other job seekers who have held those jobs.
25 The state record with the updated probability values may be persisted at 3080, such as
26 by being stored in a database.

27 **[00315]** FIGURE 31 shows an implementation of a path-independent state model
28 data record in one embodiment of CSE operation. The data record in Figure 31 may, for
29 example be generated and/or updated by the logic flow shown in Figure 30. The data
30 record may, in one implementation, correspond to a unique job state in the CSE state
31 model, indexed by a state ID 801. A listing of next states 3105 may include a plurality of

1 states and corresponding probabilities 3110, such as may be determined according to
2 the logic in Figure 30. Similarly, the data record may include a listing of previous states
3 3115 comprising states and corresponding probabilities 3120. Additional data
4 associated with the state may be stored in the state record 3125, such as but not limited
5 to a total number of resumes analyzed for the state in question, a confidence metric
6 describing confidence in and/or reliability of the probabilities in 3110 and/or 3120, state
7 job titles, state topics, and/or the like.

8 **[00316]** FIGURE 32 shows an implementation of logic flow for development of a
9 path-independent statistical model with attributes in one embodiment of CSE
10 operation. In one embodiment, a path-independent statistical model with attributes
11 may comprise a collection of job states, each having corresponding probabilities for
12 most likely next and previous job states, wherein the probabilities depend on the job
13 state itself and on the identity of one or more associated attributes, but not on any prior
14 history of job states. Resume and/or profile data may be received at 3201, and a job
15 experience sequence may be extracted therefrom 3205. Jobs from the job sequence may
16 then be mapped to CSE state model job states at 3210, such as according to the
17 embodiments described in Figs. 27A-D. The CSE may also extract additional resume
18 data 3215, such as but not limited to education levels, schools attended, awards and/or
19 honors, skills, languages spoken, number of years of experience, salary levels,
20 certifications and/or licenses possessed, and/or the like. The additional resume data
21 may be mapped to attribute states in the CSE state and/or attribute model 3220, such as
22 in a manner similar to mapping of job sequence listings to job states. Then, for each
23 state in the sequence of job states 3225, the CSE may query the next job state (J_n) and
24 previous job state (J_p) in the sequence 3230, with a null state assigned to J_n for the last
25 state of the sequence and to J_p for the first state of the sequence. The CSE may also
26 retrieve the state record for the current state 3235.

27 **[00317]** Then, for each attribute in the resume 3240, the CSE may determine
28 whether J_n exists in the state record in correspondence with that attribute 3245, such as
29 in a listing of common next job states corresponding to the state and attribute under
30 consideration. If so, then a number of occurrences, $N(J_n)$, of J_n as a next state for the
31 state and attribute under consideration may be incremented 3250. Otherwise, J_n may

1 be appended to the state record in association with the particular attribute 3255 and a
2 value for the number of occurrences of J_n initialized 3260. A determination may then be
3 made as to whether J_p exists in the state record in correspondence with the attribute
4 under consideration 3265. If so, then the number of occurrences, $N(J_p)$, of J_p as a
5 previous state for the state and attribute under consideration may be incremented 3270.
6 Otherwise, J_p may be appended to the state record in association with the particular
7 attribute 3275 and a value for the number of occurrences of J_p initialized 3280. A total
8 number of instances may then be incremented 3285, and probabilities for J_n and J_p ,
9 corresponding to the proportion of resumes having the attribute under consideration
10 and those job states respectively before and after the job state under consideration, may
11 be determined as the ratio of each of $N(J_n)$ and $N(J_p)$ with N_{tot} 3290. The state record,
12 with updated probability values, may then be persisted at 3295, such as by storing the
13 record as part of a CSE state model in a database.

14 **[00318]** FIGURE 33 shows an implementation of a path-independent model with
15 attributes data record in one embodiment of CSE operation. The data record in Figure
16 33 may, in one implementation, be generated by a method similar to that shown in
17 Figure 32. The record, corresponding to a particular job state, may be identified by a
18 unique state ID 3301, and may further include a plurality of attributes (3305, 3330).
19 Each attribute, in turn, may include listings of next states 3310 and of previous states
20 3320, states comprising states and associated probabilities (3315, 3325), such as may be
21 determined by the method described in Figure 32. In one embodiment, a hierarchy of
22 states may be generated by traversal of interconnected state structures.

23 **[00319]** FIGURE 34 shows an illustration of career path modeling using path-
24 independent and path-dependent statistical models in one embodiment of CSE
25 operation. The CSE may, in some embodiments, operate to take one or more job inputs
26 and return a job output, wherein the job output comprises a prediction of a most likely
27 next job and/or otherwise statistically noteworthy result based on the inputs. In Figure
28 34, a user may provide an experience sequence comprising five jobs (J_1 3405, J_2 , 3410,
29 J_3 3415, J_4 3420, J_5 3401) as inputs to the CSE. In one embodiment, the CSE
30 comprises a path-independent model 3425 that may generate an output job state J_6
31 3430 based only on a single job state (e.g., J_5 3401). In an alternative embodiment, the

1 CSE comprises a path-dependent model 3435 that takes multiple jobs as inputs (J1
2 3405, J2, 3410, J3 3415, J4 3420, J5 3401) to generate and return an output job state J6
3 3440. The embodiments described in Figs. 30-33 are directed to generation of the path-
4 independent CSE state model.

5 **[00320]** FIGURE 35 shows an implementation of logic flow for development of a
6 path-dependent statistical model in one embodiment of CSE operation. In one
7 embodiment, a path-dependent statistical model may comprise a collection of job states,
8 each having corresponding probabilities for most likely next and previous job states,
9 wherein the probabilities depend on the history of jobs leading up to the most recent job
10 state. Though Figure 35 is directed to an implementation bereft of attribute
11 consideration, it should be recognized that aspects of Figs. 32-33 could be incorporated
12 to yield an attribute-sensitive, path-dependent state model. The CSE may receive
13 resume data, profile data, and/or the like, such as from a resume database, at 3501, and
14 extract a job sequence comprising a plurality of jobs (J1, J2, ..., JN) therefrom 3505
15 which may subsequently be mapped to a sequence of job states. Then, for each state (J_i)
16 in the sequence 3520, the CSE may retrieve a state record corresponding to state J_i 3525
17 and set an indexing variable, m, equal to i+1 3530. A determination may then be made
18 as to whether a field corresponding to the state J_m exists in the J_i state record 3535. If
19 so, then a number of occurrences, N_{i..m} of that sequence of job states (J_i to J_m), is
20 incremented 3540. Otherwise, the J_m field is appended to the state record 3545, and the
21 value of a number of occurrences corresponding to the job sequence is initialized 3550.
22 A total number, N_{toti..m}, of instances (e.g., the number of resumes analyzed) may then
23 be incremented 3555, and a probability for the sequence determined by dividing the
24 number of occurrences of the job sequence by the total number of instances 3560. A
25 determination may then be made as to whether there are more states to analyze in the
26 job sequence 3565. If so, the indexing variable m is incremented 3570, and the CSE
27 returns to 3535. Otherwise, when all states in the sequence are exhausted, the J_i state
28 record is persisted, and the CSE moves to the next job state at 3520 (e.g., by
29 incrementing i) 3575.

30 **[00321]** To further illustrate the embodiment described in Figure 35, the following
31 example may be considered. A resume may include a work experience history

1 comprising a sequence of three jobs: J1, J2 and J3. The logic in Figure 35 would first
2 update a CSE state model based on the job sequence J1 to J2, specifically updating the
3 probability associated with J2 in a J1 state record. Next, the CSE state model would
4 update a probability associated with J2 to J3 in the J1 state record. Then, the CSE state
5 model would retrieve the J2 state model and update a probability associated with J3
6 therein. In this manner, the CSE state model will contain information pertaining to
7 probabilities of all the sequences and sub-sequences of the work experience listings in
8 the resumes that it analyzes (in this exemplary case, those sequences and sub-sequences
9 are: J1, J2; J1, J2, J3; and J2, J3).

10 **[00322]** FIGURE 36 shows an implementation of a path-dependent statistical
11 model data record in one embodiment of CSE operation. The state data record shown in
12 Figure 36 may, in one implementation, be generated by a logic flow similar to that
13 shown in Figure 35. The state, here labeled A, to which the data record corresponds
14 may be identified by a state record identifier 3601. The state record may further include
15 a second tier of "next job" states, each characterized by at least a state identifier and a
16 probability 3605. For example, in Figure 36, a next state is labeled B and has a
17 probability labeled AB corresponding to the proportion of resumes analyzed wherein an
18 individual having job A moved to job B. Under each of the second tier states, there may
19 further exist third tier states 3610, fourth tier states 3615, etc., each including at least a
20 state identifier and a probability associated with the sequence leading to the current
21 state from each state in the higher tiers. For example, in Figure 36, the state labeled D
22 at the fourth tier shown at 3615 is associated with a probability labeled ABCD that
23 characterizes the proportion of resumes wherein an individual had the sequence of jobs
24 A, B, C and D.

25 **[00323]** FIGURES 37A-B show an implementation of logic flow for development
26 and of a path-dependent statistical model in another embodiment of CSE operation.
27 Models similar to that shown in Figs. 37A-B may, in some implementations, include a
28 two-stage method, the first comprising a setup stage wherein the model is established as
29 a collection of job state couplets (Figure 37A), and the second comprising an application
30 of the model to a specific job sequence and/or target job state to yield a target job state
31 probability (Figure 37B). Resume data, profile data, and/or the like is received at 3701

1 and a job sequence (J1, J2, ..., JN) is extracted 3705. The job sequence may then be
2 converted into corresponding job states (JS) 3710, such as according to the
3 embodiments described in Figs. 27A-D. Then, for each JS in the sequence 3720, the
4 CSE may read the JS and the next state (NS) in the sequence to establish a couplet
5 comprising a pointer between JS and NS 3715. The state model may then be queried
6 3720 to determine whether a match exists to the JS/NS couplet 3725. If not, then an
7 entry may be created in the state model corresponding to the couplet 3730 or, if so, then
8 the number of instances for that couplet's may be incremented 3735. The couplet entry
9 may be stored 3740 in association with a user ID, resume ID, and/or other identifier
10 associated with the resume from 3701, as well as with a JS sequence number (n),
11 associated with the position of JS in the job sequence from 3705. A determination may
12 then be made as to whether additional states exist in the sequence 3745 and, if so, the
13 CSE may return to 3715 to analyze the next sequence state.

14 **[00324]** Figure 37B illustrates an implementation of logic flow for application of
15 the state model to obtain a probability associated with a given target job state given a
16 sequence of past job states, in one embodiment of CSE operation. The target job state is
17 obtained at 3750, and the sequence of past job states, comprising a plurality of couplets
18 of job state and next state, is entered 3755. Then, for each couplet or pair, the state
19 model may be searched 3757 to obtain matching pairs 3759. The CSE may then apply a
20 filter to extract desired and/or relevant matches. For example, the CSE may query the
21 CSE 3761 to obtain results 3763 out of the matching pairs from 3759 that have common
22 associated UserIDs across pairs. For example, the CSE may have found pairs (A, B) and
23 (B, C) at 3757-3759, corresponding to jobs A, B and C. To establish that the sequence A
24 to B to C exists for any specific users, the CSE could then seek common user IDs existing
25 in both the (A, B) and (B, C) records.

26 **[00325]** The CSE may also want to ensure that the sequence exists in the proper
27 order. For example, if a common user ID exists in the (A, B) and (B, C) records, this
28 does not necessarily imply that a user has the specific job sequence A to B to C in their
29 resume and/or profile data. The user may, instead, have a sequence such as B to C to A
30 to B. The CSE may, therefore, query results for proper JS chain sequence ordering 3765,
31 such as may be based on the JS sequence number (n) stored at 3740.

1 **[00326]** The CSE may thus obtain 3767 and count 3769 the non-targeted results,
2 that is the single-resume job sequence matches to the JS existing chain from 3755, but
3 not including the target state from 3750. The CSE may then search the state model 3771
4 to obtain "goal results" 3773 comprising couplets of the last state in the JS existing chain
5 with the target state. A filter process similar to that shown at 3761-3765 may then be
6 applied to the sequence comprising the non-targeted results plus the goal results 3775.
7 The number of filtered goal results are counted 3777 and the ratio of the number of goal
8 results to the number of targeted results may be computed 3779, stored, and/or the like.
9 This ratio may be interpreted as the proportion of analyzed resumes having the
10 sequence of jobs corresponding to the JS existing chain from 3755 leading into the
11 target job state from 3750.

12

APT

13 **[00327]** FIGURE 38 is of a mixed block, data and logic flow diagram illustrating
14 embodiments of APPARATUSES, METHODS AND SYSTEMS FOR ADVANCEMENT
15 PATH TAXONOMY (hereinafter "APT"). From a high level, the APT 3801 allows users
16 (e.g., advancement "seekers") 3833a to interact with APT servers 3802 through
17 interfaces on their client(s) 3833b across a communications network 3813. Although the
18 following discussion will frequently use examples of seekers wishing to advance their
19 careers, it should be noted, that such seekers may similarly use the APT to advance their
20 educational achievement, their financial goals, and/or the like. To that end, seekers
21 3833a may provide 3833b relevant (e.g., job) experiences they have had leading up to
22 their current desire to seek advancement beyond their past and current experiences
23 3805, 3806, 3807 (hereinafter "experience information") to the APT. Similarly, seekers
24 3833a may provide 3833b targeted advancement milestones, objectives and/or goals
25 (hereinafter "advancement information" or "target information") to the APT. In turn,
26 the APT 3801 may obtain that advancement experience information 3810 and use that
27 information 3802 to provide the seeker with next states in their advancement goals
28 3809.

29 **[00328]** Upon obtaining the user advancement experience information 3810, the
30 APT may analyze the experience information (e.g., and perhaps other information

1 associated with the user found in the user's profile) against a state structure 3812. By
2 analyzing the advancement seeker's experiences and goals against a statistical state
3 structure, the APT may determine what next states 3814 may form the advancement
4 seeker's next advancement milestone(s) and/or paths to their desired milestones and/or
5 advancement goals 3809. It should be noted that in one embodiment, the state structure
6 may take the form of generated by the CSE. In one embodiment, the state structure is
7 stored in APT state structure database table(s); as such, the state structure may be
8 queried with advancement experience information, advancement information,
9 experience information, state identifier (e.g., state_ID), proximate state identifier (e.g.,
10 next_state_ID), topics/terms, topic_ID, and/or the like. When queried, the state
11 structure may return state records (i.e., states) that best match the query select
12 commands, and those states may themselves further refer to other proximate states;
13 where the proximate states are related advancement states (hereinafter "adjacent state,"
14 "advancement state," "next state," "proximate state," "related state," and/or the like)
15 that may include likelihoods of moving from the state to the related advancement state.
16 Upon determining what next states may form the advancement path and/or milestone
17 for the seeker 3814, the APT may generate a user path topology showing the user their
18 advancement path. This topology may be used to update the seeker's client 3833b
19 display 3818 with an interactive (e.g., career) advancement path.

20 **[00329]** FIGURE 39 is of a logic flow diagram illustrating embodiments of the APT.
21 A user need not be, but may be, logged in to an existing account to the APT to make use
22 of its advancement pathing abilities 3901. In either case the user will engage the APT
23 (for example, in a web embodiment of the APT); a user may engage the APT by
24 navigating their web browser to an address referencing the APT's information server,
25 which will act as an interface/gateway between the seeker and APT. It should be noted
26 that a web interface is one of many interface and/or mechanisms by which the APT may
27 be deployed and/or implemented; for example, in alternative embodiments the APT
28 may be a stand alone application, a server messaging system that accepts inputs and
29 provides outputs to disparate clients, etc. In accessing the APT 3901, the seeker may
30 start to provide experience advancement information. The experience advancement
31 information may include both desired advancement milestones and/or goals (although

1 this is not required) and experience information, which includes experience the seeker
2 already has. The experience information may be provided by way of submission of
3 structured information via a web form, parsing submitted resume's (e.g., via attachment
4 and/or uploading of a resume file), aggregating experience information in a profile over
5 time, allowing the user to select a pre-existing state matching their own (e.g., letting
6 them find a job/title/occupation matching their current occupation) in graph topology
7 representing an hierarchical interconnected state structure (e.g., see 4705 of Figure 47),
8 and/or the like. In one example embodiment, the user may submit current work
9 experience via web form, which may include: the dates of employment, the employer's
10 name (e.g., employing company), seeker title/position, descriptive resume information
11 about their employment, and/or the like 3902. In another embodiment, the seeker may
12 submit experience information beyond their instant post that includes: previous
13 positions, their educational background, and/or the like 3902. In addition, the seeker
14 may similarly provide their advancement information. For example, the seeker may
15 provide that they currently have the title of Retail Administrator, without more, and see
16 what are the next most likely career path opportunities from that role, without having
17 any explicit advancement goal. However, should the seeker also provide a milestone
18 and/or goal, e.g., Manager of a retail chain, then the APT will construct paths and
19 experiential states that show the seeker's the different routes by which the seeker may
20 advance to their desired milestone/goal. It should be noted that build/find path facilities
21 that are described are not exclusive mechanisms for building paths, and browsing
22 through the topology is also supported as will be detailed in further figures.

23 **[00330]** Depending on the information supplied by the seeker and the seeker's
24 desire to see advancement path variations, the APT may provide at least three different
25 types of advancement path analysis 3904: targeted paths 3923 (see Figure 40 for
26 examples), iteration-wise paths 3924 (see Figure 41 for examples), and N-part open-
27 ended paths 3925 (see Figure 42 for examples). Upon obtaining selections from a
28 seeker for one of the types of analysis 3904, or upon making a determination that the
29 seeker provided advancement experience information best suitable for only one of the
30 types of analysis 3904, and upon performing the respective analysis 3923, 3924, 3925,
31 the APT will construct an advancement path based on the seeker's advancement

1 experience information and present it based on a selected visualization style 3906. The
2 visual style may be selected by the seeker from a set of visualization template styles, or
3 selected by the APT and/or administrator.

4 **[00331]** Upon applying the visualization style to the determined advancement path
5 3906, this visualization of the advancement path is provided to the client for display
6 3908. It should be noted that, e.g., career, advancement paths may be stored and
7 shared as between users. In one embodiment, regardless of how the path is determined,
8 The seeker may then interact with the visualized path and the APT may obtain the user
9 interactions 3909. The APT may then determine if any of the user interactions provided
10 new experience information, advancement information, or modifications to the
11 constructed path such that new paths need to be generated 3910. If the interactions are
12 such that require providing more information 3910 then the seeker is allowed to again
13 provide more advancement experience information or otherwise modify factors
14 affecting the generated path 3902. Otherwise 3910, the APT will determine if the user
15 interactions 3909 require that the display is updated 3911. If the user modified or
16 provided inputs, indicia and/or otherwise operated on path objects or values that
17 require that the path visualization and/or screen is updated, the data obtained from the
18 user interactions 3909 is then used by the APT to effect updates the career path display
19 3908. Otherwise, the APT may conclude 3912 and/or wait for further interactions.

20

Path-Independent Targeted APT

21 **[00332]** FIGURE 40 is of a logic flow diagram illustrating path-independent (i.e.,
22 targeted) path construction embodiments for the APT. It should be noted that Figures
23 41, 42, 43 and 44 offer mechanisms that may supplement, alter, and/or otherwise
24 provide embodiments alternative to Figure 40. Upon obtaining seeker experience
25 advancement information 3902 and determining that a targeted independent
26 advancement path is desired 3904 of Figure 39, the APT will use advancement
27 experience information to establish a start state and a target state 4014.

28 **[00333]** In one embodiment, the seeker experience advancement information may
29 be provided by the seeker by way of a web form as shown in Figures 47, 48, 49. The web
30 form may be served by an information server, and the web form fields may serve as a

1 vessel into which the seeker may provide structure information, attach a resume, specify
 2 advancement experience information, or otherwise provide both experience information
 3 and specify the desired advancement milestone and/or goal. In one embodiment, this
 4 information is submitted to the APT and is stored as field entries in the APT database
 5 table for the seeker, e.g., in a seeker profile record. In another embodiment, this
 6 information is provided in XML message format such as the following:

7

```

8 <Advancement Experience Information ID = "experience12345">
9   <Experience Information>
10     <Job 1>
11       <Title 1> Assistant to the Management Consultant </Title 1>
12       <Start_Date> 03/14/89 </Start_Date>
13       <End_Date> 5/15/03 </End_Date>
14       <DescriptionTerms>
15         <Term1> training </Term1>
16         <Term2> process </Term2>
17         <Term3> development </Term3>
18         <Term4> costs </Term4>
19         <Term5> coffee </Term5>
20       </DescriptionTerms>
21     </Job 1>
22     <Job 2>
23       <Title 1> Assistant Management Consultant </Title 1>
24       <Start_Date> 05/16/03 </Start_Date>
25       <End_Date> 6/15/09 </End_Date>
26       <DescriptionTerms>
27         <Term1> training </Term1>
28         <Term2> process </Term2>
29         <Term3> development </Term3>
30         <Term4> costs </Term4>
31       </DescriptionTerms>
32     </Job 2>
33     <Job 3> ... </Job 3>
34   </Experience Information>
35   <Advancement Information>
36     <DescriptionTerms>
37       <Term1> Executive </Term1>
38       <Term2> Consultant </Term2>
39     </DescriptionTerms>
40   </Advancement Information>
41   <Filter Information>
42     <DescriptionTerms>
43       <Filter1> Salary > $100,000 </Filter1>
44       <Filter2> Region Zipcode [e.g., 10112] < 25 miles </Filter2>
45       <Filter3> Degree < Masters </Filter3>
46       <Filter4> Growth > 20% </Filter4>
47       <Filter5> Relocation Expenses = TRUE </Filter5>
  
```

```

1         <Filter6> Expected Next Year Occupation Demand Level > 20,000
2     jobs </Filter6>
3         <Filter7> Signing Bonus > $10,000 </Filter7>
4         <Filter8> Annual Technology Stipend > $5,000 </Filter8>
5         <Filter9> Annual Health Insurance Stipend > $25,000 </Filter9>
6         <Filter10> Regular Travel = False </Filter10>
7         <Filter11> Salary Level > [Top] 10% [in field] </Filter11>
8     </DescriptionTerms>
9 </ Advancement Experience Information ID >
10

```

[00334] Turning for a moment to Figures 47 and 48, the Figures show alternative example embodiments of how start states and target states may be selected. In another embodiment, the seeker may navigate a state structure topology such as may be seen in 4705 of Figure 47. This may be achieved by clicking on advancement topics 4709 that will zoom in to show various advancement states 4712, which the user may specify as being start state, intermediate state, and end state 4714 of Figure 33. In yet another embodiment, the seeker may enter a topic, career choice, title or other information indicative of a desired state 4724 in a field, which will be submitted as a query to the state structure; the state structure in return will return states that most closely match the supplied search term 4726, which the user in turn may select 4726 and which may be displayed, zoomed in on, and further manipulated in a topology display area 4727 of Figure 47. In yet another embodiment, the user may similarly supply terms to identify both a start state and target state 4824, 4826, 4828, which will form the basis of a path between start state and target state 4833 of Figure 48; in this embodiment, the APT similarly identify potential matching states for each of the supplied terms 4828 and constructed various paths that match the results from those terms 4837 of Figure 48. The seeker may then select from the list of paths 4837 and the path topology display area 4839 will be updated to reflect the selected path 4839 of Figure 34. So for example, the seeker may specify their current position an Assistant Administrator in a retail hardware store and that they have a goal of becoming a Regional Manager of a chain of hardware stores. In such an embodiment, the APT may use the provided seeker advancement information as a basis with which to query the state structure to identify the current advancement state, and the target advancement state. For example, for the starting state, the APT may use the most current job information, e.g., the employer name, the title, and description describing the current job, and query the state structure

1 for states that most closely match current job information; for example, a select
2 command may be performed on the state structure for stats that most closely match all
3 the supplied terms, and use the highest ranking match as the selected current state.
4 Similarly, the target job information may be used to find a target state.

5 **[00335]** Turning back to Figure 40, upon establishing a start state and a target
6 state 4014, the APT prepares to search for paths connecting the start and target states
7 4015, 2616.

8 **[00336]** It should be noted that no target state need be selected, and in such an
9 instance, the APT will use the start state to query the state structure for potential states
10 that may be of interest to a seeker with no particular target as will be discussed later in
11 Figures 41-43 regarding iteration-wise implementations. Such iteration-wise
12 implementations allow a seeker to gauge and possibly forge their own pathways after
13 being presented with the various likelihoods of those adjacent and potential
14 advancement states.

15 **[00337]** Continuing with the description of a targeted implementation, it should be
16 noted that while the APT may make use of a start and target state, specification of
17 intermediate states are also contemplated. However, it should be noted that
18 intermediate paths may be constructed by pair-wise re-processing of paths as discussed
19 in Figure 40; for example, if a seeker initially chooses a start state of Janitor and target
20 state of CEO, the APT may construct a path of Janitor state -> Manager state -> CEO.
21 However, seekers may themselves change and/or specify an intermediate state of
22 Regional Administrator. This intermediate state of Regional Administrator may be used
23 by the APT as a target state with Janitor state being the starting state; from which a first
24 path may be constructed as between the two states, e.g., Janitor state -> Facilities
25 Administrator state -> Regional Administrator state. In turn, the APT will then use the
26 target Regional Administrator state as a starting state and CEO as target state to
27 construct a second path, e.g., Regional Administrator state -> Regional Manager state ->
28 CEO state. Thereafter, the APT may join the first resulting path together with the second
29 resulting path, as the intermediate Regional Administrator state is the same for both
30 paths, and result in a new seeker directed path, e.g., Janitor state -> Facilities
31 Administrator state -> Regional Administrator state -> Regional Manager state -> CEO

1 state. A practically limitless number of pair-wise re-processing operations may be
2 employed as a seeker seeks out and selects intermediate states for a path.

3 **[00338]** In preparing to search for connecting paths as between a start state and
4 target state, the APT may use specified minimum likelihood thresholds, P_{\min} , and a
5 maximum number of path state nodes N_{\max} 4015. In one embodiment, an administrator
6 sets these values. In an alternative embodiment, a seeker may be presented with a user
7 interface where they are allowed to specify these values; such an embodiment allows the
8 seeker to tighten and/or loosen search constraints that will allow them to explore more
9 “what if” advancement (e.g., career) advancement path scenarios. The APT may then
10 establish an iteration counter, “i”, and initially set it to equal “1” 4016. Using the start
11 state, the APT may query the state structure for the next most likely states 4017. In an
12 alternative path-dependent embodiment, the APT may use the seeker’s provided
13 experience information, i.e., the entire state path, as a starting point and query the state
14 structure for next most likely states following the seeker’s last experience state (more
15 information about path-dependant traversal may be seen in Figures 42 and 44).

16 **[00339]** As the state structure maintains the likelihood of moving from any one
17 state to another state, the APT may query for the top most likely next states having
18 likelihoods greater than the specified minimum probability P_{\min} . For example, if a P_{\min} is
19 set to be 50% probability, i.e., 0.5, and the start state 4050 has the following partial list
20 of related next states: state A with $P=0.5$ 4051, state B with $P=0.7$ 4052, state C with
21 $P=0.9$, and state Z with $P=0.1$ 4054; then of next states A, B, C and Z, only states A, B,
22 and C have likelihoods above the P_{\min} threshold, and as such, only those states will be
23 provided to the APT 4017. In an alternative embodiment, instead of specifying a
24 likelihood threshold, P_{\min} instead may specify the minimum number of results for the
25 state structure to return (e.g., P_{\min} may be set to 10, such that the top 10 next states are
26 returned, regardless of likelihood/probability). The APT may then determine if any
27 and/or enough matches resulted 4018 from the query 4017. If there are not enough (or
28 any) matches that result 4018, then the APT may decrease the P_{\min} threshold by a
29 specified amount (e.g., from 0.5 to 0.25, from 10 to 5, etc.); alternatively, the APT (or a
30 seeker) may want to try again 4029 by loosening constraints 4031, or otherwise an error
31 may be generated 4030 and provided to the APT error handling component 4021.

1 **[00340]** If there are matching 4018 next states (e.g., A 4051, B 4052, C 4053)
2 proximate to the start state 4050, then the APT may pursue the following logic, in turn,
3 as to each of the matched next states (i.e., whereby each of the next states (e.g., A 4051,
4 B 4052, C 4053) will form the basis for alternative advancement paths (e.g., Path 1,
5 4091, Path 2, 4092, Path 3 4093, respectively) 4033.

6 **[00341]** Upon identifying matching next states 4017, 4018, the APT may append
7 4081 a next state (e.g., A 4051) 4022 to the start state. Upon appending a next state to
8 the start state 4022, the APT will then determine if appended next state (e.g., A 4051)
9 matches any of the target state (e.g., 4099) criteria 4023. In one embodiment this may
10 be achieved by determining if the next state has the same state_ID as the target state. In
11 an alternative embodiment, the state structure provides the state record of the target
12 state to the APT, and the APT uses terms from the target state as query terms to match
13 to the state record of the next state; when enough term commonality exists, the APT
14 may establish that the next state is equivalent to, and/or close enough to the target state
15 to be considered a match.

16 **[00342]** If the appended next state 4022 does not match the target state 4023, then
17 the APT will continue to seek out additional intermediate 4027 state path nodes (e.g., D
18 4061 and F 4071) until it reaches the target state (e.g., 4099). In so doing, the APT will
19 determine if the current state node path length “i” has exceeded the maximum specified
20 state node path length N_{\max} 4027. If not 4027, the current state node path length “i” is
21 incremented by one 4028. Thereafter the last appended state (e.g., A 4051) will become
22 the basis for which the query logic 4017 may recur (i.e., the appended state effectively
23 becomes the starting state from which proximate nodes may be found by querying the
24 state structure as has already been discussed 4017) For example, in this way next state A
25 4051 becomes appended to the start state 4050, and then the appended 4022 state A
26 4051 becomes a starting point for querying 4017, where the state structure, may in turn,
27 identify a state node proximate to the appended state, e.g., state D 4061; in this manner
28 state D 4061 becomes the next state to state A 4051. By this recurrence 4022, 4027,
29 4028, 4017, the APT grows the current path (e.g., Path 1 4091).

30 **[00343]** If the current state node path length “i” has exceeded the maximum
31 specified state path length, N_{\max} 4027, then the APT may check to see if there is another

1 next state for which a path may be determined 4036. For example, if the maximum
 2 allowable state path length is set to $N_{\max} = 2$, and the APT has iterated 4028, 4017 to
 3 reach state F 4071 along Path 1 4091, then the current state path length (i.e., totaling 3
 4 for each of states A 4051, D 4061, and F 4071) would exceed the specified N_{\max} ; in such a
 5 scenario where N_{\max} has been exceeded 4027, if the APT determines there are additional
 6 states next to the start state 4036 (e.g., B 4052, C 4053), then the APT will pursue and
 7 build, in turn, a path stretching from each of the remaining next states (e.g., Path 2 4092
 8 from next state B 4052, and Path 3 4093 from next state C 4053). If there is no next
 9 state 4036 (e.g., each of stats A 4051, B 4052, and C 4053 have been appended to the
 10 start state 4050), the APT may then move on to determine if any paths have been
 11 constructed that reached the target state 4037. If no paths reaching the target have been
 12 constructed 4037, then the APT (e.g., and/or the seeker) may wish to try again 4029 by
 13 loosening some of its constraints 4031. In one embodiment, the maximum state path
 14 length N_{\max} may be increased, or minimum likelihoods P_{\min} may be lowered 4031 and
 15 the APT may once again attempt to find an advancement path 4016. If there is no
 16 attempt to try again 4029, the APT may generate an error 4030 that may be passed to a
 17 APT error handling component 4021, which in one embodiment may report that no
 18 paths leading to a target have been found.

19 **[00344]** However, if paths have been constructed 4037, then the APT may
 20 determine the likelihoods of traversing each of the resulting paths 4024. For example, if
 21 we have a start state 4050 and a target state of 4099, the APT may have found three
 22 states next to the start state with a sufficient P_{\min} (e.g., over 0.5); e.g., next states
 23 including: state A with $P=0.5$ 4051, B with $P=0.7$ 4052, and state C with $P=0.9$ 4053.
 24 Continuing this example, if the APT continues to search for states proximate to each
 25 next state (as has already been discussed), it may result three different state paths: Path
 26 1 4091, Path 2 4092, and Path 3 4093, all arriving at the target state 4099. Each of the
 27 paths may have a probability or likelihood of being reached from the start state 4050; in
 28 one embodiment, the likelihood may be calculated as the product of the likelihood of
 29 reaching each of the states along the path. For example, the Path 1 4091 calculation
 30 would be $P_A * P_D * P_F$, (i.e., $0.5 * 0.9 * 0.9 = 0.405$). Similarly, for Path 2 4092, the

1 calculation would be $P_B * P_E$ (i.e., $0.7 * 0.5 = 0.35$). Similarly, for Path 3 4093, the
 2 calculation would be $P_C * P_A * P_D * P_F$, (i.e., $0.9 * 0.9 * 0.9 * 0.9 = 0.6561$).

3 **[00345]** As such, the APT may determine the likelihoods for each of the paths
 4 connecting to the target state(s) 4024. Upon determining the path likelihoods 4024, the
 5 APT may then select path(s) in a number of manners 4025. In the example three paths
 6 4091, 4092, 4093, the most likely path is Path 3 having a likelihood of 0.6561, the next
 7 most likely path is Path 1 having a likelihood of 0.405, and the least likely path is Path 2
 8 having a likelihood of 0.35. In one embodiment, the APT may select the path having the
 9 greatest likelihood, e.g., Path 3 4091. In another embodiment, a threshold may be
 10 specified, such that the APT will provide/present only the top paths over the threshold
 11 (e.g., if we used P_{min} as the threshold and set it to 0.5, only Path 3 would be selected with
 12 its likelihood of 0.6561 exceeding that threshold). In another embodiment, all paths are
 13 presented to the user (e.g., in ranked order) so that the seeker may explore each of the
 14 paths. Upon selecting 4025 determined paths 4024, the APT may store the paths in
 15 memory, and/or otherwise return 4086 the resulting paths 4026 for further use by the
 16 APT, e.g., provide the resulting paths for visualization to the seeker 3906, 3911 of Figure
 17 39.

18 **Path-Independent Iteration-wise APT**

19 **[00346]** FIGURE 41 is of a logic flow diagram illustrating iteration-wise path-
 20 independent path construction embodiments for the APT. Upon obtaining seeker
 21 experience advancement information 3902 and determining that a iteration-wise
 22 independent advancement path is desired 3904 of Figure 39, the APT may use
 23 experience information to establish a start state and identify suitable subsequent states
 24 for advancement consideration 4132.

25 **[00347]** As has already been discussed in Figure 40, in one embodiment, the seeker
 26 experience information may be provided by the seeker by way of a web form as shown in
 27 Figures 47, 48, 49. Unlike in the targeted embodiments discussed in Figure 40 where a
 28 seeker may supply both a start state and target state 4828 of Figure 48, a iteration-wise
 29 approach allows a seeker to identify a starting state in any number of ways as has
 30 already been discussed in Figure 26 (e.g., identifying a category 4709, and zooming into

1 a related state 4712, and making selections 4714 of Figure 47 to add a selected state to a
2 path 4841, 4846 of Figure 34; typing in a search term 4724 to find matching states from
3 a state structure 4726 of Figure 47, and selecting those matching states to act as a
4 starting state for a path; and/or the like).

5 **[00348]** In preparing to search for states proximate to a starting state, the APT may
6 obtain a starting state (e.g., from experience information, from selection/indication
7 obtained from a seeker via a user interface, and/or the like) and use a specified
8 minimum likelihood thresholds for considering proximate states P_{\min} 4132, as has
9 already been discussed above. Upon obtaining a start state and a minimum likelihood
10 4132, a seeker may also provide state filter information 4134. In one embodiment, state
11 filter information may comprise: salary requirements, geographic region and/or
12 location requirements, education requirements, relocation expense requirements,
13 minimum occupational growth rates, expected demand levels for a state, and/or the
14 like. This information may be supplied to the web interfaces discussed in Figures 47, 48,
15 49 and used as has already been discussed in Figure 26. For example, additional criteria
16 4848 may be specified and supplied into text fields 4849. In one embodiment, these
17 attributes may be stored in an attributes database table, that table may have a state_ID
18 field that makes those attributes associated with a particular state; as such the attributes
19 may be selected by a state, and may be used as criteria for filtering. Although in one
20 embodiment, when selecting a state 4850 will show additional information associated
21 with that state 4859, in an alternative embodiment, upon indicating that filtering should
22 be used 4848, a user is able to place filtering criteria into fields 4849 of Figure 48 that
23 will be made part of the query to the state structure, which may have an associated
24 attributes database, and such filtering criteria will be used to filter out unwanted states.
25 These filter criteria may be part of the XML query structure as has already been
26 described in Figure 26. Upon obtaining a start state and minimum threshold 4132 and
27 filter information 4136, a query is provided to the state structure and any associated
28 attribute database 4138. The APT then obtains states next states proximate to the
29 starting state having a minimum likelihood threshold and whose associated attribute
30 information also satisfies the requirements of the supplied filter selections 4138. In
31 another embodiment, a threshold may be based upon minimum likelihood and

1 maximum number of results. If there are no matches 4140, the seeker may adjust the
2 starting point and minimum thresholds and attempt to identify next states again 4132.
3 In another embodiment, an error may be generated indicating no matches 4142. If
4 there are matching states 4140, in one embodiment, those matching states 4140 may be
5 appended to the starting state and made a part of the advancement path 4144. Those
6 matching next states 4140 may then be displayed 4146. It should be noted that when
7 making a selection of a state 4850, and supplying any filter criteria 4859, the APT may
8 obtain matching 4140 states that may be tenuously appended as potential next states
9 4860 of Figure 48. Seekers may make such appending more permanent by indicating
10 they would like to add a state to a path they are constructing 4846, 4843 of Figure 48,
11 which may result in the updating and/or modification of the path depiction that is
12 displayed 4146. Upon updating the display 4146, the APT may allow a seeker to continue
13 on from the last selected/added state and iterate and continue to build a desired path
14 further 4132; otherwise operations may return to Figure 39 3986. It should be noted in
15 one embodiment, this path-dependant iteration-wise mechanism may be use to select
16 intermediate states in the targeted path-dependant mechanism described in Figure 40.

17

Path-Dependent Iteration-wise APT

18 **[00349]** FIGURE 42 is of a logic flow diagram illustrating iteration-wise path-
19 dependant path construction embodiments for the APT. This is an alternative path-
20 dependant embodiment of Figure 45. Upon obtaining seeker experience advancement
21 information 3902 and determining that a iteration-wise independent advancement path
22 is desired 3904 of Figure 39, the APT may use experience information to establish a
23 start state and identify suitable subsequent states for advancement consideration 4132.

24 **[00350]** As has already been discussed in Figures 40 and 41, in one embodiment,
25 the seeker experience information may be provided by the seeker by way of a web
26 interfaces as shown in Figures 47, 48, 49. Unlike in the targeted embodiments discussed
27 in Figure 40 where a seeker may supply both a start state and target state 4828 of Figure
28 48, a iteration-wise approach allows a seeker to identify a starting state in any number
29 of ways as has already been discussed in Figure 40. In an alternative embodiment, the
30 APT take into account all the seeker's experience information. While in Figure 41,

1 examples were provided where a single experience state was provided and/or otherwise
2 selected by the seeker, however, in this path-dependant embodiment, a seeker's full
3 experience information may be used as a basis of path discovery. Some seekers may have
4 no experience history or a single entry, and in such instances, this path-dependant
5 embodiment will look much like that path-independent embodiment. In one
6 embodiment, a seeker may supply this experience information into structured web
7 forms, which may be stored as structured data in a seeker profile associated with the
8 seeker (e.g., a seeker may enter their resume job experiences into a web form). In an
9 alternative embodiment, a seeker may provide their resume, which in turn may be
10 parsed into structured data, the resulting structured data serving as experience
11 information.

12 **[00351]** In preparing to search for states proximate to a path-dependent starting
13 state, the APT may use a specified minimum likelihood threshold for considering a state
14 proximate to the latest state in their experience information P_{\min} 4250. In one
15 embodiment, a seeker may supply experience information, which will serve as path-
16 dependant ("PD") criteria 4252, which as described in Figure 40 (for example a state
17 structure as may be represented, in one embodiment, by way of the XML structure),
18 may include a temporal sequence and description of advancement progression (e.g.,
19 jobs 1, jobs 2, etc.). The APT may determine a state for each of these advancement
20 progression entries and create a path describing the seeker's past state path progression
21 and use that path as a basis to search the state structure (e.g., as has been described
22 above and in greater detail in patent application serial no. ____00/000,000____ filed
23 ____MONTH_00____, 20____, entitled "APPARATUSES, METHODS AND SYSTEMS
24 FOR A CAREER STATISTICAL ENGINE," attorney docket no. 17199-033US); the last
25 progression entry (e.g., the latest job held by a seeker) may be used as a basis from
26 which the seeker will further build out their advancement (e.g., career) path. In one
27 embodiment, the state structure may return state, which may be used by the APT as
28 state advancement experience information. For example, the job entries (e.g., Job 1, Job
29 2, etc.) from the structured (e.g., XML) advancement experience information in Figure
30 40 may be supplied to the state structure, which in turn may return equivalent job

1 states. Instead of using the Figure 40 advancement experience information, a state
 2 version of that information may be used by the APT, for example:

```

3
4 <State Advancement Experience Information ID = "experience12345">
5   <Experience Information>
6     <State ID 1> 111111 </State ID 1>
7     <State ID 2> 222222 </State ID 2>
8     <State ID 3> 333333 </State ID 3>
9   </Experience Information>
10  <Advancement Information>
11    <DescriptionTerms>
12      <Term1> Executive </Term1>
13      <Term2> Consultant </Term2>
14    </DescriptionTerms>
15  </Advancement Information>
16  <Filter Information>
17    <DescriptionTerms>
18      <Filter1> Salary > $100,000 </Filter1>
19      <Filter2> Region Zipcode [e.g., 10112] < 25 miles </Filter2>
20      <Filter3> Degree < Masters </Filter3>
21      <Filter4> Growth > 20% </Filter4>
22      <Filter5> Relocation Expenses = TRUE </Filter5>
23      <Filter6> Expected Next Year Occupation Demand Level > 20,000
24  jobs </Filter6>
25      <Filter7> Signing Bonus > $10,000 </Filter7>
26      <Filter8> Annual Technology Stipend > $5,000 </Filter8>
27      <Filter9> Annual Health Insurance Stipend > $25,000 </Filter9>
28      <Filter10> Regular Travel = False </Filter10>
29      <Filter11> Salary Level > [Top] 10% [in field] </Filter11>
30    </DescriptionTerms>
31  </State Advancement Experience Information ID >
32

```

33 **[00352]** In the above state version of advancement experience, the state structure
 34 provided state equivalents of the job entries in the Figure 40 experience information,
 35 and this state experience information may be supplied to the state structure as a path
 36 comprising State ID 1, State ID 2, and State ID 3 representing Job 1, Job 2 and Job 3
 37 from the XML description in Figure 40. Results from querying the state structure with
 38 an existing state progression path will provide the APT and use the latest advancement
 39 progression entry as a starting point; e.g., from the above state advancement experience
 40 information, State ID 3 would be the state from which a further advancement path
 41 would be build by the APT, i.e., State ID 3 would be the path-dependant start state to
 42 which additional path advancement states would be appended 4252.

1 **[00353]** Upon populating the APT with path-dependant criteria (e.g., with
2 experience advancement experience information, state advancement experience
3 information, and/or the like) 4252 and obtaining a minimum likelihood threshold 4250,
4 a seeker may also provide state filter information 4254, which may be used to modify
5 the path-dependent criteria 4254 (as has already been discussed in Figure 41). In one
6 embodiment, state filter information may comprise: salary requirements, geographic
7 region and/or location requirements, education requirements, relocation expense
8 requirements, minimum occupational growth rates, expected demand levels for a state,
9 and/or the like. These filter criteria may be part of the XML query structure as has
10 already been described in Figure 40.

11 **[00354]** Upon obtaining a minimum threshold 4250, populating the APT with
12 path-dependant criteria 4252 and filter information 4254, a query may be provided to
13 the state structure and any associated attribute database 4256. For example, the state
14 advancement experience information (or subset thereof) may be provided to the state
15 structure as a query. Upon obtaining query results from the state structure, the APT may
16 determine which of the returned states to use that satisfy the filter selections 4254 and
17 minimum thresholds specified and retrieve the state records (and any associated
18 attributes) related to the determined state(s) 4256. The APT may then determine if any
19 state results match 4258; if not, the seeker may adjust the parameters of the search by
20 starting over 4250, or alternatively an error is generated 4259.

21 **[00355]** If there are matching states 4258, in one embodiment, those matching
22 states 4258 may be appended to the path-dependant starting state and made a part of
23 the advancement path 4260. Those matching next states 4258 may then be displayed
24 4261. It should be noted that when making a selection of a state 4850, and supplying
25 any filter criteria 4859, the APT may obtain matching 4258 states that may be visually
26 appended as potential next states 4860 of Figure 48, providing highlighting to show
27 potential path connections. Seekers may make such appending appear more permanent
28 4263 by indicating they would like to add a state to a path they are constructing 4846,
29 4843 of Figure 48, which may result in the updating and/or modification of the
30 associations between states and generation of a path depiction that is displayed 4261.
31 Upon updating the display 4261, the APT may allow a seeker to continue 4262 on from

1 the last selected/added state 4260. If no continuation is desired or needed, operations
2 may return to Figure 39 4286. Otherwise, if continuation is desired 4262, the APT may
3 allow a user to update their previous experience information 4263, 4264. If a user
4 wishes to append or add states representing past experience (e.g., if the seeker did not
5 initially supply all of their experience information as path dependent criteria 4252) 4264
6 and specifies such, the APT will allow them to append such experience states as path-
7 dependence criteria 4252. In one non-limiting example, a seeker may build up 4846,
8 4843 of Figure 48 states representing their experiences in this manner. Alternatively,
9 the seeker may not wish to append experience states 4263, yet the APT may determine if
10 any changes to any of the path-dependence criteria was affected by the seeker (e.g., a
11 seeker may have changed an originally supplied experience state to another, the other
12 perhaps showing a promotion in their most current work employment) 4265. If no
13 changes to path-dependant criteria were determined 4265, the APT may continue to
14 iterate and build a path based on the last appended state 4260, 4256. However, if there
15 has been a change in the path dependant criteria 4265, this changed criteria will form
16 the basis of iterated path dependent criteria 4252.

17

Path-Independent N-Part Open-Ended APT

18 **[00356]** FIGURE 43 is of a logic flow diagram illustrating N-part path-independent
19 path construction embodiments for the APT. This is an alternative path-independent
20 open-ended embodiment of Figure 41. Upon obtaining seeker experience advancement
21 information 2502 and determining that a iteration-wise independent advancement path
22 is desired 2504 of Figure 25, the APT may use experience information to establish a
23 start state and identify suitable subsequent states for advancement consideration 4365.

24 **[00357]** As has already been discussed in Figures 40, 41 and 42, in one
25 embodiment, the seeker experience information may be provided by the seeker by way
26 of a web form as shown in Figures 47, 48, 49. Unlike in the iteration-wise embodiments
27 discussed in Figure 27, an open-ended approach allows a seeker to identify a starting
28 state in any number of ways as has already been discussed in Figure 40; it also allows
29 the seeker to specify desired path length N. As such, the APT, an administrator, another

1 system, or the seeker may specify the desired number of states to comprise an
2 advancement path, that length being “N” 4365.

3 **[00358]** In preparing to search for states proximate to a starting state, the APT may
4 obtain a starting state (e.g., from experience information, from selection/indication
5 obtained from a seeker via a user interface, and/or the like) and use the specified path
6 length limit N, as has already been discussed above. Upon obtaining a start state and
7 limit 4365, a seeker may also provide state filter information 4367. In one embodiment,
8 state filter information may comprise: salary requirements, geographic region and/or
9 location requirements, education requirements, relocation expense requirements,
10 minimum occupational growth rates, expected demand levels for a state, and/or the
11 like. This information may be supplied to the web interface discussed in Figures 47, 48,
12 49 and used as has already been discussed in Figure 40. For example, additional criteria
13 4848 may be specified and supplied into text fields 4849. Although in one embodiment,
14 when selecting a state 4850 will show additional information associated with that state
15 4859, in an alternative embodiment, upon indicating that filtering should be used 4848,
16 a user is able to place filtering criteria into fields 4849 of Figure 48 that will be made
17 part of the query to the state structure, which may have an associated attributes
18 database, and such filtering criteria will be used to filter out unwanted states. In
19 another embodiment, browsing an associated hierarchy through nested pop-up menus
20 5330 of Figure 53 or traversal and selection of nodes in a topography 4722, 4727 of
21 Figure 47 provide another mechanism for identifying states and building paths. These
22 filter criteria may be part of the XML query structure as has already been described in
23 Figure 40. Upon obtaining a path limit and filter information 4368, the APT may set the
24 current path length “i” to equal “1” 4369. The APT may then provide a query to the state
25 structure and any associated attribute database 4370, including filter criteria, as has
26 already been discussed. The APT then obtains states next states proximate to the
27 starting state having a minimum likelihood threshold and whose associated attribute
28 information also satisfies the requirements of the supplied filter selections, and may
29 append those next states to the current advancement path 4371. In an alternative
30 embodiment, a seeker may traverse by categorical hierarchy selections as shown in 5335
31 of Figure 53, whereby a seeker can iteratively make state selections by identifying states

1 through hierarchical selections. Thereafter, the APT may increment the path length
2 counter “i” by one 4371 to track the growth of the path length resulting from the
3 appending 4370. If the maximum path length N has not been reached 4373, the APT
4 may iterate and similarly conduct queries on the appended next states, to extend the
5 path 4370. If the path length has been reached 4373, operations may return to Figure
6 39 4386.

7

Path-Independent N-Part Open-Ended APT

8 **[00359]** FIGURE 44 is of a logic flow diagram illustrating N-part path-dependent
9 path construction embodiments for the APT. This is an alternative path-dependent
10 open-ended embodiment of Figure 42. Upon obtaining seeker experience advancement
11 information 3902 and determining that an iteration-wise independent advancement
12 path is desired 3904 of Figure 39, the APT may use experience information to establish
13 a start state and identify suitable subsequent states for advancement consideration
14 4474.

15 **[00360]** As has already been discussed in Figures 40, 41, 42 and 43, in one
16 embodiment, the seeker experience information may be provided by the seeker by way
17 of a web form as shown in Figures 47, 48, 49. Unlike in the iteration-wise embodiments
18 discussed in Figure 42, an open-ended approach allows a seeker to identify a starting
19 state in any number of ways as has already been discussed in Figure 40; it also allows
20 the seeker to specify desired path length N. As such, the APT, an administrator, another
21 system, or the seeker may specify the desired number of states to comprise an
22 advancement path, that length being “N” 4474. While in Figure 43, examples were
23 provided where a single experience state was provided and/or otherwise selected by the
24 seeker, however, in this path-dependant embodiment, a seeker’s full experience
25 information may used as a basis of path discovery. Some seekers may have no
26 experience history or a single entry, and in such instances, this path-dependant
27 embodiment will look much like that path-independent embodiment. In one
28 embodiment, a seeker may supply this experience information into web structured web
29 forms, which may be stored as structured data in a seeker profile associated with the
30 seeker (e.g., a seeker may enter their resume job experiences into a web form). In an

1 alternative embodiment, a seeker may provide their resume, which in turn may be
2 parsed into structured data, the resulting structured data serving as experience
3 information.

4 **[00361]** In preparing to search for states proximate to a path-dependent starting
5 state, the APT may discern a path-dependent starting state as has already been
6 discussed in Figure 28, and use the specified path length limit N 4474, as has already
7 been discussed above. Upon obtaining a path limit N, the APT may set the current path
8 length “i” to equal “1” 4475. A seeker may then provide state filter information 4476,
9 which may be used to obtain resulting states matching the filter criteria 4477. In one
10 embodiment, state filter information may comprise: salary requirements, geographic
11 region and/or location requirements, education requirements, relocation expense
12 requirements, minimum occupational growth rates, expected demand levels for a state,
13 and/or the like. This information may be supplied to the web forms discussed in Figures
14 47, 48, 49 and used as has already been discussed in Figure 40. For example, additional
15 criteria 4848 may be specified and supplied into text fields 4849. Although in one
16 embodiment, when selecting a state 4850 will show additional information associated
17 with that state 4859, in an alternative embodiment, upon indicating that filtering should
18 be used 4848, a user is able to place filtering criteria into fields 4849 of Figure 48 that
19 will be made part of the query to the state structure, which may have an associated
20 attributes database, and such filtering criteria will be used to filter out unwanted states.
21 These filter criteria may be part of the XML query structure as has already been
22 described in Figure 40. Upon obtaining a path limit 4474 and filter information 4476
23 and obtaining the filtered states 4477, a seeker may supply experience information,
24 which will serve as path-dependant criteria 4478 as has already been described in 4252
25 of Figure 42, and in Figure 40. The APT will then return states matching the
26 aforementioned criteria and may then create associations between states that appear to
27 append those matching next states the path-dependant starting state, which are thereby
28 made a part of the associated states representing the advancement path 4480.

29 **[00362]** Seekers may make such appending 4480 more permanent by indicting
30 4481 they would like to add a state to a path they are constructing 4846, 4843 of Figure
31 48. As such, the APT may allow a user to update their previous experience information

1 4481, 4482. If a user wishes to append or add states representing past experience (e.g.,
2 if the seeker did not initially supply all of their experience information as path
3 dependent 4478) 4482 and specifies such, the APT will allow them to append such
4 experience states as path-dependence criteria 4482. In one non-limiting example, a
5 seeker may build up 4846, 4843 of Figure 48 states representing their experiences in
6 this manner. Alternatively, the seeker may not wish to append experience states 4481,
7 and then the APT may increment the path length by one 4483 to indicate that the
8 current path has grown by one. Upon incrementing the current state, it may result in the
9 updating and/or modification of the path depiction that is displayed 4485. Upon
10 updating the display 4485, the APT may determine if the maximum path length N is less
11 than the current path length i; if the current length of "i" is longer, then operations may
12 return 4486 to Figure 39 4286. Otherwise 4484, the APT may continue to grow to set
13 lengths N by iterating 4479.

14

Path Gap Analysis

15 **[00363]** FIGURE 45 is of a logic flow diagram illustrating gap analysis
16 embodiments for the APT. In one embodiment, a seeker may access the APT (e.g., either
17 anonymously, be logged into the system, and/or the like) 4585. In so doing, the seeker
18 may provide the APT with a start state 4586 and target state 4587, as has already been
19 discussed in Figure 44. The APT may also populate additional states (e.g., B, C...N) in
20 an embodiment that allows for multi-segment gap analysis. As such, the APT will
21 determine if it needs to analyze across multiple pat states 4508, and if so, it will add
22 those states for analysis and subsequent iteration 4509, otherwise 4508, the APT will
23 continue by initiating querying 4588. The APT may use the start and target states as a
24 basis to query the target state structure to discern states proximate to the target state
25 4588. In an alternative embodiment, the APT may supply an intermediate target state;
26 this may be achieved by first specifying a start state and an intermediate target state and
27 generating a path therebetween as already discussed; thereafter another path is
28 generated as between the intermediate target state being supplied as a starting state and
29 specifying a final target state, and once again determining a path therebetween; the two
30 paths being connected. If the APT does not obtain any matches 4589, the seeker will be

1 afforded an opportunity to restart 4586, and/or alternatively an error message may be
2 generated 4590. If the APT does identify matches 4589, then the APT may query the
3 feature attributes and transition values as stored in an attributes table occurring in time
4 between the start and next state (e.g., as between A and B, then B and C, and N-1 and N;
5 N being the target state) 4590; when there are only two states, there is just a start state
6 and target state. For example, the APT may query an attributes database with attributes
7 associated with states from the state structure (e.g., as in the advancement taxonomy),
8 wherein the attributes maintain information differences in salary as between different
9 individuals having the same employment state. As such, the APT examines the
10 attributes database record entries associated with each state and determine the common
11 gap attributes; in another embodiment, an administrator specifies which attribute types
12 have the most common gap attributes. Thereafter, the APT may calculate the feature
13 gaps 4591 as discussed in greater detail in Figure 45. Similarly to the previous query for
14 features 4590, the APT may query for state change indicators as between states (e.g., as
15 between AB_i, BC_i ...N-1N_i) 4592. Similarly to the feature gap calculation, the APT may
16 calculate the state change indicators 4599 as discussed in greater detail in Figure 46. In
17 one embodiment, the APT may determine if these calculated gaps are statistically
18 significant 4593 (e.g., by determining comparing it the value exceeds a common
19 standard deviation for the gap type). If these are statistically significant, then the APT
20 may return these gap feature attributes and transition indicators (e.g., as discussed in
21 greater detail in Figure 46), which may be provided to the seekers (e.g., to see how their
22 salary compares to others living in the same region and having the same vocational
23 post) 4596; in one embodiment, this gap analysis may be employed by APT for
24 benchmarking. If there is no statistical significance 4593, then an error message may be
25 returned, noting that no significant gap attributes exist 4594 and the APT will allow a
26 user to try 4595 and recurse 4586 if the seeker so wishes 4595, otherwise gap analysis
27 operations cease. In one embodiment, such gap analysis may be instantiated when user
28 selects additional options 4947, 4966 of Figure 49 after having selected start and target
29 states.

30 **[00364]** FIGURE 46 is of a state path topography transition diagram illustrating
31 gap analysis embodiments for the APT. In one embodiment, a gap analysis may be

1 determined as between start state A 4610 and target state C 4601, having one
 2 intermediate state B 4605. In such an example, each state has a gap measurable
 3 attributes represented by F 4641, 4651, 4661, which may be float, integer, textual,
 4 and/or functions that represent stored attributes in an attributes database table. The
 5 features may be qualitative and/or quantitative. Features may include skills topics,
 6 terms, attributes (e.g., salary, vacation, etc.) and/or the like. Gap transition indicators
 7 may also represent part of a transition from one state to another. Transition indicators
 8 may include years of experience, obtainment of education (e.g., obtaining a new degree),
 9 attributes, and/or the like. In one example, state A (e.g., a state representing an
 10 assistant graphics designer position) have a set 4620 of attributes F 4661 having a salary
 11 of \$50,000 4663 and a skill of using Microsoft Paint 4664. State B (e.g., a state
 12 representing a graphics designer position) may have a set of attributes F 4651, having a
 13 salary of \$60,000 4653, a skill set of Microsoft Paint, Photoshop, and Team
 14 Management skills 4654, and provide 20 days of vacation time 4656. Also, in this
 15 example, state C (e.g., a state representing a managing graphics designer position) may
 16 have a set of attributes F 4641 having a salary of \$75,000 4643, and skill requirements
 17 of Team Management, Administration 2546.

18 **[00365]** A state may have a certain set of attributes associated with it that is
 19 exclusive to that state. The difference between state A and B, in one embodiment, may
 20 be represented as the features of B subtract out the same duplicative features of A. For
 21 example, there is a \$10,000 difference in salary as between state A 4610, 4663 and state
 22 B 4605, 4653. There also are indicators driving people from state A to B, e.g., like years
 23 of experience, the obtainment of a specific degree, and or the like. In one example it
 24 make take 5 years of experience for a transition to occur on average, e.g., $AB_i = 5$ 4607,
 25 $BC_i = 5$ 4603; these are indicators of change between states. As such, in one
 26 embodiment gap indicators as between states A and B may be calculated as follows:

27 **[00366]** $G_{i(A \rightarrow B)} = B_F - A_F + AB_i$ 4699, or

28 **[00367]** Salary: \$10,000; Skill: Photoshop, Team Management; 5 years transition,

29 **[00368]** As such, gap indicators as between states B 4605 and C 4601 may be
 30 calculated as follows:

1 **[00369]** $G_{i(B \rightarrow C)} = C_F - B_F + BC_i$ 4698, or

2 **[00370]** Salary: \$15,000; Skill: Administration, -Paint, -Photoshop; 5 years
3 transition and a Masters Degree.

4 **[00371]** The gap analysis may work as an additive between any two states. So the
5 state change drivers takes drivers of change between two states and identifies them as
6 subtractive features as well as indicators. As a consequence, gap indicators as between
7 state A 4610 and C 4601 may be calculated as follows:

8 **[00372]** $G_{i(A \rightarrow C)} = C_F - B_F - A_F + AB_i + BC_i$ 4697, or

9 **[00373]** Salary: \$25,000; Skill: Team Management, Administration, -Paint; 10
10 years transition and a Masters Degree.

11

Interaction Displays

12 **[00374]** FIGURES 47-49 are of a screen shot diagram illustrating embodiments for
13 the APT. In Figure 47, the entry screen to the APT is shown 4701; however it should be
14 noted that the APT may be used without having a user profile or account. In one
15 embodiment, a seeker may initially interact with a state topology overview 4705, which
16 may also have panels for accepting inputs from the seeker to search out states, and an
17 area to show information and results from inputs provided to the APT 4707. In one
18 embodiment, a seeker may move their cursor to an advancement (e.g., career) category
19 displayed in the state topology overview and make a selection 4709. Upon selecting the
20 category and/or state, the display area 4709 will focus in on the item selected 4712 and
21 may provide the seeker with options (e.g., start/add to my path, get details about this
22 state, find a job based on this state) 4714. It should also be noted that state paths may be
23 represented in numerous ways; e.g., while in some figures the state paths are depicted as
24 interconnected nodes on a graph topography 4855, 4844 of Figure 48, in another
25 embodiment the path may be represented as a series of numbered boxes 4847 of Figure
26 48 having information relative to each state displayed within. A seeker may select to
27 view the state path 4845 of Figure 48 by providing indications that they wish to visualize
28 the state path differently. In one embodiment, a template architecture is contemplated
29 where numerous visual depictions are available and use the underlying state path and

1 state structure topology as data constructs onto which the templates may be mapped.
2 Further, the APT may provide a key showing what the various weight lines 4711 signify
3 as connectors between (e.g., circular) nodes. For example, if a seeker selected an option
4 to “get details” 4716, the APT may display attribute detail information about the selected
5 state in a information display panel 4718, which may include the number of people in
6 this job (both current and projected), expected rate of growth for the state, and other
7 associated attribute information. The seeker may navigate about the topology by
8 selecting a scale slider 4720 which will allow the seeker to zoom out and see more of the
9 topology 4722 (or conversely, to zoom in and see more detail; panning and other
10 arrangement options may also be employed).

11 **[00375]** In another embodiment, a seeker may enter a search term they believe
12 relates to a (e.g., career) state they have an interest in 4724, which will result in the APT
13 showing its top matches 4726 in the information panel as well as highlighting relevant
14 identified experience states in the topology itself 4727. It should be noted that in one
15 embodiment, the topography will adjust its overall view (e.g., zoom level) to show the
16 path results, and when making selections of states, the topography will traverse and
17 provide a fly-by depiction of the topography on to to selected states.

18 **[00376]** In Figure 48, the APT allows a seeker to provide both start and target
19 query terms 3424 (as well as additional options 4848 that would allow the seeker to
20 provide additional attribute filter criteria). In one embodiment, the additional filter
21 criteria may be entered in the APT information panel 4849 and be used as part of the
22 basis of a query to find matches in a state structure. As a seeker types 4826, suggested
23 terms and topics are displayed 4826. If a state is selected 4850, other related states are
24 pointed to and highlighted 4860. In an alternative embodiment, upon providing and
25 determining start and target states 4828, 4830, 4832, the APT may determining a path
26 with intermediary states (e.g., 4831) as between the start and target states 4833; and
27 provide the seeker with the ability to choose as between multiple paths that have
28 differing numbers of intermediary states and likelihoods of attainment by the seeker
29 4835. For example, in one embodiment a seeker may make selections 4837 as among
30 the various available paths between the start and target states, and the topology display
31 area will be updated to reflect the different paths 4839. It should be noted that the APT

1 allows a seeker to either build up or modify a path. In one embodiment this may be
 2 achieved by selecting adjacent states 4846 to states in a path 4855 and making
 3 selections to add the selected state to the path 4843. In one embodiment, a path list
 4 4844 may show the current set of states that comprises the current path such that a
 5 seeker may be apprised of the current path even if it is not fully visible in the topography
 6 display area.

7 **[00377]** In Figure 49, the APT allows a seeker to vary visualizations in numerous
 8 ways. For example, a user may elect to vary the visual depiction of the display topology
 9 by engaging an option widget 4943, which may in turn show a dialogue box that allows
 10 the user to turn on/off the ability to show, e.g., common next states, show most common
 11 paths, show tips, show trace, and/or the like 4945. For example, selecting the show tips
 12 option will highlight potential and/or likely next states for consideration for a seeker. In
 13 another embodiment, if the seeker selects show trace, a breadcrumb trail of their path is
 14 highlighted. Another option is “find jobs in path,” which when selected will allow a
 15 seeker to apply for one or more jobs that are identified along a constructed path. In
 16 another view upon selecting a state, the user may be presented with options to find a job
 17 and inform the user that the state is a common path state 4947, 4966. Another
 18 embodiment shows that upon selecting a “find a job” option 4947, the user may be
 19 presented with job listings 4949 and sponsored ads 4951. In one embodiment,
 20 occupational profile tags may be used with sponsored ads. In one embodiment, profile
 21 ad tags may be called as follows:

```

22 http://ads.monster.com/html.ng/site=mons&affiliate=mons&app=op&size=728x90&pp=1
23 &opid=***&path=(DynamicPathValue)&dcpc=#####&ge=#&dcel=#&moc=#####&dccl=##&mi
24 l=#&state=##&tile=
25 http://ads.monster.com/html.ng/site=mons&affiliate=mons&app=op&size=300x250&pp=
26 1&opid=***&path=(DynamicPathValue)&dcpc=#####&ge=#&dcel=#&moc=#####&dccl=##&m
27 il=#&state=##&tile=
28 http://ads.monster.com/html.ng/site=mons&affiliate=mons&app=op&size=(TBD)&pp=1&
29 opid=***&path=(DynamicPathValue)&dcpc=#####&ge=#&dcel=#&moc=#####&dccl=##&mil
30 =#&state=##&tile=
31 #####: These values are set by the user's cookie.
32

```

33 **[00378]** In another example embodiment, in Figure 50, job listings may be
 34 presented as part of a job carousel, as illustrated, where a user may view job listings
 35 5033 in a “lazy susan” format 5093; the user may spin job widgets 5033 to the left 5091

1 or right 5092 by click-dragging or selecting “spin” arrows 5091, 5092 and rotating jobs
2 out of view 5044, 5055. For example, if a user clicks on the left spin arrow 5091, job
3 listing 0 5244 will spin into view and job listing 4 5066 will spin out of view. It should
4 also be noted that the job carousel may also be used for job ads outside the context of
5 the APT; in one embodiment, a user may click on a job listing 5077 and an apply for job
6 menu/button/widget may appear 5093, which would allow a user to move on to apply
7 for the job. In an alternative embodiment, the APT may prompt or otherwise request
8 user identifying information 5094 (e.g., a unique identifier, a user name and password,
9 a cookie having same, and/or the like), which may be used to obtain a seeker’s
10 experience information and begin a job application process. Moving back to Figure 49, it
11 should be noted that a seeker may engage and save a desired path to their profile 4965.
12 Also, it should be noted that a user may select a tab pane that provides information to
13 help a seeker advance/continue their education, including educational listings and ad
14 space 4953. The APT also provides the opportunity for the seeker to select path
15 segments 4955, which in turn will provide the seeker with opportunity to select a path,
16 overwrite the path, modify the path, and/or the like 4957, 4959.

17

Interaction Interface Component

18 **[00379]** FIGURE 51 is a logic flow diagram illustrating embodiments for invoking
19 and displaying an APT. In one embodiment, the APT may be manifested in a web
20 environment as discussed throughout in Figures 47-49. In one such embodiment, the
21 APT may be supplied, in one example embodiment, as Flex-based Shockwave Flash
22 (SWFs) that interact with the CSM via a web services layer and with the parent through
23 Javascript-based callbacks, which may be delivered via a web page served by an
24 information server for use by seeker clients. Once the web page is loaded in a web client,
25 the interface may be instantiated and connect with the APT, its database, its component
26 collection, and in one embodiment, through the information server. Upon instantiation
27 of the web page, presentation and operation of the interface may start by querying the
28 seeker’s web client to obtain display environment and browser capabilities information:
29 this may include the type of browser, cookies (with account information, if any), plug-in
30 capabilities (if any), Javascript support, Java support, version numbers thereof, screen

1 resolution and dimensions, window sizes, and/or the like. In one embodiment, this may
2 be done with an HTTPBrowserCapabilities Request.Browser inquiry via a Microsoft .Net
3 object call. The query may occur, and the seeker may access the APT by logging into an
4 account (or by determining that a cookie contains identifying information allowing for
5 login) or anonymously. Thereafter, the seeker's client may provide and the APT may
6 obtain a user's system information, and if logged in and otherwise unsaved, this
7 information may be stored in a profile database table 5103. The APT may then provide
8 an interface appropriate to the seeker's client. In one embodiment, templates such as
9 cascading style sheets, HTML templates, and/or the like may be supplied providing the
10 seeker with a preliminary career path interaction interface 5105. In another
11 embodiment, Flash-based content rendering may be used. In one embodiment, a seeker
12 may select an initial display type 5107; for example, a seeker may select a topography
13 based interface 4844 of Figure 48, linear information view 4847, a straight-line list view
14 4844, a nodal path view 4833, and/or other views 5107. Upon providing such indication,
15 the appropriate and selected view may be provided to and loaded by the seeker's web
16 client and set 5109; in loading the template the appropriate interaction interface widgets
17 are loaded.

18 **[00380]** Upon loading the template interface view 5109, the APT may then begin
19 generating a representation of a given path for display in accordance with a given APT
20 interaction interface template. For each node representing a state to be rendered to
21 display 5111, the APT may query a APT database for seeker advancement states 5113. In
22 one embodiment, this may be a seeker's advancement experience information. In
23 another embodiment, it may be a clean state with no state topography, where a seeker
24 may begin searches for job states, as has already been discussed earlier in Figures 47-49.
25 Upon obtaining states from the APT, the state may be loaded into the interface element
26 representing the state 5114. For example, in a topography map, a user's start state may
27 be loaded as starting node in that topography. In one embodiment, a user may filter
28 results 5115; the filters may be defined by the APT or the seeker. In one embodiment, a
29 user may, for example, specify likelihood thresholds, salary levels, and/or the like. So for
30 example, if a user starts with a blank topography and performs a search for a starting
31 state, the user may specify filter attributes, which may be supplied as SQL query selects,

1 and which will act to narrow the returned states. The APT may continue to iterate if
2 there are additional interface widgets that need to be put into effect 5116, 5111,
3 otherwise, the APT may then move on to determine the type of analysis to be used for
4 path determination 5117. In one embodiment, the APT may allow a seeker to examine
5 paths independent of their own advancement experiences in a path-independent
6 approach, in a path dependant approach, perform gap analysis, examine seeker's status
7 at a given state relative to the aggregate (e.g., comparing the aggregate salary for a state
8 to the seeker's salary at a given state), and/or the like 5117. Upon having obtained an
9 initial set of seeker states 5111, and selecting an analysis type 5117, the APT may perform
10 a next-state and/or path determination analysis by using selected states as starting
11 and/or target states and using that to query the state structure 5119. Upon obtaining
12 analysis results 5119, the APT may provide user interface elements along with widgets
13 representing states for display on the seeker's client 5112. Upon displaying the pathing
14 interface 5121, the APT may monitor for seeker interaction with the pathing interface
15 and/or for further information 5123, as has already been discussed with regard to
16 Figures 47-49. If there is no interaction 5125, then the APT may continue to cycle
17 looking for additional input 5123. If there is seeker interaction 5125, then the APT may
18 check if a new display set type has been requested 5126; for example, if a user selects to
19 view a path linearly 4845 instead of as a topography 4843 of Figure 48. If a seeker elects
20 to change the display set type 5126, then the APT will load in a corresponding template
21 through which reimaging of the display may continue. If there is no indication of
22 changing the display type 5126, then the APT may determine if the interaction interface
23 is being dismissed and/or terminated; if so, termination will ensue and execution will
24 come to an end. Otherwise, the APT will go on to determine if what changes need to take
25 place 5129 as a result of user interaction 5125. For example, if a widget node is right-
26 clicked upon, or other selection indicia is provided where a selection indicator (e.g., a
27 cursor) intersects with a widget (e.g., a node representing an experience state), then the
28 APT may determine if it needs to query its CSE component 5131. For example, if a user
29 decides to change an experience path by adding a state to the path, changing one,
30 and/or the like 4841, 4843 of Figure 48, then the APT would need to obtain updated
31 state and/or path information by querying the CSE component for updated data 5133. If
32 no query is required 5131 or upon obtaining the results form the query, the APT will

1 determine if re-drawing the display is required 5135. If no re-draw is required, the APT
2 may continue to monitor for user input. If updates are required, e.g., to account for
3 updated state information obtained from the state structure 5131, then the interface may
4 be re-drawn to account for the update 5137, and then the APT may continue to monitor
5 for interactions 5123.

6 **[00381]** FIGURE 52 is a logic flow diagram illustrating embodiments for tracking
7 seeker interactions with a APT. In one embodiment, upon initiating interactions with
8 the APT, e.g., as in Figure 52, the APT may track seeker interactions by associating
9 feedback widgets with elements of advancement experience information 5252.
10 Momentarily moving to FIGURE 53, it is of a block diagram illustrating feed back
11 widgets for interactions with a APT. As has already been discussed, in one embodiment
12 a seeker may provide a title, keywords, and/or the like to the APT 5301 and look for a
13 match 5303. Such a search will result in search results that may be displayed and/or
14 highlighted by the APT in a number of ways, including a list 5305, 4724, 4726, 4727 of
15 Figure 47. In one embodiment, seekers may provide feedback in a number of ways. In
16 one embodiment, the seeker may inform the APT that search results are not appropriate
17 matches 5310, in which case the seeker may specify a job state manually 5330. In one
18 embodiment this may be achieved by allowing a seeker to navigate a hierarchical topic
19 structure wherein various tiers represent top level topics and related nested topics. In
20 one such embodiment, a seeker may make such selections by using pop-up menus that
21 are populated from the structure model's topic tables. In one embodiment, the XML
22 topic model structure may be used. Once selecting an appropriate advancement, e.g., job
23 identifier, 5330 or selecting matches 5305 and asking to confirm either setting 5315,
24 5335, a feedback widget may be presented 5320 in the information inspection area of a
25 APT interface, e.g., 4707, 4718 of Figure 47. In one embodiment, a slider widget 5320
26 may be employed, wherein the seeker may move the slider to represent a range of a good
27 match 1.0 to a poor match 0.0 by position of a slider 5320; that value may then be
28 selected 5325 and stored in the APT. The ratings may be for all kinds of attributes, for
29 example, instead of a confidence rating for the search results, a user may opt to "rate
30 this job" 4843 of Figure 48 by right-clicking on an experienced state and be presented
31 with a rating widget 5340 where they may confirm how bad or good an experience was

1 5340, 5345. Similarly, a user may decide to provide comments about a job or experience
2 through a text box 5350, the contents of which may be saved to the APT database. In
3 addition, users may rate 5360 the comments and submit those 5355 to the APT. Moving
4 back to Figure 52, various other feed back attributes may be employed and/or stored,
5 such as: text boxes to allow seekers to provide comments regarding a selected state, job
6 satisfaction ratings for experienced states (e.g., using sliders), experience requirements
7 (e.g., using toggles and/or text boxes to obtain additional requirements), and/or the like
8 5252.

9 **[00382]** Upon associating feedback with topic and/or job related information 5252,
10 the APT may then track the users view and interaction with any given job profile 5254.
11 In one embodiment, tracking may take place by doing the following for each viewing of a
12 job/advancement profile by a seeker 5256. For each such interaction by a seeker with a
13 job profile 5256, the APT may load in an appropriate feedback widget when a job profile
14 is loaded for the user 5258. For example, when a job profile is loaded to represent a state
15 in the path topology, various feedback widgets may be loaded; for example, a database
16 table may contain various attributes that are associated with a given state and/or job
17 and also are associated with various user interface templates and/or widgets, which may
18 be loaded by the APT. Once the feedback widgets are loaded for an associated job profile
19 5258, the APT may then monitor for interaction with the feedback widgets 5260 (for
20 example, as already discussed in Figure 51). If no interaction is detected, the APT may
21 continue to monitor 5260 until the interface is terminated. If the user does interact with
22 the feedback widget(s) 5262, the values obtained from that interaction are stored in the
23 APT database record associated with job profile 5264. The APT then determines if the
24 user supplying the feedback has authority to do so 5266; for example, if the user is not
25 currently employed for the selected type of job for which the feedback is proffered, and
26 it was not previously an experience of the user, then such feedback may be deprecated
27 (e.g., given low weight, may not be stored, and/or the like). The APT may then
28 determine the weight to be given based on the user authority; for example, feedback
29 from users whose most current employment is the same state for which they are offering
30 feedback will have higher weight than for feedback from users who had such experience
31 further back in their career; which in turn may have higher weight than feedback from a

1 user who has no such experience and/or less related equivalent job state experience. In
2 one embodiment, users with current experience in the equivalent state receive a weight
3 of 1.0, while users having experience in the past receive a weight of 0.5, while users
4 having no experience receive a weight of 0.1. Optionally, the APT may then collect
5 behavioral (e.g., usage frequency), demographic, psychographic, and/or the like
6 information and store it as associated attribute information 5270. For example, a user's
7 profile may include their geographic region, and as such, feedback from users in one
8 region may be analyzed distinctly from users in differing regions. The APT may then
9 store the feedback as records entered as attributes in a APT database, which is
10 associated with job information, state information, and/or the like and the APT may
11 continue to cycle for any selected job profiles 5272, 5256.

12 **[00383]** In one embodiment, as the APT continues to track feedback information
13 relating to job profiles 5254, it may periodically query its database for the feedback for
14 purposes of analysis 5274. In one embodiment, a cron job may be executed at specified
15 periods to perform an SQL select for unanalyzed feedback from the APT database. The
16 APT may determine if any filter (e.g., demographic and/or other selection criteria)
17 should be used for the analysis 5276. If so, such modifying selectors may be supplied as
18 part of the query 5278. The returned feedback records are analyzed 5280, in one
19 embodiment, using statistical frequency. For example, if a substantial number of seeker
20 provide low confidence ratings for search results of a particular state, e.g., Systems
21 Programmer, resulting from a particular query term, e.g., programmer; then this
22 information may be used to demote state structure associations. In one example
23 embodiment, each demotion may act to subtract the occurrence of a state traversal link.
24 The APT may then allow a user to make additional subset selections 5284, which result
25 in further results narrowing through more queries 5274. Otherwise the APT may
26 determine if there is any indication to terminate 5286 and end, or otherwise continue on
27 tracking user interactions with the job profile 5254.

28

Benchmarking

29 **[00384]** FIGURE 54 is of a logic flow diagram illustrating benchmarking
30 embodiments for the APT. In one embodiment, the APT allows one to select criteria

1 such as a user identifier as a target of benchmarking 5405. In one embodiment, an APT
2 interface may allow a logged in user to make a benchmark selection for any given piece
3 of advancement experience information by right clicking on a desired target; e.g., right
4 clicking on the target state in a path 4843 of Figure 48, which brings up an option to
5 perform benchmark analytics on the user and/or the target state. The APT may
6 determine the seeker's current job state identifier 5410. For example, if the user makes a
7 specific selection of a state representing advancement experience information in a state
8 graph topology as in 4843 of Figure 48, then the selected item's state ID will be the
9 seeker's current job state 5410. Such a scenario allows the seeker to perform "what if"
10 analytics on any state in the topography, and on any state in their experience path, or on
11 any state in a career path. However, if no specific state is selected, in another
12 embodiment, the seeker's most current experience information, e.g., their most current
13 employment position, may be used as the current job state 5410. The APT may then look
14 up the seeker's current experience characteristics. In one embodiment, a seeker may
15 populate their user profile with characteristics of their advancement experience
16 information. For example, for any given historical employment position, the seeker may
17 also supply characteristic attributes regarding that employment positions, attributes
18 that may include salary, geographic location, hours of work (e.g., total number of hours
19 worked per year), vacation duration, benefits, and/or the like. With this information
20 stored in the seeker's user profile, it may determine these values by finding the seeker's
21 profile record by the seeker ID and retrieving the relevant attributes stored therein 5415.
22 In an alternative embodiment, the APT may make an inquiry to companies connected to
23 the APT that the seeker claims as elements of past advancement experiences, and the
24 APT may query those employer records through a gateway to automatically retrieve the
25 attribute information. Upon obtaining the characteristics of the seeker's current job
26 5415, the APT determines if path dependant analysis is desired, and if so, the APT will
27 determine the seeker's previous job states and corresponding characteristics, iterating,
28 5422 until all such information for each, e.g., career, experience is determined 5422. Or
29 if independent path analysis is required or upon determining the path dependant
30 information 5422, then the APT may provide a list of relevant job characteristics to the
31 seeker for selection 5425. For example, after making a selection of a single or multiple
32 states in the state path graph topology and selecting "benchmark" as in 4843 of Figure

1 48, the APT may present the user with a list of available benchmarking attributes in a
2 dialogue box as shown in 5550 of Figure 55, which provides check box selectors for one
3 or more characteristics. The APT will then determine if the seeker selected one or more
4 criteria for benchmarking purposes 5427. If there is more than one selection 5427, then
5 APT may determine if default weights for each of the criteria are to be used 5430. In one
6 embodiment, an attribute database table may maintain weights for each of the
7 attributes. In an alternative embodiment, each of the selected weights may be equal. If
8 there is indication that the seeker does not want to use default weights, the seeker may
9 then provide weights 5435. In one embodiment, weights may be entered into a text box
10 5552, by making selections on a slider 5554, by making selections of weights from a pop-
11 up menu 5556 of Figure 55, and/or the like interface widget. Once the weights are
12 obtained, they may be stored 5437 in a user profile as weights associated with the
13 current job state, and/or otherwise persisted for use by the APT 5437. Once weights are
14 established 5430, 5437, or no weights are used 5427, the APT may then query the CSE
15 for the selected state and attribute data table for associated attributes to provide
16 statistical surveys and benchmarking information 5438. For example, in one
17 embodiment, the CSE may be queried with the user's current job state for all other
18 instances of that job state encountered by other job seekers that were used to make up
19 the advancement state structure in the CSE, and each of those returned query states
20 having a state_ID may be used as a basis to query an attribute table with such state_IDs
21 for associated attribute information. The returned attribute search selection results may
22 be averaged, aggregated, and or run through statistical packages such as SAS (e.g., via
23 API, pipe, messages, and/or the like) to generate covariance and other statistical
24 information and/or plots. Such analyses may include statistical processing and
25 evaluation (e.g., the calculation of means, medians, variances, standard deviations,
26 covariances, correlations, and/or the like). For example, the salary attributes from the
27 select may be aggregated, summed and averaged and as such provide a benchmark
28 against the user's current job state. In an alternative embodiment, the user may provide
29 filter information which may be supplied as query select attribute as well 5415; for
30 example, a user may wish to have the average salary for a geographic region. Upon
31 obtaining the statistical attribute information 5438, the APT may use the returned
32 information to generate visualization plots for display 5440. Thereafter the seeker may

1 make changes to the weights and job state selections so as to vary the benchmark results
2 5445, and if so, benchmarking may recurse 5420. In another embodiment, gap analysis
3 may be performed 5438 and displayed 5440.

4 **[00385]** FIGURE 55 is of a block diagram illustrating benchmarking interface
5 embodiments for the APT. In one embodiment, for the specification of attributes to be
6 benched marked 5550, the APT may use an interface selection mechanism that allows a
7 seeker to specify which characteristics are to be benchmarked, locations, settings and
8 weights for each selected attribute 5552, 5554, 5556, as has already been discussed.
9 Upon selecting an attribute, e.g., salary, the APT may generate a curve showing where a
10 curve representing a plot of other states with attribute values. For example, for salaries,
11 a curve representing the distribution of salaries for states in the state structure may be
12 plotted with 5560, 5561 and the curve allows a user to plot an orange dot along a curve
13 5563 that auto-populates a text box 5566 with format validated salary information by
14 querying the attribute database. The user may then confirm that the value is correct, if
15 the salary is monthly, annual, and if commissions are included 5567. In one
16 embodiment, orange dots will show if there are at least 4 or 5 other relevant user
17 records that exist with the specified attribute for that sate. Similar benchmark plots may
18 be achieved for salary and vacation time 5570. Also, plots employing attribute weights
19 may also be employed 5580. In yet another embodiment, multi-dimensional 5590 plots
20 may show state attribute distributions across, e.g., vacation days 5596, salary 5594, and
21 likelihood 5598 distributions.

22

Cloning

23 **[00386]** FIGURE 56 is of a mixed logic and block diagram illustrating path cloning
24 embodiments for the APT. In one embodiment, the APT may determine to periodically
25 engage and process unanalyzed seeker profiles 5601. In one embodiment, a batch
26 process may be engaged by cron at specified intervals. If the interval quantum has not
27 occurred, then the APT will continue to wait until the occurrence of the period 5601.
28 Upon occurrence and/or passage of the quantum 5601, the APT may queue all analyzed
29 profiles 5603. In one embodiment, this may be achieved by querying the APT database
30 for seeker's profiles without stored, e.g., career, state paths representing the seeker's

1 experience information. Upon returning the query results, for each such seeker profile,
2 the APT will iterate 5605. The APT will identify a, e.g., career, state path representing
3 the seeker's experience information. As has already been discussed in earlier figures, the
4 APT may map each, e.g., career, experience item (e.g., job experience) to a state and by
5 querying the state structure for a states matching each of the seeker's experience items
6 (See Figures 38, 39, et al.). As such, upon identifying all the associated states for each
7 seeker experience item, the APT may build and thereby identify a state path for each
8 seeker's experience information 5608. The APT may then store this state path in the
9 seeker profile 5619 and set a flag in the profile indicating that the path has been
10 constructed and date when that path was constructed. The APT may determine if there
11 are more profiles in the queue and if it must continue to iterate for other unanalyzed
12 profiles 5612. If there are more unanalyzed profile, the APT may continue to generate
13 career paths for each 5605, otherwise the APT may continue on and wait to repeat the
14 process upon the occurrence of the next specified quantum 5601. In this embodiment,
15 the APT continues to update state paths for every seeker's experience information.

16 **[00387]** In so doing, all seeker's paths become available for analysis. In one
17 embodiment, the APT provides an interface and a mechanism to identify and "clone" a
18 specified seeker, by finding another seeker with identical and/or similar, e.g., career,
19 state path. In one embodiment, the APT provides a web interface 5677 where an
20 interested party, e.g., an employer, may provide the experience information of a source
21 candidate to be cloned. The APT may allow the interested party to enter a search for a
22 specific candidate 5620, where results to the search terms may be listed 5622 for
23 selection by the interested party 5622. In one embodiment the interested party enters
24 terms into a search field 5620, engages a "find" button 5624, and the APT will query for
25 matching candidates and list the closest matching results 5622 from which the
26 interested party may make selections 5622. In another embodiment, the interested
27 party may search their file system for a source candidates experience information (e.g., a
28 resume) or provide such 5630. In one embodiment, the APT allows the interested party
29 to search their computer's file structure and list files for selections by engaging a
30 "submit resume" button 5626, which will bring up the a file browser window through
31 which the interested party may specify (e.g., drag-n-drop a resume document 5630) the

1 source experience information. After the interested party selects what experience
2 information it wishes to be the source 5628, the interested party may ask the APT to
3 “make a clone,” i.e., to identify another seeker having similar background and/or
4 experiences.

5 **[00388]** As such, the APT may analyze the source’s experience information and
6 generate an experience path as has already been discussed. In one embodiment, upon
7 obtaining a source experience path 5614, the APT may display the source’s path 5692.
8 The APT may then query its database for other seekers having the same experience
9 information 5616. In one embodiment this may be achieved by using the source’s
10 state_IDs for each entry comprising its experience state path as a basis to select from its
11 database. Then for the query results, for each candidate having all the matched states,
12 the APT may further filter and rank the results 5617. It should be noted that an
13 interested party may also apply attributes as a filter 5617, 5637; for example, by
14 searching for other candidates with the same career path, but that have a set salary
15 expectation (e.g., less than \$50,000); one embodiment, the filter attributes may be
16 provided in a popup menu 5637, a text field, a slider widget, and/or the like mechanism.
17 In one embodiment, the APT may provide higher ranks for matches from the same
18 regions, having experiences in the same order, and having other associated attributes
19 (e.g., salary) that are most similar to the source seeker. In one embodiment, the APT
20 may provide a pop-up menu interface to select the manner in which results are ranked
21 5647. In one embodiment, the rank clones 5646 may be displayed showing their
22 matching paths 5693, 5694, 5695. The APT may rank the results by listing the paths that
23 have the greatest number of states in common with the source more prominently than
24 those having less matching states. The APT may then display the next closet “clone” or
25 list of clones 5618, 5693, 5694, 5695 for review by the interested party. In one
26 embodiment, the interested party may send offers, propositions, solicitations, and/or
27 otherwise provide a clone with information about advancement opportunities. In one
28 embodiment, a user may make checkbox selections 5696 of the desired clones and
29 request to see the resumes of those selected clones 5644, upon which the APT will
30 provide access to those clones. In another embodiment, an offer may be made by
31 selecting the button 5644. In this way, interested parties may identify qualified

1 individuals for advancement. It should be noted that a seeker's experience information
2 may also include a state experience path comprising their education history. As such, in
3 one embodiment, the APT may clone not only a seeker's, e.g., career, path experience,
4 but also their education path experience.

5 **Advancement Taxonomy**

6 **[00389]** FIGURE 57 is of a mixed block and data flow diagram illustrating
7 advancement taxonomy embodiments for the APT. In one embodiment, the APT may
8 act as a "rosetta stone" as between a state structure (e.g., a states table) 5719f, 5710,
9 attribute information (e.g., an attributes table) 5719j, and an experience structure 5719h.
10 In one embodiment, the APT may take process experience structure records 5701 from
11 the experiences table 5719h and map them to the appropriate state. Similarly, in one
12 optional embodiment, the APT may take attribute records 5703b from the attributes
13 table 5719j and map them to the appropriate state. As a consequence, the state structure
14 and its states 5719f will be associated with both the experience structure and its
15 Occupational Classification codes (hereinafter "OC_code") and with attribute
16 information and its attribute_ID. An example of an OC_code is an Occupational
17 Information Network (ONet) Standard Occupational Classification (SOC) code.
18 Similarly, once an association is made for either experience structure 5722a or attribute
19 information 5723a into a state 5719f, 5711, the APT may push (i.e., cause a database
20 write of a value in a record field) a unique state_ID into the experience table 5733 and
21 attribute table 5734. As such, with an experience table having a state_ID, it may use that
22 state_ID to access the appropriate state in a state structure, and in turn, look up an
23 associated attributes table entry; and vice versa for the attributes table being able to
24 map to the experience structure. In another embodiment, the APT may push its own
25 state_ID 5733 and an attributes_ID 5723b into the experience table, and its own
26 state_ID 5734 and an OC_code 5722b into the attributes information 5719i so as to
27 minimize database traversal. In another embodiment, simultaneous writes 5722, 5723,
28 5733, 5734 may take place.

29 **[00390]** In one embodiment, the APT 5708 may use the experience table's title, job
30 description, skills, category, keyword and other field values as basis to discern and map

1 to a matching state in the state structure, as has already been discussed in Figures 38 et
2 seq. In one embodiment, the CSE 5708 may similarly use values stored in the attributes
3 table 5719j. However, in an alternative embodiment, attribute information 5719h and
4 experience information 5719i may be related by being assigned by administrators who
5 will fine tune said associations.

6 **[00391]** In another embodiment, attributes 5719i that are related to experience
7 information 5719 h assume a relationship that is discerned as between the experience
8 information 5719h and a state 5719f. For example, a career system, such as
9 Monster.com, may track attributes for various job listings that may be stored in a job
10 listing table. Such job listings often have numerous attributes and many other attributes
11 may be discerned through statistical analysis of seekers that interacted with job listings.
12 These job listings often have an OC_code, and as such may already be related to
13 experiences. As has already been discussed, the APT may associate unmapped
14 experiences 5701, 5719h to states 5710, 5719f, and when so doing, it may relate attribute
15 information 5719i that has already been associated to the unmapped experience 5719h
16 to states in the same process. In another embodiment, structured resume information,
17 i.e., experiences 5719h may be mapped to an OC_codes as described in patent
18 applications: serial no. 11/615,765 filed December 22, 2006, entitled "APPARATUSES,
19 METHODS AND SYSTEMS FOR AN INTERACTIVE EMPLOYMENT SEARCH
20 PLATFORM," attorney docket no. 17199-030US2; and serial no. 11/615,768 filed
21 December 22, 2006, entitled "A METHOD FOR INTERACTIVE EMPLOYMENT
22 SEARCHING AND SKILLS SPECIFICATION," attorney docket no. 17199-030US1; the
23 entire contents of both applications is hereby expressly incorporated by reference.

24 **[00392]** FIGURE 58 is of a block diagram illustrating advancement taxonomy
25 relationships and embodiments for the APT. Further to the base taxonomy introduced
26 in Figure 57, the APT may similarly forge relationships of various information bases
27 through the state structure as shown in Figure 58. Figure 58 shows one embodiment of
28 some of the APT Controller's database component's tables cast into numerous types of
29 connections and relationships (one to one, 1:1; one to many, 1->M, many to many, M-
30 >M; many to one, M->1) as between various structures and supporting database tables.

1 It should be noted that in one embodiment, the APT database tables are all
2 interconnected in a manner where every table is related to every other table.

3 **Advertisement Generation, Selection And Distribution System** 4 **Registration**

5 **[00393]** The present disclosure includes a discussion of systems, methods, and
6 apparatuses for generating, managing and distributing advertisements. The disclosed
7 system is particularly suited for use with a system user's (sponsor's) one or more or
8 narrow listing offers (base data entries), where traditional passive advertising would be
9 ineffective. The disclosed examples are discussed in the context of job placement
10 advertisements, which may be considered as narrow listing offers because they generally
11 are only relevant to a small audience of qualified applicants. Furthermore, job
12 placement advertisements are usually directed to attracting applicants for a limited pool
13 of available positions. It is to be understood that while the system is described in the
14 context of job placement advertisements, the system provides an administrator with
15 significant flexibility and freedom to configure the system for any other number of
16 narrow listing offer or advertisement applications, such as classified product listings
17 including automobiles, personals, real estate, etc.

18 **Overview**

19 **[00394]** Figure 59 provides an overview of various entities that may interact with
20 the system at various points during system utilization. According to an embodiment of
21 the invention, the Career Advertisement Network ("CAN") 5900 is a central element in
22 the system facilitating the functionality described herein. A system user (e.g., an
23 employer) 5930 connects with the CAN 5900 over the internet 5950. The system user
24 may either create a new advertising campaign or manage an existing advertising
25 campaign. According to an embodiment, the advertisements created and distributed by
26 the CAN 5900 are based on corresponding narrow listings, configured in this example
27 as job opportunity listings stored within a job listing web site's job listing database (e.g.,
28 Monster DB) 5910.

1 **[00395]** The system user 5930 uses the CAN 5900 to create advertisements based
2 on selected job listings (based data entries) stored within the job listing DB 5910.
3 Advantageously, the CAN 5900 incorporates a flexible system user interface, wherein a
4 system user (e.g., employer or sponsor)5930 may determine how much
5 interaction/input they wish to have in the advertisement creation process. Depending
6 on the needs of a particular user, most of functionality associated with the CAN 5900
7 can be implemented based on system-driven processes. Alternately, some system
8 functionality may be may configured so that a system user 5930 can work with a system
9 administrator 5905 to create/manage an advertisement campaign.

10 **[00396]** As illustrated in Fig. 59, the CAN 5900 coordinates creating and
11 distributing advertisements with content provider system 5915. In an embodiment, the
12 content provider system 5915 creates and distributes online content from the content
13 provider databases 5925 or other sources. The CAN system creates and distributes
14 advertisements that are incorporated into the content as it is presented to an internet
15 user 5935. Generally, the content provider may be configured as a web site that
16 provides an internet user 5935 with online content such as news, entertainment, sports,
17 online media or other types of online content. Furthermore, depending on the actual
18 implementation, the content provider may be an affiliate web site, a partner web site or
19 even hired as an advertisement placement provider.

20 **[00397]** For example, the content provider 5920 may be configured as a sports
21 news web site. The content provider 5920 distributes various sports news content from
22 the content providers database 5925. The CAN 5900 may be configured to coordinate
23 incorporating CAN generated advertisements into the content distributed by the content
24 provider's system 5915. The CAN 5900 is configured to create the advertisement based
25 on a variety of factors, some of which may include: a content provider's content, a
26 content provider's advertisement system configuration, web user 5935 characteristics,
27 and/or any variety of other distribution metrics established by the system user 5930 or
28 system administrator 5905. In some embodiments, the CAN 5900 may be configured
29 with an advertisement tracking module configured to track and record data associated
30 with a web user's interaction with the displayed advertisement.

1 **[00398]** Figs. 60A-60B are high-level flow diagrams illustrating a system user
2 campaign management functionality associated with the system: campaign creation and
3 campaign management. The system user attempts to log onto the CAN in step 6000. If
4 the user is not a registered user, the system user must go through a registration process.
5 However, at this point (step 6005) the system branches between campaign management
6 (6010-6030) and campaign creation (in Fig. 60B).

7 **[00399]** If the system user selects the campaign creation branch in step 6005, the
8 system user moves into a campaign selection module 6010. In one embodiment, after
9 the system user's campaign has been selected, the system user can then choose an ad
10 from a management dashboard (shown in Fig. 61A) in step 6015. In an alternate
11 embodiment, the system maintains a CAN widget (shown in Fig. 61B) instead of the
12 management dashboard. Step 6020 involves modifying the parameters of the campaign
13 in step 6030 or the advertisements that make up a campaign in step 6030. By way
14 example only, the system user can modify campaign scheduling parameters, add or drop
15 advertisements from a campaign, modify the advertisement distribution scope; or view
16 campaign performance metrics. Furthermore, the system user can modify single
17 advertisement parameters, designate a particular advertisement for evolution engine
18 optimization or view specific advertisement performance metrics.

19 **[00400]** Fig. 60B illustrates aspects of the system user campaign creation process.
20 The system user logs onto the system in step 6050 and enters the campaign creation
21 module in step 6055. The system user registers with the system and then is directed to
22 the create new data entry module in step 6065. The system is configured so that the
23 system user can upload existing data entries to the system in step 6070. Alternately, the
24 system user may interact with a listing creation module to create data entries for
25 incorporating into a campaign. Once the system user designates the particular data
26 entries for conversion to advertisements, the system user works with the CAN system to
27 step through the ad creation and distribution process (6075-6095). More specifically,
28 the system user establishes conversion/extraction rules in step 6075 that the system
29 users to extract the key terms from a base data entry. The key terms are then used to
30 populate an empty advertisement template.

1 **[00401]** In step 6080, the system user selects either basic or premium targeting
2 parameters. The system identifies advertisement distribution opportunities based on a
3 limited data pool. In contrast, if the user establishes premium targeting parameters, the
4 system uses an expansive data pool to determine when to incorporate an advertisement
5 into distributed content. The expansive data pool may be derived from a web user's
6 surfing characteristics obtained through the system or system partners/affiliates. After
7 establishing the targeting parameters, the system user establishes distribution
8 parameters in step 6085. The distribution parameters are user driven parameters, (e.g.,
9 a user can determine the distribution scope for an advertisement - whether the
10 advertisement should be incorporated with news content, multiple sports news sites
11 content or a single sports news site's content, or a single sports news site's football
12 content, etc...). In some embodiments of the invention, the user establishes
13 advertisement evolution parameters in step 6090 and performance metrics in step
14 6095.

15 **[00402]** Figs. 61A and 61B illustrate an campaign management dashboard and a
16 campaign management widget, respectively. In Fig. 61A, the management dashboard
17 6100 can be configured to display distribution parameters, performance metrics and/or
18 a variety of other parameters associated with a particular campaign 6160. In Fig 59, the
19 dashboard is configured to display advertisement parameters including the distribution
20 frequency 6173, the advertisement generation rule set user to create the advertisement
21 6176, the number of impressions 6179 or the number of click-throughs 6182 associated
22 with a particular advertisement 6170, 6171, 6172, 6190, 6191. Advantageously, through
23 the management dashboard, the system user can easily evaluate the
24 performance/generation/distribution metrics associated with an advertisement. In
25 some implementations, the dashboard may be configured to display campaign overview
26 parameters 6155 (e.g. the total number of qualified resumes that can be correlated to the
27 advertisements within a campaign, the setting campaign specific distribution
28 parameters, etc...). Also, in some embodiments the dashboard may be configured to
29 display the advertisement in a popup window when the system user mouses over the
30 advertisement identifier. In Fig. 61B, the advertisement management widget 6125 is
31 illustrated displaying campaign-level parameters (including sponsor name 6120 and

1 campaign name 6122). However, it is to be understood that the widget can also be
2 configured with display functionality similar to that described above with regard to the
3 management dashboard. The system user may change the displayed
4 parameters/metrics on the widget by selecting the desired setting from the file menu
5 6124 and/or the settings menu 6126.

6 **[00403]** Fig. 62A is an example of a base data entry (narrow listing) directed to a
7 software engineering job opportunity hosted by an employment opportunity host web
8 site (e.g., Monster.com). By way of example only, the CAN may be configured to identify
9 and extract certain key parameters from the full data entry during the advertisement
10 generation process. It is to be understood that the CAN may be configured to coordinate
11 with a variety of data entry host web sites beyond the employment opportunity context.
12 For example, the key term identification and extraction process may be keyed to a
13 variety of shopping web sites, real estate listing web sites, and/or travel web sites. In
14 the example illustrated in Fig. 62A, directed to an employment opportunity example, the
15 CAN extracts key terms such as the name of the data entry sponsor 6255. Alternately,
16 depending on the characteristics of an employment opportunity, the key terms may be
17 based on parameters such as: opportunity title 6260, offered salary 6265, geographic
18 location, 6270, job description 6275, required qualifications 6280, and/or key benefits
19 6285. Depending on the established generation rule set, different combinations of key
20 terms may be used to generate the advertisement.

21

Ad Generation

22 **[00404]** Figure 62B shows an exemplary process 6220, which would be performed
23 by the MEN, for generating an ad 6221 from a job listing 6250. The first step 6220a of
24 the ad generation process begins with the creation of an ad shell 6221a. Next in step
25 6220b, the job listing 6250 is input to the process. The job listing is then parsed to
26 extract the job listing terms 6220c. In the parsing step, an advertisement generator
27 module categorizes and tags the information extracted from the job listing. For
28 example, the listing may be parsed to extract one or more of the following: the job title,
29 company, job location, job category, experience required, level of education, salary,
30 company logo, graphic, etc. The parsing can be implemented in a number of ways. For

1 example, the job listings can be pre-tagged with a mark up language, such as XML, to
2 identify the contents of the job listing. In another embodiment, the relevant data from
3 the job listing could be extracted using data entry personnel. In a further embodiment,
4 the data can be extracted from the job listings using natural language processing
5 software, such as the open source program Mallet (mallet.cs.umass.edu) to identify and
6 categorize the relevant information.

7 **[00405]** The extracted listing terms are then used to determine the key terms
8 6220d. The key terms represent the extracted and categorized portions of the job
9 listings. For example, in the context of job listings key terms might include job title,
10 company, logo, location, full-time/part-time, industry, category, experience, career
11 level, education required, salary, a descriptive tag, and a full description. The key terms
12 can then be used to create the generated ad 6221. Since the detailed job listing is likely
13 to have more key terms than can be reasonably inserted into the ad, only a subset of the
14 key terms might be used in the ad. In one embodiment, for example, the decision of
15 which key terms to use in the ad can be based upon templates implemented as ad
16 generation rule sets. For example, the template might indicate that the job title is used
17 as the title of the ad and company and salary are used as detail for the ad.

18 **[00406]** In the alternative, the key terms of the ad can be examined on an ad hoc
19 basis to look for unique details or selling points that are used to choose which terms to
20 use in the ad. This could be embodied by various rule based decisions programmed into
21 the ad generation algorithm. One rule, for example, might check if the salary for the
22 type of job listed is higher than average. If so, the system could choose to highlight the
23 salary by referring to it the title of the ad. Another rule might check to determine if the
24 job is in a desirable location. If so, this fact would be highlighted by the generated ad. A
25 further rule might include a graphic or company logo in the ad if one is present in the
26 job listing. A further rule could include a company logo if the company is above a
27 certain size or is otherwise well known because candidates are more likely to apply for a
28 job with a company they are familiar with.

29 **[00407]** Using either of the described techniques, or a combination of the
30 techniques, one or more of the key terms are selected for the title 6220e and inserted
31 into the ad shell 6221b. In addition, the details for the body of the ad 6220f are also

1 selected from one or more of the key terms, which are inserted into the ad shell 6221c to
2 create a final ad. In adding the title and details into the ad, the ad generation algorithm
3 might add additional language to the inserted key term or the facts provided by the key
4 term might be rewritten. For example, as shown in the Figure 62B embodiment, the job
5 listing term "Qualification - Two or more years of extensive experience" is rewritten as
6 over 2 years of experience.

7 **Ad Optimization**

8 **[00408]** Figure 63 shows the generation of multiple ads from a single job listing
9 6311 (based on full data entry 6250 from Fig. 62A). Generating and distributing
10 multiple ads can based on the same data entry optimizes ad effectiveness. To create
11 multiple ads, the advertisement generator 6320 chooses generation rule sets (different
12 key terms for the title and ad detail) used to generate ads 6321a-d. The generated ads
13 6321a-d could be used to optimize the ad placement process by displaying the ads to
14 multiple candidates to determine which of the ads are most effective. Or the ads could
15 be targeted based upon information known about the site visitor to provide the most
16 effective ad for the particular individual that will be viewing the ad.

17 **[00409]** As shown in Figures 64-65, the effectiveness of each of the multiple
18 generated ads can be determined by displaying the ads to various site visitors and
19 recording their responses. The site visitors' reactions to the ads can be gauged in a
20 number of ways. One basic measurement of ad success is click-throughs, i.e., the
21 number of site visitors that click on the ad link for more information. An enhanced
22 measurement of ad effectiveness is the number of site visitors that click on the ad and
23 ultimately submit a job application. This metric is in some ways better than simple
24 click-throughs because a submitted application demonstrates that the site visitor, not
25 only found the ad compelling, but also found the underlying offer relevant enough to
26 submit an application/resume. In contrast, a click-through metric may only indicate that
27 the ad is broadly appealing, without identifying whether the ad is attracting relevant
28 candidates. Another metric of ad effectiveness would track whether the candidates who
29 submitted applications based on clicking through to the ad were ultimately hired. While
30 this information would indicate that the ad brought in a suitable candidate, it may only

1 be useful when there are a relatively large number of offers to be fulfilled. For example,
2 if only one position needs to be filled, the hiring of a candidate ends the need for the ad.

3 **[00410]** As shown in Figure 64, to gather the ad effectiveness data the
4 advertisement server displays each of the ads 6421a-d to groups of site visitors 6405a-d,
5 either directly or indirectly via an affiliate content server. Data about the effectiveness
6 of each of the ads, e.g., click-throughs or applications submitted, is communicated back
7 to the advertisement server. This information is associated with its respective ad and
8 stored in an ad data store. For example, if one of the site visitors 6405a responds
9 positively to ad 6421a, that fact is recorded in the ad data store. Additionally, meta-data
10 about the site visitor 6405a, such as their location, education level, etc., may also be
11 associated with the record indicating ad success. This information can then be used to
12 determine whether the particular ad is effective at reaching a target audience.

13

Ad Evolution

14 **[00411]** An alternative to judging the effectiveness of particular ads, would be to
15 judge the effectiveness of the various ad generation techniques used to create the ads.
16 Determining the effectiveness of the ad generator has the advantage of being applicable
17 beyond the limited case of a specific ad. For example, determining that a particular ad
18 generation technique or algorithm is effective allows that ad generator to be used to
19 produce additional effective ads for other listings. A further advantage could be had by
20 associating the ad generator effectiveness with characteristics of the job listing for which
21 the ad is created or with characteristics regarding the site visitors who responded to the
22 ad. For example, measuring ad effectiveness might demonstrate that certain types of
23 candidates value high pay over a flexible work schedule or visa versa. An employer, or
24 the system, could then choose the ad generator that will best attract the most relevant
25 candidates. That ad generator would be used to generate the ad.

26 **[00412]** Figure 65 shows an embodiment for optimizing the effectiveness of
27 particular ad generators. As shown in the figure, ad server 6520 communicates with the
28 ad data store 6521 to record the effectiveness of various ad generators 6520a-d, rather
29 than the ads themselves. The ad generators 6520a-d might represent any of the various
30 ad generation techniques discussed above. For example, each of the ad generators may

1 be a different ad generation template that is applied to the listing to generate the ad. Or,
2 the ad generators may be rule based software algorithms that make ad hoc
3 determinations based on the key terms extracted from the job listing.

4 **[00413]** Similar to the ad optimization discussed above, information recorded
5 about the effectiveness of the ad generators might include, the total responses received
6 by its generated ads, which could be measured as click-throughs. The responses may
7 also, or alternatively, be recorded with associated data about the responding site visitor,
8 which would enable the system to use ads/ad generators targeted to the site viewer in
9 questions or targeted to the desired candidates. The quality of the ad generator can also
10 be measured by the number of candidates successfully hired after responding to its
11 generated ads. In the context of ad generators, this information is particularly valuable
12 because successful ad generators can be used to produce ads for other employers based
13 on their listings. The ad optimization data recorded might also include details about the
14 job in question, which could then be used to identify whether particular generators work
15 better for certain jobs. For example, a particular ad generator might work better at
16 writing ads compelling to sales people, while another ad generator might write ads
17 compelling to researcher scientists.

18

Ad Targeting/Distribution

19 **[00414]** Figure 66 shows the an embodiment for the placement of ads through the
20 interaction of the advertisement server 6620 and an affiliate content server 6630. In
21 this embodiment, one or more affiliate content servers 6630 request ad serving from the
22 advertisement server 6620. The request may include stored or categorical information
23 about the site visitor 6605 from the visitor data store 6631. The information about the
24 site visitor 6605 could have come from a number of sources. For example, the site
25 visitor information could have been derived during the current web content
26 session/interaction between the affiliate content server and the site visitor, such as
27 information derived from the site visitors viewing and interaction with affiliate content
28 pages 6606a and 6606b. Or, it could come from information received and stored during
29 the site visitor's previous interactions with the affiliate content server. In addition, two
30 or more different affiliate content servers might share or combine their saved site visitor

1 data. Site visitor data might also have been previously stored by the advertisement
2 server. For example, the site visitor data might be associated with a site visitor
3 identifier, e.g., a cookie, the advertisement server might keep data indicating visitor
4 characteristics associated with the identifier. In this way, the advertisement server can
5 correlate information about the site visitor that was previously provided by one or more
6 affiliate content servers and the advertisement server can track which ads were of
7 interest to the site visitor. For example, if the site visitor had previously clicked on ads
8 for java programmers, the advertisement server will have a highly relevant data point to
9 indicate that programming jobs are of interest to that visitor.

10 **[00415]** In one embodiment, the site visitor's use of the affiliate content server
11 service results in information about the site visitor being provided to the affiliate
12 content server. For example, the affiliate content server may provided personality tests,
13 skills evaluations, I.Q. tests, career recommendations, etc., such as the tests provided at
14 tickle.com. When the site visitor uses the affiliate content server, information about
15 himself or herself is gathered. In another example, the site visitor may have filled out a
16 user profile that provides detailed information about him or herself. In another
17 example, the affiliate content server may provide social or business networking services,
18 such as myspace.com or linkedin.com. In using such a service, the site visitor may
19 create a web page providing details about the users skills and interests. In another
20 example, the affiliate content server may provide career services like monster.com. In
21 using the service the site visitor may have uploaded a resume or created a user profile
22 that indicates the user's skill set, educational background, work experience, interests,
23 etc. In this way, it can be seen that the affiliate content server and the listings server
24 may be one in the same entity.

25 **[00416]** Regardless of the mechanism in which information about the site visitor is
26 recorded. Some or all of the recorded information is provided by the affiliate content
27 server 6630 to the advertisement server 6620. In addition, information about the site
28 visitor might be added based on context of the affiliate content site or added by
29 categorizing the affiliate content site itself. In other words, a particular affiliate content
30 site might indicate for itself that particular characteristics apply to all of its site visitors,
31 or this determination may be made by the operators of the advertisement server. For

1 example, an affiliate content site directed to semiconductor industry news might set
2 forth that all of its site visitors are involved in the semiconductor industry. Or, the
3 advertisement server might make that determination. In addition, the affiliate content
4 server operator may want to have more control over the companies and types of ads that
5 are displayed on its site and could set characteristics accordingly. For example, an
6 affiliate content site devoted to alternative energy might want to exclude advertisements
7 from oil companies.

8 **[00417]** Once the characteristics for a particular site visitor are received and
9 gathered by the advertisement server, the advertisement server can use this information
10 to retrieve an ad from the ad data store 6621 and provide an advertisement that is
11 narrowly targeted to suit the site visitor. The search for an ad targeted to the site visitor
12 in question can be accomplished by database searches over the ad data store and
13 associated matching algorithms. For example, if the affiliate content server provides the
14 advertisement server with three facts about the site visitor, the advertisement server can
15 search the ad data store for ads that most closely match the provided criteria. Any
16 discovered ad 6622 can then be inserted into content page 6606c and presented to the
17 site visitor by the affiliate content server 6630. If multiple matching ads are discovered,
18 multiple ads can be shown to the site visitor or the ads can be narrowed down based on
19 additional criteria, such as the amount of revenue generated by the ad, the amount of
20 time since the ad was last displayed, etc. The number of ads displayed to the site visitor
21 may be determined by the number of ad positions made available by the affiliate content
22 server. If the number of relevant ads matching the site visitor is less than the number of
23 available ads, the extra ad space can be filled in with broad based ads or ads driving
24 traffic for the system as a whole. For example, ads might drive traffic to the original
25 listing service, e.g., the job listing server.

26 **[00418]** Figure 67 shows an embodiment of the disclosed system through which
27 the advertisers 6700, e.g., employers with job listings, can have the placement of their
28 ads tailored to the site visitors to which the ads will be presented based on site visitor
29 characteristics. As shown in the figure, site visitor data 6705a-n is stored by the affiliate
30 content server 6730. However, the amount of information known about the various site
31 visitors is not uniform, as represented by the different sizes of the site visitor data files.

1 So, for example, more information is known about site visitor 6705a than is known
2 about site visitor 6705f.

3 **[00419]** With this collection of information, an advertiser can choose to only
4 display ads to site visitors with a certain level of known details or with specific known
5 details. For example, an advertiser might only want his ads shown to people working in
6 information technology and living in or around New York City, or only to known college
7 graduates. The advertisement server can also price the ads based upon the level of
8 details known about the site visitor. For example, an advertiser might buy fewer, more
9 expensive, narrowly targeted ads. Or, the advertiser could buy a larger number of
10 cheaper less targeted ads.

11 **[00420]** In an alternative addition to the embodiment, the affiliate content
12 provider might choose the amount of site visitor information they would like to provide
13 to the advertisement server. In this way, if a site provider is sensitive about releasing
14 particular information or certain amount of information, it can choose to release less
15 than all of its information to the advertisement server.

16 **[00421]** Another aspect of the disclosed invention is that the compensation
17 provided to the affiliate content server can be determined based upon the amount of
18 information about the site visitor provided. In this way, affiliate content providers can
19 be rewarded by providing more information, which will result in more effective ad
20 choice and placement. If an affiliate content provider would like to provide reduced
21 information about their site visitors, they can receive broader less profitable ads.

22 **[00422]** Figure 68 shows an exemplary process and flow diagram for ad serving in
23 accordance with the disclosed system. One or more advertisers 6800 submit a request
24 for advertising, which is received by the ad server 6820a. Along with the request, the
25 advertiser may include information indicating the site visitor criteria identifying
26 preferred candidates for the advertisement and the amount the advertiser is willing to
27 pay. In the alternative, the relevant site visitor criteria can be extracted from the
28 content of the listing. The advertiser might also have specific requests. The advertiser,
29 for example, may require that the ad is only shown to site visitors in a particular
30 geographical region or with a specific education level. Or, the advertiser might simply

1 indicate that its ads should only be displayed to site visitors that match a certain number
2 of its listing criteria. For example, the advertiser might choose a level where the ad is
3 only shown to site visitors that match three or more criteria relevant to its offer. The ad
4 server can then store the advertisement request details 6820b in the ad data store 6821.

5 **[00423]** An affiliate content server 6830 will interact with the ad server to request
6 ads to be displayed in the affiliate content server's content pages 6806. The affiliate
7 content server 6830 will display its content to site visitor 6805. In interacting with the
8 site visitor, or through previous interactions with the site visitor, the system develops
9 site visitor data 6831. The site visitor data 6831 may be passed to the ad server along
10 with the request for ad serving. Upon receipt of the ad serving request 6820c, the ad
11 server searches 6820d the ad data 6821 for ads matching the request, including any
12 supplied visitor data. If multiple potentially matching ads are discovered in the search,
13 the ads with the highest price and/or the closest match to the submitted data are served
14 6820e. If only one ad is discovered, that ad is served. The ad to be served 6821 is
15 inserted into the affiliated content server's content page 6806. This can be done directly
16 by the ad server 6820 or the ad could be provided to the affiliate content server for
17 insertion into the content.

18 **[00424]** In an alternative embodiment, instead of using pre-generated ads stored
19 in the ad data store, the advertisement server may directly search the job listings data
20 store for relevant job listings. If a job listing matching the ad request is discovered, an
21 ad can be automatically generated using the ad generation techniques described above.
22 The ad can then be inserted into the content displayed to the site visitor. In such system
23 the ad data store could either be eliminated or it could be used to store details regarding
24 the advertisers' requests, budgets, criteria, and ad pricing.

25 **[00425]** Figure 69 shows an embodiment disclosing advertisement distribution in
26 accordance with another aspect of the disclosed system. As shown in the figure,
27 advertisement server 6920 is in communication with affiliate content servers 6930 and
28 advertisement distribution server 6925. The affiliate content server displays site
29 content pages 6906. The figure demonstrates a number of methods for distributing ads
30 6922 to the affiliate content server. In one embodiment the advertisement is passed
31 from the advertisement server directly to the affiliate content server. It should be noted

1 that ad could either be delivered as the content of the ad itself or it can be delivered as a
2 symbolic link to the ad content. In another embodiment the ads can be served by a
3 dedicated advertisement distribution server, which might be embodied by an existing
4 commercial ad distribution network. In this way, the advertisement server can take
5 advantage of large commercial distribution networks and have access to affiliate content
6 serves that would like to source their advertisements through such a provider.

7 **[00426]** Fig. 70 illustrates aspects of the management dashboard according to
8 another embodiment of the invention. In this implementation, the performance metrics
9 are displayed directly next to generated advertisements 7050. As illustrated, a system
10 user can manipulate a series of drop-down boxes 7060 in order to view a variety of
11 performance metrics associated with the displayed advertisements 7050.

12 **[00427]** It is to be understood that the system is customizable based on the
13 particular needs of a system user. Figs. 71A-71C illustrate three different system
14 configurations with functionality that customized for the particular user. The customer
15 system configuration illustrated in Fig. 71A provides enough functionality so that the
16 customer can interact with the system to establish generation/distribution/performance
17 parameters or elect to implement the system default parameters. Fig. 71B illustrates an
18 administrator's system configuration - enabling the administrator full access to all
19 system components. In contrast, Fig. 71C illustrates an account representative system
20 configuration with access to limited generation/distribution/performance metric
21 functionality.

22

Advertisement Generation

23 **[00428]** The present disclosure includes a discussion of systems, methods, and
24 apparatuses for generating, managing and distributing advertisements. The disclosed
25 system may be configured to generate advertisements based on a system user's
26 (sponsor's) one or more or narrow listing offers (base data entries), where traditional
27 passive advertising would be ineffective. The disclosed examples are discussed in the
28 context of job placement advertisements, which may be considered as narrow listing
29 offers because they generally are only relevant to a small audience of qualified

1 applicants. Furthermore, job placement advertisements are usually directed to
2 attracting applicants for a limited pool of available positions. It is to be understood that
3 while the system is described in the context of job placement advertisements, the system
4 provides an administrator with significant flexibility and freedom to configure the
5 system for any other number of narrow listing offer or advertisement applications, such
6 as classified product listings including automobiles, personals, real estate, etc.

7 **[00429]** Figure 72 provides an overview of various entities that may interact with
8 the system at various points during system utilization. According to an embodiment of
9 the invention, the Career Advertisement Network ("CAN") 7200 is a central element in
10 the system facilitating the functionality described herein. A system user (e.g., Employer
11 or base data entry sponsor) 7205 connects with the CAN 7200 over the internet 7210 to
12 either create a new advertising campaign or manage an existing advertising campaign.
13 According to an embodiment, the advertisements created and distributed by the CAN
14 7200 are based on corresponding narrow listings, configured in this example as
15 employment opportunity listings stored within an job listing web site's job listing
16 database (DB) 7215.

17 **[00430]** The system user 7205 uses the CAN 7200 to create advertisements based
18 on selected job listings stored within the job listing DB 7215. Advantageously, the CAN
19 7200 incorporates a flexible system user interface, wherein a system user 7205 may
20 determine how much interaction/input they wish to have in the advertisement creation
21 process. Depending on the needs of a particular user, most of functionality associated
22 with the CAN 7200 can be implemented based on automated system processes.
23 Alternately, some system functionality may be may configured so that a system user
24 7205 can work with a system administrator 7220 or an interactive system module to
25 create/manage an advertisement campaign.

26 **[00431]** As illustrated in Fig. 72, the CAN 7200 coordinates creating and
27 distributing advertisements with content provider system 7225. In an embodiment, the
28 content provider system 7225 creates and distributes online content from the content
29 provider databases 7235 or other sources. The CAN system creates and distributes
30 advertisements that are incorporated into the content as it is presented to the a web user
31 7240. Generally, the content provider may be configured as a web site that provides a

1 web user 7240 with online content such as news, entertainment, sports, online media or
2 other types of online content. Furthermore, depending on the actual implementation,
3 the content provider may be an affiliate web site, a partner web site or even hired as an
4 advertisement placement provider.

5 **[00432]** For example, the content provider 7230 may be configured as a sports
6 news web site. The content provider 7220 distributes various sports news content from
7 the content provider's database 7235. The CAN 7200 may be configured to coordinate
8 incorporating CAN generated advertisements into the content distributed by the content
9 provider's system 7225. The CAN 7200 is configured to create the advertisement based
10 on a variety of factors, some of which may include: a content provider's content, a
11 content provider's advertisement system configuration, web user 7240 characteristics,
12 and/or any variety of other distribution metrics established by the system user 7205 or
13 system administrator 7220. In some embodiments, the CAN 7200 is configured with an
14 advertisement tracking module configured to track and record data associated with a
15 web user's interaction with the displayed advertisement.

16 **[00433]** Fig. 73 is a high-level flow diagram illustrating aspects of the
17 advertisement generation process. The system receives an advertisement from content
18 provider/affiliate web site. The advertisement request may specify a variety of
19 advertisement distribution parameters. There are two aspects of the
20 targeting/distribution process that are used during the generation process – content
21 advertisement formatting parameters and the actual content associated with the content
22 provider requesting the advertisement.

23 **[00434]** The system receives the advertisement request from the
24 targeting/distribution module in step 7300 and the system extracts these parameters
25 from the request. Based on the request content provider parameters, the system
26 retrieves a base data entry 7310 (e.g. If the content provider is a computer engineering
27 news web site, the system may select base data entries directed to software engineering
28 jobs). Further, the system retrieves an advertisement template 7315 based content
29 provider's advertisement request parameters detailing the format parameters associated
30 with the requested advertisement. In step 7320, the system processes the base data
31 entry in accordance with the data extraction rules. As will be described in greater detail

1 below, this process may be driven by a user interacting with a system advertisement
2 generation module or the process may be automated.

3 **[00435]** The extracted base data entry parameters are used in step 7330 to
4 populate the retrieved advertisement template from step 7315. After populating the
5 template, the generated advertisement is transferred to the system distribution module
6 in step 7340. Before it is actually distributed to the content provider, the system
7 generates an advertisement tracking record in step 7350. As such, the system can
8 maintain a record of performance metrics that detail a web user's interaction with the
9 distributed advertisement.

10 **[00436]** Fig. 73 also discusses aspects of a landing page generation process,
11 according to an embodiment of the system. Once the advertisement has been
12 distributed to the content provider in step 7350, the system may receive web user
13 interaction data or a landing page generation indicator in step 7360. When a web user
14 clicks on a distributed advertisement, the advertisement transmits a landing page
15 generation indicator back to the system indicating that the web user wants additional
16 information about the advertised employment opportunity. The system retrieves
17 landing generation parameters associated with the base data entry in step 7370.

18 **[00437]** Depending on the particular implementation, the generated landing page
19 may be the base data entry or the base data entry incorporated into a landing generation
20 template. In step 7380, the system retrieves the base data entry and a corresponding
21 landing template, which is populated with the information from the base data entry.
22 The template may also incorporate additional functionality associated with the host
23 entity associated with the base data entry. For example, in the employment opportunity
24 context, the landing page may be configured with resume building functionality or an
25 application submission module.

26 **[00438]** Fig. 74A illustrates a flow diagram illustrating aspects of the advertisement
27 retrieval process. In step 7400, the system receives the ad request parameters from the
28 content provider. There are two primary elements that comprise the ad request
29 parameters – ad content parameters 7410 and ad formatting parameters 7420. The
30 base data entry selected for distribution is selected based on the type of content

1 associated with the advertisement request (e.g., a computer software evaluation content
2 provider will likely be matched with a computer related base data entry). Also included
3 in the advertisement request are the advertisement formatting specifications that detail
4 the advertisement requirements associated with a requested advertisement.

5 **[00439]** The request parameters detail factors including whether an advertisement
6 template should include a title element 7420, a descriptive element 7422, a job type
7 element 7424, a job location element 7426, advertisement size constraints 7430,
8 advertisement text constraints 7432, advertisement image constraints 7434, web user
9 redirection link constraints 7436. Based on these advertisement request parameters,
10 the system determines what type of template to retrieve in step 7440.

11 **[00440]** Fig. 74B-74E illustrate several examples of advertisement templates that
12 may be implemented by the system. The templates may be organized based on a variety
13 of template element characteristics. For example, Figs. 74B-74E may be grouped into
14 two main categories: 1. by the number advertisements associated with a template and/or
15 2. whether the template is based solely on textual data or a combination of image and
16 textual data. Fig. 74B illustrates a premium template 7450 configured for displaying
17 single advertisement. It may be configured to display an advertisement sponsor's image
18 7452, as well as Key Term 7454 (selected to attract the attention of a web user), Job
19 Description element 7456 (often a brief overview to provide a web user some additional
20 detail beyond the key term), and Job Qualification element 7458 (including a base job
21 requirement helps ensure that only qualified applicants pursue the advertisement).

22 **[00441]** Fig. 74C illustrates a premium template with an additional data host
23 element incorporated into the template. Web users may be more likely to pursue an
24 advertisement, if they know it is hosted by a reputable data entry host. As illustrated,
25 template 7460 includes a data host element with the Monster.com host brand 7465. The
26 advertisement confidence effect may be even more significant with widely recognized
27 data hosts. Furthermore, the effect may also increase advertisement click-throughs with
28 web users who have already registered with the data host.

29 **[00442]** Fig. 74D illustrates a dual base data entry template 7470, wherein both
30 advertisements are associated with a single sponsor. This is simply a non-limiting

1 example, multiple advertisement templates do not necessarily have to be associated with
2 the same sponsor. Furthermore, the example 7470 may be configured to incorporate
3 additional advertisements. Fig. 74E illustrates a text-based template 7480. Also, the
4 template 7480 is configured with dual key terms 7485, as well as a user redirection
5 element 7487 (although the other templates do not explicitly illustrate the redirection
6 element, it is to be understood that the key terms or another template element may be
7 configured to effectuate directing a web user to a landing page or the base data entry
8 within the data host if a web user indicates they would like additional information).

9 **[00443]** Once the advertisement template has been retrieved, the system retrieves
10 and processes the base data entry. Fig. 74A illustrates aspects of the base data entry
11 parameter extraction process. The system retrieves the data extraction rules that are
12 associated with the selected base data entry in step 7500. The data extraction rules
13 detail the parsing process for deriving the elements used to populate the retrieved
14 advertisement template. Depending on the implementation, the extraction rules may be
15 established by the user 7505, based a set of system established extraction rules 7510, or
16 a user-selected variant of the system established rules. The user (advertisement
17 sponsor) may interact with the system to create an example advertisement, that the
18 system will recreate when an advertisement request selects that sponsor's base data
19 entry for distribution as an advertisement.

20 **[00444]** Alternately, the sponsor may simply correlate base data entry elements
21 with template elements. For example, in a sponsor-driven advertisement generation
22 process, the system may be configured with a user-friendly advertisement creation
23 module. In one implementation, the sponsor may be presented with an interactive
24 image of the base data entry next to the advertisement template. In this embodiment,
25 the sponsor may click, drop and drag base data entry elements into the corresponding
26 element fields in the advertisement template.

27 **[00445]** As illustrated in Fig. 75B, the sponsor may click and drag a “job title” 7555
28 element into the “key term” template field. The sponsor may select or modify base data
29 entry elements. If the sponsor wants to establish a “qualification” element field in a
30 template, the sponsor may select the whole qualification element from the base data
31 entry. Alternately, the sponsor may be provided with an editing interface and given the

1 opportunity to edit the element before incorporating it into the template. As illustrated
2 in Fig. 75A, the sponsor may establish the advertisement title 7520, the advertisement
3 logo 7522, and select the additional terms for incorporation into the template 7524. In
4 an implementation, the sponsor may designate base data entry elements for
5 incorporation into the advertisement based on available space (i.e., the sponsor can rank
6 whether they want to include a salary element 7560 before a job location element 7565
7 based on size/character constraints associated with a particular template).

8 **[00446]** The system may also be configured to implement a system driven
9 extraction rule set. The system driven rule set is configured as an input into an
10 automated base data entry element extraction process. Advantageously, the system may
11 record and correlate different sets of generation rules with performance and efficacy
12 metrics.

13 **[00447]** However, with significant resources to draw upon, the system driven
14 generation rule set may provide a higher level of advertisement efficacy. In step 7530,
15 the system retrieves a system default generation rule set selected for the particular base
16 data entry. In step 7532, the system parses the base data entry and extracts the data
17 elements according to the rule set. For example, the system generation rule set may be
18 configured to extract a job title 7555, starting salary 7560, and two qualification
19 requirements 7575 from the base data entry illustrated in Fig. 75B. In step 7534, the
20 system uses the extracted elements to populate the advertisement template. The
21 populated template is transferred to an advertisement distribution module, in step
22 7540.

23 **[00448]** Fig. 75C illustrates several examples of populated advertisement
24 templates. Advertisement 7580 is an example of a text-based advertisement with a job
25 title incorporated as a key term. The advertisement also includes a brief description
26 element and a job location element. The advertisement 7580 also incorporates a mouse-
27 over resource element 7585. When a web user moves the mouse pointer over the
28 advertisement, the distributed advertisement spawns another pop-up window
29 displaying additional job information or information about the advertisement sponsor.
30 As such, the web user is able to ascertain additional detail about the distributed
31 advertisement without leaving the initial advertisement display web page.

1 Advertisement 7590 incorporates both a host data entity element (e.g., Monster.com),
2 as well as an image-based advertisement into the base advertisement template.

3 **[00449]** Example advertisement 7595 illustrates a multiple advertisement, single
4 sponsor implementation. More specifically, advertisement 7595 includes two sub-
5 advertisements: one for a software engineer and one for a network administrator. Also,
6 the advertisement 7595 includes a host data entity element and a separate redirection
7 element. Certain sponsors may be interested in using the independent redirection
8 element in order to redirect a web user to additional base data entries that they sponsor
9 not included in the distributed advertisement.

10 **[00450]** Fig. 76A illustrates aspects of a landing page generation process. The
11 system generates a landing page (landing page), when a web user clicks on a distributed
12 advertisement, thereby indicating they would like additional information about the
13 opportunity. In step 7600, the system receives an advertisement interaction indicator.
14 This is generated and transmitted by the distributed advertisement and effectively
15 signals the system to send additional information about the advertised good, service or
16 opportunity. Upon receipt of the indicator in step 7600, the system identifies the base
17 data entry corresponding to the distributed advertisement in step 7610.

18 **[00451]** This type of advertisement is also extremely useful in the context of the
19 advertisement evolution data analysis process. Two advertisements may be created
20 using different generation rule sets based on the same base data entry. Advantageously,
21 the system may track and analyze the distributed advertisements' respective
22 performance based on being displayed to the same web user, in the same content
23 context.

24 **[00452]** The system processes the record entry associated with the base data entry
25 to determine if a landing page generation template has been selected by a user. If a
26 template has been selected, the system retrieves the template, otherwise the system
27 implements a system-default landing generation template in step 7620. The system
28 then populates the landing page template with data from the base data entry in
29 accordance with sponsor designated landing page generation rules. This process is

1 similar to the advertisement template population process described above (e.g., it may
2 be user-driven or system driven).

3 **[00453]** In step 7640, the populated landing page is transferred to a distribution
4 module for transmission back to the same web user IP address as the distributed
5 advertisement. In some implementations, the destination information is included in the
6 interaction indicator and consequently step 7640 may be omitted. In step 7650, the
7 system creates a landing page interaction tracking record. This is used by the system to
8 analyze the efficacy of distributed landing pages and derive performance metrics. The
9 distributed landing page may also transmit web user interaction data back to the system
10 as part of this analysis.

11 **[00454]** There are a variety of templates available depending on the sponsor's
12 needs. The templates may be configured to provide a copy of the base data entry as the
13 landing page. Alternately, a premium template may incorporate the base data entry into
14 a distributed page that also includes base data entry or data host entity related
15 functionality. An example of the premium template 7660 is illustrated in Fig. 76B
16 within the job application context. As shown, landing page 7660 incorporates data from
17 the base data entry 7665, but it also may incorporate sponsor information 7670 and/or
18 information or functionality associated with the base data entry host entity 7675 (e.g.,
19 searching for similar jobs on Monster.com).

20 **[00455]** Fig. 77 illustrates a sponsor management dashboard 7750 that a sponsor
21 may user to manage various aspects of the advertisement generation process. The
22 management dashboard is an effective tool to help a sponsor coordinate and keep track
23 of advertisement campaigns 7760, 7763, and 7766. The dashboard may be configured to
24 display an advertisement ID 7770, and the date it was distributed 7780 or created. As
25 illustrated in the Fig. 77, the dashboard displays generation metrics associated with the
26 advertisements 7775.

27 **[00456]** In the implementation illustrated in Fig. 77, the generation metrics include
28 a sponsor advertisement level 7773 (which may be configured as scaled number
29 indicating the user determined type of template/content associated with the
30 advertisement). In this implementation, the dash board also indicates whether a

1 landing page 7776 has been associated with the advertisement, as well as whether the
2 landing page includes premium functionality 7779. The sponsor is also able to see what
3 the current key term element is from the base data entry 7782 (e.g., whether the
4 advertisement will emphasize a job title, salary, location or other element as the
5 advertisement key word).

6 **Advertisement Targeting/Distribution**

7 **[00457]** The present disclosure includes a discussion of systems, methods, and
8 apparatuses for an advertisement targeting/distribution engine (hereafter "Engine").
9 The Engine may be configured to receive an advertisement request message from a
10 content provider, process a series of distribution parameters, select an underlying base
11 data entry targeted for distribution, and generate an advertisement generated from the
12 underlying base data entry. In one implementation, the Engine processes distribution
13 parameters from a variety of sources, such as the advertisement sponsor, an
14 advertisement affiliate/content provider, a web user, distribution parameters associated
15 with the underlying base data entry, and/or historical distribution parameters.

16 **[00458]** It is to be understood that, while the system may be described herein
17 primarily in the context of online advertisements (hereafter "Ads"), the system provides
18 an administrator with significant flexibility and freedom to configure the system for any
19 other number of information dissemination applications embodied in a wide array of
20 media, including print, World Wide Web, television and radio, signs and billboards,
21 product placement, postal and e-mail communications, and/or the like. Furthermore,
22 although the system may be described herein processes distribution parameters from a
23 variety of a sources, it is to be understood that, depending on the needs, parameters,
24 specifications, etc. of a particular implementation, the system may be scaled and
25 configured to process distribution parameters from a single source or any number of
26 combinations of sources.

27 **[00459]** Fig. 78 illustrates an implementation of the system providing an overview
28 of various entities that may interact with the Engine at various points during system
29 utilization. According to an implementation, the Career Advertisement Network

1 ("CAN") system 7800 is a central element that may be configured to include the Engine
2 and facilitate aspects of the functionality described herein. In the illustrated
3 implementation, the CAN system 7800 communicates with the CAN system database
4 7805. As will be discussed in greater detail below, the CAN 7800 processes distribution
5 parameters from a variety of sources in order to select an underlying base data entry and
6 generate a distribution advertisement based on the underlying base data entry.

7 **[00460]** As illustrated in Fig. 78, one of the sources may be a base data entry
8 sponsor 7810. For example, if the advertisement is generated from a base data entry
9 configured as a Monster.com job listing, the sponsor is the entity that sponsors the job
10 listing, whereas, the data entry host 7815 may be analogized with Monster.com. The
11 host entity manages underlying base data entries used to create the distributed
12 advertisements in host base data entry DB 7820 (in some embodiments, the host data
13 entity may also provide parameters that are used to facilitate selection of the underlying
14 base data entry). Although illustrated as independent elements in Fig. 78, in some
15 embodiments, the functionality associated with the host entity 7815 and the CAN system
16 7800 may be incorporated into a single system. It is also to be understood that system
17 facilitates a significant flexibility and other modifications of the various embodiments
18 discussed herein are also possible.

19 **[00461]** In an implementation, distribution advertisements are generated based on
20 an ad request message from content provider 7825 that is transmitted to the CAN
21 system as the content provider 7825 prepares content (e.g., a web page) from content
22 provider database 7830 for display to a web user 7835. The CAN system 7800 provides
23 a significant flexibility and scalability to meet the various needs of a number of data
24 entry sponsors 7810. Accordingly, a system administrator 7840 can access and
25 configure the CAN system to provide a variety of functionality customized for individual
26 sponsors, as described below. For example, different CAN system 7800
27 implementations can be customized to implement a basic or premium
28 targeting/distribution advertisement engine. One difference between basic or premium
29 sponsor subscriptions involves the scope of data processing capabilities of the CAN
30 7800. In a premium subscription, the CAN 7800 may be configured to target and
31 distribute advertisements at an extremely granular level of detail.

1 **[00462]** As discussed, the system provides a significant flexibility and a system
2 administrator can select from a variety of system functionality based on the needs of
3 various base data entry sponsors and/or content providers. The following discussion is
4 provided within the context of a sponsor providing a job listing as the base data entry,
5 with an online job placement system such as Monster.com as the data entry host.
6 However, it is to be understood that the system and functionality described herein may
7 be configured to facilitate any number of implementations and/or applications. For
8 example, the system may be configured to facilitate distribution of advertisements for
9 financial products, travel services, real estate properties, classified advertisements,
10 online auction entries and/or any other types of goods, services, or opportunities.

11 **[00463]** Fig. 79 is a high-level diagram illustrating aspects of the advertisement
12 targeting process according to an implementation of the system. More specifically, the
13 advertisement targeting process involves the process by which the CAN system 7800
14 selects an underlying base data entry used to create a targeted advertisement for
15 distribution. The advertisement is 'targeted' in the sense that the underlying base data
16 entry is selected based on a number of distribution parameters processed by the CAN
17 system. In Fig. 79, the sponsor can interact with the CAN 7800 to establish a variety of
18 sponsor established distribution parameters. In an implementation, the sponsor
19 distribution parameters 7900 relate to a variety of variables established during sponsor
20 registration or campaign creation processes. The sponsor distribution parameters often
21 relate to parameters that influence the selection of an underlying base data entry for
22 distribution to either a particular web user and/or sponsor requested content provider
23 characteristics.

24 **[00464]** The sponsor may work with a system administrator (or in some
25 implementations a self-guided setup wizard) during the sponsor registration process to
26 configure the CAN system to designate certain web users, categories of web users and/or
27 content providers as requested targets. For example, a sponsor that has a number of job
28 listings for entry level software engineering positions may request that the CAN system
29 distribute advertisements based on the sponsor's underlying base data entries (BDEs) to
30 a certain target distribution point. In this example, the sponsor may attempt to target
31 individual web users that are accessing computer software related content providers.

1 **[00465]** Further, the sponsors may request that the distributed advertisements are
2 displayed to web users who have been identified as being between the ages 18-24 or
3 around the age of someone currently in college studying computer technologies or
4 starting a career in a computer related industry. The CAN system 7800 may also be
5 configured to work with the sponsor to categorize a sponsor's underlying base data
6 entries based on BDE content, distributed advertisement specs, and/or any number of
7 sponsor distribution parameters. For example, the sponsor may provide a variety of job
8 listing descriptor tags that are used by the system during the underlying BDE selection
9 process.

10 **[00466]** As illustrated, the sponsor may pursue various level of scope for job listing
11 descriptors 7905 (the varying level of scope may correspond to the sponsor's
12 subscription type). For example, the sponsor may describe their job listings as falling
13 within broad groups like engineering jobs, computer science or programming jobs. In
14 some implementations, the sponsor may configure the distribution parameters to
15 achieve a greater level of granularity. In such implementations, the sponsor may
16 supplement the broad descriptors with more specific descriptors such as java
17 programmer, applet programmer or other more specific key words that describe the job
18 listing. It is to be understood that a variety of descriptors may be implemented and it is
19 not limited to actual job type characteristics. For example, the sponsor descriptors may
20 be based on salary range, sponsor identity, full-time positions, or other job descriptors.
21 Aspects of the sponsor distribution parameters are discussed in greater detail below
22 with regard to the Figs. 81A-81C and the sponsor targeting/distribution registration
23 process.

24 **[00467]** Additional distribution parameters may be incorporated within the
25 content provider's ad request message 7910. For example, in an implementation of the
26 system, the ad request message may be configured with content provider distribution
27 parameters and/or web user distribution parameters. The content provider distribution
28 parameters relate to aspects of the distribution process centralized around the content
29 provider.

30 **[00468]** In an implementation of the system, the content provider data 7915 may
31 include content provider data descriptors 7920. For example, the content provider may

1 include a descriptor of the various types of content they provide. In some
2 implementations, the descriptors may also vary in terms of the breadth of the
3 descriptor. As illustrated in Fig. 79, a content provider 7915 may provide varying
4 descriptors of their content 7920. In the illustrated example, the content descriptors
5 7920 range from very broad to become more focused, example descriptors may include:
6 technology in general, technology news, personal computer news, operating systems for
7 PCs, particular flavors of OS, etc.... By varying the descriptor breadth, a sponsor may
8 select a level of scope that meets their needs and is cost effective (i.e., the system may be
9 configured to provide greater levels of targeting granularity as premium services).

10 **[00469]** In some implementations, the content provider may be a system affiliate.
11 Affiliates or partners of the CAN system may provide data from a historical user
12 databases to the CAN system. In an implementation, partnerships may be formed to
13 exchange historical user data for a portion of the distributed advertisement revenue. As
14 such, the strategic partnerships may lead to a great deal of web user information. As
15 part of the registration process, web users who register with an affiliate partner may
16 agree to allow the affiliate to share certain web user data with partners.

17 **[00470]** Fig. 79 illustrates aspects of two examples of processing web user data
18 7925. A first type of web user data may be provided based on a web user's registration
19 with an affiliate web site and include a variety of web user characteristics such as some
20 types of demographic information. A second type of web user data may be maintained
21 by the affiliate web site involves web user/affiliate historical interaction. For example, a
22 web user may authorize an affiliate to maintain records of the web user's surfing/search
23 history within the affiliate web site, in addition to a variety of other interaction
24 characteristics. By way of example only, if the affiliate site is Monster.com, a registered
25 web user may authorize Monster.com to maintain data records regarding whether the
26 registered user has uploaded a resume; what type of job the registered user has searched
27 for within Monster.com; the types of job listings the registered web user has viewed or
28 applied for, or any other number of affiliate/web user interactions.

29 **[00471]** The system may also incorporate distribution parameters that are
30 determined specified by the base data entry host. In an implementation where the CAN
31 system is incorporated with the base data entry host, the base data entry host manages

1 several distribution parameters 7935 related to attempting to balance selecting the most
2 relevant underlying base data entry, while also satisfying distribution commitments for
3 the various BDE sponsors. The base data entry host distribution parameters 7935 may
4 include distribution selection probabilities that are assigned to underlying BDEs.

5 **[00472]** In some implementations, various sponsors may establish BDE
6 distribution advertisement campaigns or subscriptions. For example, a sponsor may
7 agree to pay a certain fee in exchange for assurances one or more underlying BDEs will
8 be distributed a certain number of times, over the course of a certain time period, and in
9 some instances to a certain categories of web users. As such, the data entry host
10 parameters may include distribution selection weighting parameters. These data entry
11 host parameters may be called sponsor distribution parameters. As will be described in
12 greater detail below, in some implementations each BDE may be assigned a
13 corresponding weighted selection probability. These weighted probabilities may be
14 used by the CAN system in creating the selection pool of BDEs for distribution or
15 making the final selection of the BDE for distribution as an advertisement (as discussed
16 for example with regard to Fig. 84B).

17 **[00473]** The probabilities may be used to make distribution more or less likely in
18 order to meet certain sponsor designated distribution goals. In some implementations,
19 a sponsor may pay additional fees for a premium distribution campaign, wherein the
20 selection probabilities associated with the sponsor's BDEs are all increased by a certain
21 factor. The data host distribution parameters are established to correlate BDE selection
22 categories (e.g., computer programming BDEs) with the distribution parameters
23 received from the content provider (e.g., the type of content provided and/or any type of
24 web user characteristics the provider may have). Once the correlations are executed, the
25 selection pool of BDEs may be created for potentially relevant BDEs that the system may
26 select for distribution. The CAN system selects an underlying BDE 7945 and transfers
27 the BDE to a CAN system ad creation module 7950.

28 **[00474]** As discussed above, the BDE selection targeting is based on data provided
29 by the content providers, in combination with sponsor and/or BDE host data. In some
30 implementations, content providers may agree to provide additional web user data and
31 become partners/affiliates of a the CAN system. Fig. 80 illustrates an affiliate

1 registration process 8000. As part of the registration process, an affiliate establishes a
2 series of content provider distribution characteristics 8005, which can be broken down
3 into four types according to one implementation of the system. The first establishes a
4 type of cookie 8020 that can be included with a distributed advertisement, as well as the
5 types of cookie data that can be processed by the system. Depending on the
6 implementation, the affiliate may distribute data host 8025 (e.g., Monster.com) cookies,
7 as well as affiliate accessible cookies 8030. At this point, the affiliate establishes the
8 types of data that will be collected by affiliate cookies and transferred to the CAN system
9 as part of the partnership.

10 **[00475]** Depending on the implementation, various affiliates may be configured
11 uniquely positioned to collect specialized web user data 8010. For example, affiliates
12 configured as a employment placement websites may have databases with resume data
13 associated with a particular web user or job search related data. Another example is an
14 affiliate configured as a social networking web site. The social networking web site may
15 utilize surveys to collect web user characteristic data that provides perspective on the
16 likes or dislikes of a particular web user. These specialized data characteristics assist in
17 selecting particularly relevant underlying BDEs for distribution to a web user.

18 **[00476]** Another type of content provider distribution parameter relates to
19 characteristics associated with the provider's content 8015. For example, the
20 distribution parameters may include content descriptors 8035 (e.g., technology;
21 technology news; personal computer news; etc...). Also, the content provider may
22 provide affiliate web site characteristics 8040 such as values of average/peak affiliate
23 network traffic volume or other affiliate web site characteristics.

24 **[00477]** Another aspect of the affiliate registration process involves transferring
25 available historical affiliate-web user access characteristics 8045. For example, web
26 users may allow an affiliate to participate in transferring historical web user
27 search/surfing data to help develop the CAN system database. This type of data may
28 relate to previous affiliate-web user access/interaction characteristics (e.g., searching
29 the affiliate web site for data related to computer programming jobs) 8050. Alternately,
30 the affiliate access characteristics may define what types of distributed advertisements
31 are supported by the content provider/affiliate 8055. The various distribution

1 parameters are established and associated with an affiliate/content provider data record
2 within the CAN system database 8075. Accordingly, in an implementation, the CAN
3 system can easily determine which types of data parameters are included with ad
4 request from a particular affiliate/content provider by accessing a data record associated
5 with a content provider ID. Establishing the content provider distribution parameters
6 may be only the first step toward configuring the Engine.

7 **[00478]** Fig. 81A illustrates a flow diagram associated with the sponsor registration
8 process. A base data entry (BDE) sponsor starts the registration process with a CAN
9 system interface or by working with a CAN system administrator host 8100. The system
10 creates a sponsor management record 8105. The sponsor management record includes
11 a listing of sponsor identified BDEs 8110, as well as descriptors related to categorizing a
12 sponsor's BDEs. As discussed above, BDEs may be categorized through a variety of BDE
13 data characteristics,(e.g., job type with various scaling - computer programming, java
14 programming, etc..., other characteristics may include salary, income, job requirements
15 or other descriptive job characteristics). Depending on the implementation, the BDE
16 categorization processing may be sponsor driven 8117 (e.g., with the sponsor providing
17 various groupings or tags for BDEs) or system driven 8118 (e.g., the CAN system parses
18 a BDE and generates tags or uploads existing BDE tags from a data entry host entity).

19 **[00479]** Another step involves establishing the sponsor subscription parameters
20 (which may then be used to correlate one or more advertisement distribution
21 parameters). In an implementation, the sponsor distribution parameters include BDE
22 subscription/campaign parameters 8120. For example, a sponsor is able to designate
23 certain BDEs that are hosted by the data entry host for selection and distribution as
24 advertisements. Sponsors may agree to pay a fee in exchange for assurances that their
25 BDEs will be distributed (and/or displayed) a certain number of times, over a certain
26 period of time. In some embodiments, a sponsor may also establish distribution
27 parameters that affect the selection of underlying BDEs for distribution to target web
28 users.

29 **[00480]** In order to establish subscription parameters, a sponsor may interact with
30 the CAN system (or a system administrator) to view registered affiliate reports 8133 (as
31 illustrated in Fig. 81B), visitor data reports 8130 (as illustrated in Fig. 81C) and/or

1 affiliate web site maps/content characteristics 8136. Advantageously, the sponsor
2 independently or with the assistance of a system administrator may designate certain
3 affiliates as distribution targets for their selected BDEs. In an implementation, the
4 designation process may be supplemented by historical CAN system data analysis.

5 **[00481]** The sponsor may also establish an ad distribution scope 8126. More
6 specifically, the sponsor can determine how they want to prioritize the level of
7 granularity associated with the target affiliate/content providers. For example, a
8 sponsor may designate a system defined broad collection of computer programming
9 web site content providers/affiliates as for a course ad distribution 8140. Alternately,
10 the sponsor may select a specific Java programming news web site as distribution
11 targets 8145. In a premium campaign, the sponsor may elect to pursue distribution at a
12 granular level. By way of example only, after viewing affiliate data reports, the sponsor
13 may designate whether they want coarse 8140 or granular content provider distribution.
14 Also, by way of example only, the sponsor can indicate they want to distribute their
15 advertisements to a web site that is a forum or weblog for discussions about
16 programming for linux operating systems 8145, as opposed to general computer
17 programming discussions.

18 **[00482]** Similarly, the sponsor may designate a demographic distribution scope
19 8129 of a target demographic. In order to try to fill a software engineering job position,
20 a sponsor may elect to fulfill a certain number of distributions/impressions directed to
21 granular targets 8145 instead of coarse targets 8140 (e.g., distributing advertisements to
22 affiliates that have identified a web user whose registration data indicates they have a
23 Master's degree in Computer Science 8145, as opposed to an affiliate identifying a web
24 user as someone who accessed an affiliate web page discussing general personal
25 computer peripherals 8140.

26 **[00483]** Fig. 81B illustrates aspects of affiliate web site data reports that a sponsor
27 may view in determining sponsor distribution/subscription parameters according to an
28 implementation of the CAN system. By way of example only, the sponsor may view data
29 reports about classes of affiliates and/or content providers 8163, as well as
30 characteristics associated with individual affiliates/content providers. Some of the
31 individual affiliate/content provider characteristics may include various visitor metrics

1 8165, such as the number of registered users 8167, total/average web site traffic for
2 registered/anonymous users broken down by a periodic duration (e.g.,
3 monthly/weekly/daily/hourly visitor traffic data) 8169; a registered user retention rate
4 (e.g., how many times a particular registered user returns over a given period of time)
5 8171; and/or the rate of new registrations for a given period of time 8173.

6 **[00484]** Additional affiliate/content provider characteristics may include types of
7 cataloged historical visitor data 8175, as well as individual visitor data records (as
8 illustrated and discussed with regard to Fig. 81C). The affiliate report may include
9 examples of the types of visitor registration data 8177; affiliate specific information
10 derived with the web user's authorization from web user/affiliate interaction such as
11 resume/employment history data 8179; web user survey data 8181; a web user's content
12 provider searching history 8183; and/or a content provider page access history 8185.

13 **[00485]** In some implementations, the affiliate/content provider may provide a
14 content distribution map 8187 providing an overview of the various types/categories of
15 content associated with their site, as well as how the web site is set up. These maps may
16 be useful to sponsors who are attempting to determine the sponsor distribution scope
17 8126. A sponsor may visualize how many clicks a web user would need to execute to
18 reach the content that the sponsor is interested in. For example, a sponsor who wants to
19 place software engineering job advertisements may determine that web user takes may
20 access the software engineering discussion forum directly from the content provider's
21 home page. In some implementations, the content distribution maps may be
22 complemented with traffic distribution maps that illustrate average, total periodic,
23 and/or peak traffic flows across the content provider's web site 8188.

24 **[00486]** Another example, of an affiliate distribution parameter relates to the
25 content provider indicators that specify the types of advertisements that the content
26 provider/affiliate is capable of supporting 8189. For example, a content provider may
27 indicate that they can accommodate pop-up, pop-overs and web banner advertisements.
28 The content provider may also provide display/formatting specifications as part of
29 establishing the content provider distribution parameters.

1 **[00487]** Fig. 81C illustrates aspects of an affiliate visitor report 8190. These types
2 of reports may be based on data of anonymous web users who have accessed a content
3 provider. Alternately, the report may be based on data that web users have authorized
4 content provider/affiliate 8193 to collect and transfer to the CAN system related to web
5 user interaction data. Accordingly, the system enables a sponsor 8191 to access the
6 reports 8190 through advertisement server 8192. As such, the sponsor 8191 may make
7 distribution decisions based on various types of web user characteristic data 8194a-n
8 during the base data entry selection/targeting process. In some implementations,
9 sponsors 8191 (e.g., employers with job listings) can review various categories of content
10 provider's web user characteristics and select a group of web users to designate as target
11 web users. Accordingly, if the web user is identified, the web user. For example, a
12 sponsor searches a content provider's visitor data may be achieved using keywords and
13 a search engine. As shown in the figure, site visitor data 8194a-n may be stored by the
14 affiliate content server 8192. However, the amount of information known about the
15 various site visitors is not uniform, as represented by the different sizes of the site visitor
16 data files. So, for example, more information is known about site visitor 8194A than is
17 known about site visitor 8194F. As such, the sponsor's established distribution
18 demographic scope becomes an important aspect of the targeting process that may be
19 customized to meet the needs of the sponsor.

20 **[00488]** With this collection of information, a sponsor can opt to display ads only
21 to site visitors with a certain level of known characteristic or with particular known
22 characteristic. For example, an advertiser might only want his ads shown to people
23 working in information technology and living in or around Austin, Texas, or only
24 displayed to web users with known levels of education, such as college graduates or
25 PhDs. In some implementations, the CAN system may be configured to price
26 distribution ads based upon the level of details known about a particular web site visitor.
27 For example, a sponsor might buy fewer, more expensive, narrowly targeted ads;
28 alternately, the sponsor could buy a larger number of cheaper less targeted ads.

29 **[00489]** In another implementation of the system, the affiliate content provider
30 might choose the amount of site visitor information they would like to provide to the
31 advertisement server. In addition to allowing web users to authorize data

1 collection/transfer, this method allows a content provider to address any web user's
2 privacy concerns. Another aspect of system involves providing compensation (e.g., a
3 portion of revenue generated by the sales of distribution advertisements) provided to
4 the affiliate/content provider. In an implementation, the portion of revenue may be
5 based upon the amount of information about the site visitor provided. In this way,
6 affiliate content providers can be rewarded by providing more information, which
7 results in more effective ad selection and placement. If an affiliate content provider
8 would like to provide reduced information about their site visitors, they can receive
9 broader less profitable ads.

10 **[00490]** In an implementation, the affiliate may create a web user record as a user
11 interacts with the affiliate/content provider web site. The data record may be created at
12 the request of the web user during active web user interaction 8205 or passive web user
13 interaction 8225 (e.g., web site records an anonymous non-registered user's search
14 history and identifies future interaction by placing a cookie on the user's terminal). In
15 an active interaction implementation, the web user may actively provide user-
16 identifying characteristic data during a web site registration process 8210.

17 **[00491]** Depending on the implementation, the affiliate may limit the availability
18 of certain user data (e.g., associating only zip code, gender, general age group
19 information with an anonymous user id, while maintaining the user's name, mailing
20 address in strict confidence). The system may also be configured to create a web user
21 data record during web user interaction. For example, a web site may be configured to
22 utilize user information extracted during user surveys, or during processing uploaded
23 resumes or other types of user provided data characteristics. In some implementations,
24 the web user may be requested to approve the collection and distribution of the collected
25 data to the CAN system. The Affiliate may upload user data to the CAN after identifying
26 a registered (or non-registered anonymous user) 8220. Depending on the
27 implementation, the data may be uploaded as part of an ad request or at certain
28 intervals after a user visits (or re-visits) the affiliate web site.

29 **[00492]** Some implementations may be configured to facilitate passive interaction
30 data collection and distribution 8225. Two of the many ways passive data collection and
31 processing may be effectuated through the distribution of cookies 8230. Affiliates may

1 agree to distribute cookies so that if and when a web user with a particular type of
2 content provider cookie accesses the base data entry host 8245 (e.g., a registered user
3 for a new web site visits a job employment site, the job site collects affiliate/web user
4 interaction data 8245). Another example involves collecting web user/content provider
5 interaction data based on a cookie placed on the web user's terminal each time the web
6 user accesses the content provider (e.g., as the web user conducts searches on a content
7 provider, the content provider collects interaction data).

8 **[00493]** The active/passive interaction data 8220, 8245, 8250 is then transmitted
9 to the CAN system for aggregation into a CAN system database. In the event that the
10 user is a non-registered user, the system may be configured to determine whether the
11 non-registered user data matches any stored anonymous user data records 8260. The
12 CAN system then processes and manages the user data associated with both registered
13 and anonymous non-registered users 8265. Depending on the implementation and the
14 scope of the collected user data, the system may be configured to coordinate a variety of
15 a data management tasks, such as grouping similar user data records together and/or
16 creating varying levels of group descriptors (e.g., technology characteristics, computer
17 characteristics, computer programming characteristics, java programming
18 characteristics, etc...) 8270.

19 **[00494]** Fig. 82B illustrates an embodiment for the placement of ads through the
20 interaction of the advertisement server 8272 and an affiliate content server 8274. In
21 this embodiment, one or more affiliate content servers 8274 request ad serving from the
22 advertisement server 8272. The request may include stored or categorical information
23 about the site visitor 8280 from the visitor data store 8273. The information about the
24 site visitor 8280 could have come from a number of sources.

25 **[00495]** For example, the site visitor information could have been derived during
26 the current web content session/interaction between the affiliate content server and the
27 site visitor, such as information derived from the site visitors viewing and interaction
28 with affiliate content pages 8276a and 8276b. Or, it could come from information
29 received and stored during the site visitor's previous interactions with the affiliate
30 content server. In addition, two or more different affiliate content servers might share
31 or combine their saved site visitor data.

1 **[00496]** Site visitor data might also have been previously stored by the
2 advertisement server. For example, the site visitor data might be associated with a site
3 visitor identifier, e.g., a cookie, the advertisement server might keep data indicating
4 visitor characteristics associated with the identifier. In this way, the advertisement
5 server can correlate information about the site visitor that was previously provided by
6 one or more affiliate content servers and the advertisement server can track which ads
7 were of interest to the site visitor. For example, if the site visitor had previously clicked
8 on ads for java programmers, the advertisement server will have a highly relevant data
9 point to indicate that programming jobs are of interest to that visitor.

10 **[00497]** In one embodiment, the site visitor's use of the affiliate content server
11 service results in information about the site visitor being provided to the affiliate
12 content server. For example, the affiliate content server may provided personality tests,
13 skills evaluations, I.Q. tests, career recommendations, etc., such as the tests provided at
14 tickle.com. When the site visitor uses the affiliate content server, information about
15 himself or herself is gathered. In another example, the site visitor may have filled out a
16 user profile that provides detailed information about him or herself.

17 **[00498]** In another example, the affiliate content server may provide social or
18 business networking services, such as myspace.com or linkedin.com. In using such a
19 service, the site visitor may create a web page providing details about the users skills
20 and interests. In another example, the affiliate content server may provide career
21 services like monster.com. In using the service the site visitor may have uploaded a
22 resume or created a user profile that indicates the user's skill set, educational
23 background, work experience, interests, etc. In this way, it can be seen that the affiliate
24 content server and the listings server may be one in the same entity.

25 **[00499]** Regardless of the mechanism in which information about the site visitor is
26 recorded. Some or all of the recorded information is provided by the affiliate content
27 server 8274 to the advertisement server 8272. In addition, information about the site
28 visitor might be added based on context of the affiliate content site or added by
29 categorizing the affiliate content site itself. In other words, a particular affiliate content
30 site might indicate for itself that particular characteristics apply to all of its site visitors,
31 or this determination may be made by the operators of the advertisement server. For

1 example, an affiliate content site directed to semiconductor industry news might set
2 forth that all of its site visitors are involved in the semiconductor industry. Or, the
3 advertisement server might make that determination. In addition, the affiliate content
4 server operator may want to have more control over the companies and types of ads that
5 are displayed on its site and could set characteristics accordingly. For example, an
6 affiliate content site devoted to alternative energy might want to exclude advertisements
7 from oil companies.

8 **[00500]** Once the characteristics for a particular site visitor are received and
9 gathered by the advertisement server, the advertisement server can use this information
10 to retrieve an ad from the ad data store 8273 and provide an advertisement that is
11 narrowly targeted to suit the site visitor. The search for an ad targeted to the site visitor
12 in question can be accomplished by database searches over the ad data store and
13 associated matching algorithms. For example, if the affiliate content server provides the
14 advertisement server with three facts about the site visitor, the advertisement server can
15 search the ad data store for ads that most closely match the provided criteria. Any
16 discovered ad 8278 can then be inserted into content page 8276c and presented to the
17 site visitor by the affiliate content server 8274.

18 **[00501]** If multiple matching ads are discovered, multiple ads can be shown to the
19 site visitor or the ads can be narrowed down based on additional criteria, such as the
20 amount of revenue generated by the ad, the amount of time since the ad was last
21 displayed, etc. The number of ads displayed to the site visitor may be determined by the
22 number of ad positions made available by the affiliate content server. If the number of
23 relevant ads matching the site visitor is less than the number of available ads, the extra
24 ad space can be filled in with broad based ads or ads driving traffic for the system as a
25 whole. For example, ads might drive traffic to the original listing service, e.g., the job
26 listing server.

27 **[00502]** Fig. 83 illustrates an flow diagram associated with ad targeting according
28 to another implementation of the system. Web user 8340 attempts to access content
29 provider 8300. In a passive data collection implementation, the content provider 8350
30 may transfer a cookie 8303 to the web user's terminal 8340. The system may also
31 implement aspects of an active data collection implementation - e.g., the web user 8340

1 may register with the content provider 8306. After the registration, the web user may
2 navigate through various pages associated with the content provider's web site 8309. As
3 the web user navigates through the various pages, the content provider collects data
4 associated with items of interest, advertisements displayed to the web user, content
5 (and/or categories of content) provided to the web user.

6 **[00503]** As part of the user's navigation, the user may click a link for a particular
7 page (or type of content) (e.g., requesting a computer programming news front page)
8 8315. As part of creating the page and responding to the web user, the content provider
9 8350 accesses content provider databases 8370 to retrieve the requested content, the
10 content provider may then create an ad request message 8318 (described in greater
11 detail with regard to Fig. 84A) and transmit the ad request message 8321 to the CAN
12 system 8360. The CAN System 8360 receives and processes the Ad request message
13 8324 and creates a potential BDE pool 8327(described in greater detail with regard to
14 Fig. 84B). The CAN system then selects the underlying BDE from the pool 8324, creates
15 the distribution advertisement and transmits it back to the Content Provider 8327 for
16 incorporation with the web user's requested content 8330. The requested content with
17 the incorporated advertisement is then transmitted from the content provider to the
18 web user 8333, where it is displayed to the web user 8336. In some implementations,
19 the ad request message may include content destination address parameters, as such the
20 advertisement may be created by the CAN system and transmitted to the web user
21 terminal for incorporation with the requested content from the content provider.

22 **[00504]** Figure 84A illustrates aspects of an implementation of the distributed
23 advertisement request message generation process. The Content Provider receives a
24 request for certain content from a web user 8400 (e.g., the web user may have clicked
25 on a link at the Content Provider's web site). The content provider (or affiliate) may
26 retrieve their content, as well as create an ad request message that is transmitted to the
27 CAN requesting an distribution advertisement for incorporation with their content. The
28 ad request message is populated differently base on whether the requesting Content
29 Provider/Affiliate is registered with the CAN system 8410.

30 **[00505]** If the Content Provider is not registered or is implementing in a limited
31 trial version of the CAN System, the Ad creation process may initiate a registration

1 process for the Content Provider 8412. Also, in this implementation, the Content
2 Provider may be requested to provide a general data descriptor 8414 that provides a
3 high-level description of the types of content that the Content Provider provides to web
4 user. The general data descriptor is used to populate the ad request message, which is
5 then transmitted to the CAN system as a request for a default distribution ad 8416 (e.g.,
6 the CAN may determine that requesting entity is a content provider directed to new
7 broadcasting and request an underlying BDE from a related field, such as a journalism
8 job opportunity).

9 **[00506]** A registered Content Provider (or one implementing a full trial version of
10 the system) populates the ad request message with a variety of Content Provider
11 Distribution Parameters 8415. For example, a registered Content Provider may simply
12 provide a Content Provider ID and/or an ad specification ID 8420 (as described below,
13 these parameters are may be used by the CAN system to correlate various stored
14 Content Provider and/or advertisement characteristics, such as Content Provider
15 descriptors, advertisement specification formats or any other number of CAN system
16 stored content provider characteristics used during the BDE selection and/or Ad
17 creation process). The Content Provider may provide additional content category data
18 8425 (e.g., a computer news web site, may provide various data related to the general
19 type of content requested by a user, such as - Personal Computer News stories). In
20 some implementations, this data may be further supplemented by a descriptor related to
21 the specific content being requested 8430 (e.g., a specific news story about the latest
22 personal computer CPU).

23 **[00507]** Once the Content Provider distribution characteristics have been
24 populated, the Content Provider may incorporate web user data about the web user into
25 the ad request message 8435. If no web user data exists the ad request is based
26 primarily on the Content Provider distribution characteristics, as the Content Provider
27 prepares a request for the distribution ad 8440. The Content Provider may have a
28 variety of identified web user data, for example the Content Provider may have active
29 8442 or passive 8443 web user interaction data. For example, the Content Provider may
30 populate the ad request with the web id 8444 associated with a registered web user
31 collected during a registration process, or during content provider/web user interaction

1 (e.g., an id that the CAN system may use to access a variety of user characteristics that
2 have been previously uploaded to the CAN system). In another example, the Content
3 Provider may populate the ad request message with collected dynamic web user data
4 (e.g., cookie data collected from an anonymous web user during interaction with the
5 Content Provider). In other implementations, the ad request message may be
6 configured to include a wide variety of other distribution parameters that may be used
7 by the CAN in selecting the BDE for distribution as an advertisement. Once the ad
8 request message is populated, the Content Provider finalizes and transmits the Ad
9 Request message to the CAN system.

10 **[00508]** Fig. 84B illustrates aspects of the CAN system processing an ad request
11 message and selecting an underlying BDE for distribution as an online advertisement.
12 The CAN system receives the ad request message from the Content Provider/Affiliate
13 8450. The CAN system determines whether the requesting Content Provider is a
14 registered content provider 8455. If the requesting content provider is a registered
15 content provider, the CAN system extracts content provider distribution characteristics
16 8457. The CAN system may be configured to extract a Content Provider ID, which may
17 be correlated to a wide of a Content Provider distribution characteristics that are stored
18 during a registration process and/or through ongoing data updates.

19 **[00509]** For example, the Content Provider ID may be correlated with ad
20 formatting specification 8459, Content Provider content descriptors 8461 such as,
21 technology news, computer news, etc..., content descriptors related to the specific web
22 user requested content 8463 such as, a descriptor about the requested link or news
23 article. In some implementations, the CAN system may be configured to extract these or
24 other distribution parameters that the Content Provider uses to populate the ad request
25 message (instead of being stored on the CAN system and correlated to the Content
26 Provider ID). These and other distribution parameters may be used to identify a
27 number of potential BDEs that are used to create a BDE distribution pool 8469.

28 **[00510]** For example, if Content Provider descriptor 8461 indicates the content
29 provider is involved in computer programming news and that the web user requested
30 content is a link to an article discussing a new java programming technique. The BDE
31 distribution pool may be created to contain twenty-five BDEs related to computer

1 programming employment opportunities. In an implementation, the CAN system may
2 implement a scaling module, in which the distribution pool may be sub-divided into
3 granular groups of BDEs, such as Java programming opportunities, AJAX programming
4 opportunities, or other sub-groups.

5 **[00511]** If the Content Provider is not a registered affiliate 8455, the CAN system
6 may analyze the requesting entity 8465 (e.g., by retrieving content provider
7 characteristics from a CAN maintained content provider database, by extracting a
8 general content provider descriptor from the ad request message or a number of other
9 processes). For the un-registered Content Provider, the CAN system may derive a
10 general distribution pool 8467. For example, if the Content Provider does not have an
11 Content Provider ID with stored characteristics and/or was not able to populate an ad
12 request message, the CAN system may determine that the Content Provider is in a
13 computer related industry by analyzing the requesting address (e.g.,
14 www.javacomputernews.com) and create a general distribution pool that includes
15 twenty-five computer industry job opportunities.

16 **[00512]** Once the BDE distribution pool has been created, the CAN system
17 determines whether the ad request message includes web user data 8471. If the ad
18 request message does not include web user data, the CAN System selects a BDE from the
19 distribution pool base on sponsor subscription characteristics associated with the BDEs
20 in the pool. In contrast, if the CAN system extracts a web user data record 8473, the
21 system retrieves web user characteristics from the CAN system database. The CAN
22 system may analyze a user data record (if one exists) 8475, user data extracted from the
23 ad request message 8477 (web user cookie data) or some combination of the two. The
24 user data is then used to adjust the BDE pool 8479 (e.g., add, delete, or substitute BDEs
25 with BDEs from the pool). In an example, the CAN system may retrieve a data record
26 that indicates an identified web user is (or has been) a Java programmer. This
27 information may be used to delete non-java programming opportunities from the BDE
28 distribution pool. Further, the CAN system may be configured to retrieve additional
29 BDEs that related to Java programming opportunities to supplement the BDE pool.

30 **[00513]** After the contents of the BDE pool is adjusted, the CAN system may
31 conduct an initial ranking of the BDEs in the distribution pool according to a number of

1 factors 8479. For example, the content provider distribution characteristics, the web
2 user characteristics, or some combination of the two may be used to create a ranking of
3 BDEs based on their relevance to the content provider and/or the web user. Further,
4 the system may utilize BDE sponsor data to refine the initial rankings. If web user data
5 does not exist, the CAN system may derive a ranking based on the content provider's
6 and/or the sponsor's distribution parameters. In an example, the initial system derived
7 BDE ranking 8479 may be reordered to fulfill sponsor distribution specifications 8481,
8 (e.g., if a BDE in the distribution pool is designated as a premium weighted BDE, it may
9 be selected for distribution before regular subscription BDEs). In another example, if
10 the BDE pool includes a BDE that must be distributed in order to fulfill a sponsor's
11 distribution or impression quota, it may be selected for distribution over BDEs that may
12 have more relevant subject matter. These implementations facilitate balancing
13 distributing advertisements that are particularly relevant to a content provider, a web
14 user or both, as well as meeting the distribution requirements associated with a
15 particular sponsor's underlying BDEs. Once the balancing is achieved, the CAN system
16 selects the BDE and transfers it to the an Ad creation module associated with the CAN
17 system 8483.

18 **[00514]** Figure 85 shows an exemplary process and flow diagram for ad serving in
19 accordance with the disclosed system. One or more advertisers 8500 submit a request
20 for advertising, which is received by the ad server 8520a. Along with the request, the
21 advertiser may include information indicating the site visitor criteria identifying
22 preferred candidates for the advertisement and the amount the advertiser is willing to
23 pay. In the alternative, the relevant site visitor criteria can be extracted from the
24 content of the listing. The advertiser might also have specific requests. The advertiser,
25 for example, may require that the ad is only shown to site visitors in a particular
26 geographical region or with a specific education level. Or, the advertiser might simply
27 indicate that its ads should only be displayed to site visitors that match a certain number
28 of its listing criteria. For example, the advertiser might choose a level where the ad is
29 only shown to site visitors that match three or more criteria relevant to its offer. The ad
30 server can then store the advertisement request details 8520b in the ad data store 8521.

1 **[00515]** An affiliate content server 8530 will interact with the ad server to request
2 ads to be displayed in the affiliate content server's content pages 8506. The affiliate
3 content server 8530 will display its content to site visitor 8505. In interacting with the
4 site visitor, or through previous interactions with the site visitor, the system develops
5 site visitor data 8531. The site visitor data 8531 may be passed to the ad server along
6 with the request for ad serving. Upon receipt of the ad serving request 8520c, the ad
7 server searches 8520d the ad data 8521 for ads matching the request, including any
8 supplied visitor data. If multiple potentially matching ads are discovered in the search,
9 the ads with the highest price and/or the closest match to the submitted data are served
10 8520e. If only one ad is discovered, that ad is served. The ad to be served 8521 is
11 inserted into the affiliated content server's content page 8506. This can be done directly
12 by the ad server 8520 or the ad could be provided to the affiliate content server for
13 insertion into the content.

14 **[00516]** In an alternative embodiment, instead of using pre-generated ads stored
15 in the ad data store, the advertisement server may directly search the job listings data
16 store for relevant job listings. If a job listing matching the ad request is discovered, an
17 ad can be automatically generated using the ad generation techniques described above.
18 The ad can then be inserted into the content displayed to the site visitor. In such system
19 the ad data store could either be eliminated or it could be used to store details regarding
20 the advertisers' requests, budgets, criteria, and ad pricing.

21 **[00517]** Figure 86 illustrates an embodiment disclosing advertisement distribution
22 in accordance with another aspect of the disclosed system. As shown in the figure,
23 advertisement server 8620 is in communication with affiliate content servers 8630 and
24 advertisement distribution server 8625. The affiliate content server displays site
25 content pages 8606. The figure demonstrates a number of methods for distributing ads
26 8622 to the affiliate content server. In one embodiment, the advertisement is passed
27 from the advertisement server directly to the affiliate content server. It should be noted
28 that ad could either be delivered as the content of the ad itself or it can be delivered as a
29 symbolic link to the ad content. In another embodiment, the ads can be served by a
30 dedicated advertisement distribution server, which might be embodied by an existing
31 commercial ad distribution network. In this way, the advertisement server can take

1 advantage of large commercial distribution networks and have access to affiliate content
2 serves that would like to source their advertisements through such a provider.

3 **Advertisement Evolution**

4 **[00518]** The present disclosure includes a discussion of systems, methods, and
5 apparatuses for an Advertisement Evolution Engine (hereafter "Engine"). The Engine
6 may be configured to track advertisement performance and generate new
7 advertisements based on performance metrics such as user responses to and/or
8 interactions with existing advertisements. In one embodiment, the disclosed Engine is
9 configured to interact with three entities; each of which is communicatively coupled to
10 the Engine: (i) a content provider capable of transmitting and receiving data to a web
11 user, for example media and advertisement content, (ii) a web user capable of receiving
12 and interacting with displayed advertisements, as well as providing feedback and (iii) an
13 underlying database entry host for generating and/or populating advertisements.

14 **[00519]** It is to be understood that, while the system may be described herein
15 primarily in the context of web-printed advertisements (hereafter "Ads"), the system
16 provides an administrator with significant flexibility and freedom to configure the
17 system for any other number of information dissemination applications embodied in a
18 wide array of media, including print, World Wide Web, television and radio, signs and
19 billboards, product placement, postal and e-mail communications, and/or the like.

20 **[00520]** Furthermore, although the system may be described herein primarily in
21 the context of evolving Ad generation templates, it is to be understood that, depending
22 on the needs, parameters, specifications, etc. of a particular implementation, the system
23 may be configured to evolve other advertising system functionality or processes. For
24 example, a system administrator may configure iteratively optimized system modules
25 including: Ad tracking routines, Ad targeting, Ad distribution routines, resume/job
26 seeker profile generation routines, information dissemination/display routines, and/or
27 the like.

28 **[00521]** FIGURE 87 provides a conceptual illustration of evolved advertisements
29 according to an embodiment of the present invention. More specifically, the figure

1 illustrates a representation of Ad evolution analogous to the archetypal illustration of
2 organismic evolution. An Ad generation template lineage may begin at primitive and/or
3 untested "ancestral stages" 8708, and proceed through intermediate stages of
4 adaptation (8709, 8710) to a "modern" stage (8711). In the example shown in the figure,
5 the Ad evolves from a simpler form (8708) into one of greater complexity (8711) by
6 incorporating elements of the 'successful' advertisements as determined by the system.
7 The characteristic features and/or elements of an evolving Ad (8715, 8720, 8725) may
8 be selected based on their capacity to make the Ad more effective.

9 **[00522]** FIGURE 88 provides an overview of various entities that may interact with
10 the Engine at various points during system utilization. According to an embodiment of
11 the invention, the Career Advancement Network ("CAN") 8801 includes the Engine and
12 facilitates aspects of the functionality described herein. A content provider 8805
13 connects with the CAN 8801 via a Content Provider System 8810 to request an Ad 8812
14 from the CAN 8801 for incorporation with provider content from Content Provider
15 Databases 8820. The combined provider content 8815 and generated advertisement are
16 displayed together to web user 8845 on terminal 8850.

17 **[00523]** Depending on the particular system specifications, the Content Provider
18 System 8810 may connect directly with the Engine as illustrated in Fig. 88. Alternately,
19 the Content Provider System may connect with the Engine via an intermediary CAN
20 8801 module. While in yet another implementation, the Content Provider System 8810
21 connects with the Engine via an intermediary network such as the internet 8825.

22 **[00524]** The CAN 8801 retrieves Ad generation templates 8830 from an Ad
23 generation template database 8838 based in part on provider content characteristics
24 included as part of the ad request 8812. The CAN system extracts and incorporates
25 content from base data entries (BDEs) 8837 which are stored on a BDE host entity's
26 underlying data base 8836 (e.g., a database with Monster job listings). The system
27 processes extracted BDE content in accordance within an Ad Generator module 8840 in
28 accordance with a set of generation rules stored in the ad generation template DB 8838.
29 The generated ads 8835 may be sent to the content provider system 8810 for
30 incorporation with the provider content 8815 before final distribution across the
31 internet 8825 to web user 8845.

1 **[00525]** For example, the content provider 8805 may be configured as a sports
2 news web site. The content provider 8805 distributes various sports news content from
3 the content provider's database 8820. The CAN 8801 may be configured to coordinate
4 incorporating CAN generated advertisements 8835 into the content 8815 distributed by
5 the content provider's system 8810. The CAN 8801 is configured to create the
6 advertisement based on a variety of factors, some of which may include: a content
7 provider's content 8815, a content provider's advertisement system configuration, web
8 user 8845 characteristics, and/or any variety of other distribution metrics.

9 **[00526]** In one embodiment, the Ad generator selects an Ad generation template
10 based in part on parameters generated in an Ad targeting/distribution process.

11 **[00527]** Web users 8845 view distributed Ads on terminal computer systems 8850
12 and provide passive/active feedback 8855 to the CAN. In one embodiment, web users
13 8845 register responses that are relayed to a feedback evaluator module 8860 within the
14 Engine. The feedback evaluator module 8860 processes user responses 8855 and
15 generates a set of Ad scores 8865 based on the responses. The scores may then be issued
16 to an Ad evolver module 8875, which uses the scores to process Ad generation templates
17 and/or create new templates 8880 for future use. These templates are managed by the
18 Ad generation template databases 8838.

19 **[00528]** FIGURE 89 is a high-level flow diagram illustrating aspects of the
20 advertisement evolution process in one embodiment of the system. The system receives
21 provider content from content provider/system affiliate 8901, selects a BDE 8903, and
22 selects an Ad generation template from Ad generation template database (DB) 8905.
23 Extracted content from a BDE is organized into an Ad or Ads at 8910 in accordance with
24 instructions in the Ad generation template. The extracted content from a single BDE
25 may be incorporated into a variety of different Ads based on BDE organizing
26 instructions contained in various Ad generation templates.

27 **[00529]** Depending on the system implementation, Ads may be widely distributed
28 and/or targeted to particular web users 8915. After distributing the generated Ads 8915,
29 the system may receive web user feedback in the form of web user-Ad interaction and/or
30 Ad evaluations 8920. The system receives the feedback and derives a series of

1 performance metrics by analyzing web user registered responses interacting with
 2 (*passive* feedback, e.g., click-throughs, mouse-overs, etc.) and/or reacting to (*active*
 3 feedback, e.g., ratings, etc.) the Ads. The system subsequently manages ranking Ad
 4 generation templates and/or generating new Ad generation templates 8925 for
 5 populating the Ad generation template database 8930 based on derived performance
 6 metrics 8920. In managing the ad generation template database, the system is able to
 7 incorporate Ad features/elements from successful distributed Ads into future
 8 generations of distributed Ads that are expected to elicit particular user responses. The
 9 following discussion will discuss aspects of the Figure 89 flow diagram in greater detail.

10 Ad generation 8910

11 **[00530]** FIGURE 90 illustrates Ad generation in one embodiment. A content
 12 provider requests an Ad from the CAN and may supply provider content characteristics
 13 that detail a content provider's content, a content provider's advertisement system
 14 configuration, web user 8845 characteristics, and/or any variety of other distribution
 15 metrics.

16 **[00531]** This triggers the CAN to select a BDE 9001 from an underlying Monster
 17 BDE database 8838 - in this case pertaining to a Monster.com job listing. The system
 18 then extracts BDE content such as the listing's job title 9005, sponsor company 9010,
 19 location 9015, status 9020, job category 9025, required/desired experience 9030,
 20 expected career level 9035, required/desired education level 9040, salary 9045, job tag
 21 9050, and job description 9055 as BDE elements.

22 **[00532]** In one embodiment, the BDE may be supplied in an XML format with a
 23 form such as:

```
24 <Job_Listing_BDE>
25   <ID> 012345 </ID>
26   <title> Associate Director Climate Physics </title>
27   <company> National Climate Labs (NCL) </company>
28   <location> Miami, FL </location>
29   <status> Full-time, Employee </status>
30   <category> Science </category>
31   <experience> 8+ years </experience>
32   <level> Manager </level>
33   <education> Doctorate </education>
34   <salary> $200,000/yr </salary>
```

```

1      <tag> Leadership Opportunity in Climate Physics </tag>
2      <description>
3          NCL is seeking an individual with proven
4          scientific accomplishments and leadership
5          qualities to guide its Climate Physics Group.
6          This group is comprised of nationally
7          recognized atmospheric research scientists and
8          Engineers working toward measurable
9          improvements in climate change prediction.
10     </description>
11 </Job_Listing_BDE>

```

13 **[00533]** The incorporation of BDE content into an actual distributed Ad is
14 instructed by an Ad generation template 9060. Analogous to the way an organism's
15 genetic sequence determines the expression of many of its traits, the Ad generation
16 template 9060 may determine Ad format, traits, characteristics, and/or elements of an
17 advertisement. There exist a vast array of Ad characteristics that an Ad generation
18 template may control or determine, including which BDE elements are incorporated
19 into/omitted from an advertisements, as well as Advertisement format and
20 characteristics such as layout, arrangement, and/or size of BDE elements and/or other
21 Ad features, colors, fonts, multimedia content (e.g., images, animations, video, audio,
22 etc.).

23 **[00534]** The generation template may also coordinate Ad themes, background
24 designs, incorporation of and arrangement of interactive elements/widgets. Also, the
25 generation template may be configured to define the Ad size, Ad type (e.g., in an
26 internet-based Ad embodiment, the Ad type may specify pop-up, pop-under, banner,
27 hover, interstitial web page, rich media banner, e-mail solicitation, redirect, and/or the
28 like), Ad combinations, proximities of different Ads, and/or the like. In the embodiment
29 shown in Fig. 90, the template instructing the expression of extracted BDE elements in a
30 particular Ad is encoded in an XML format 9065. The resulting Ad is displayed at 9070.

31 **[00535]** A single BDE may be incorporated into a diverse array of different Ads,
32 yielding a sort of Ad biodiversity, via the employment of different Ad generation
33 templates. FIGURE 91 illustrates this principle in one embodiment, wherein a single
34 BDE 9101 (as in Fig. 90) is processed via the Ad generator 9105 module through four
35 generation templates (9105a, 9105b, 9105c, 9105d) to yield four different Ads with
36 different styles and/or designs (9110a, 9110b, 9110c, 9110d). The Ads in this

1 demonstrative embodiment, despite being built from the same BDE 9101, each
2 emphasize a different aspect of the job opportunity listing in the Ad heading (e.g., 9110a
3 emphasizes the job title, while 9110d emphasizes salary and education level
4 requirements). The different manner in which each Ad organizes and displays the BDE
5 elements 9101 may affect users differently and translate into different degrees of Ad
6 performance success.

7 **[00536]** In one embodiment, two Ads generated from the same BDE using different
8 Ad generation templates may be displayed simultaneously and/or in close proximity to
9 each other. The simultaneous close proximity display facilitates a unique opportunity to
10 obtain feedback for two distributed advertisements. More specifically, this
11 implementation involves displaying the advertisements to a web user with the identical
12 demographic information (because they are each displayed to the same web user).
13 Accordingly, the system is able to obtain premium feedback regarding how distributed
14 advertisements perform and more effectively gage which Ad (and, thus, which
15 underlying Ad generation template) is more successful.

16 **[00537]** The Engine processes user feedback correlated to existing Ad generation
17 templates to yield new Ad generation templates. In one implementation of the system,
18 an initial pool of Ad generation templates may be set up at the outset of Engine
19 operation, in order to supply a diverse set of Ad types for users to interact with and/or
20 react to. The initial pool of Ad generation templates may reflect a variety of different
21 compositions and be created by a variety of different means in various embodiments
22 within the scope of the present invention. For example, in one embodiment, the initial
23 pool of Ad generation templates may be manually created by a system administrator. In
24 another embodiment, the Engine may analyze a collection of existing Ads to extract Ad
25 generation templates. This may be accomplished, for example, by employing a variety
26 of image, OCR, and/or text recognition and processing tools. In a further embodiment,
27 a system user created generation template may be compared against system generation
28 templates.

Web User Response Registration (8855, 8920)

1
2 **[00538]** Evolutionary success from a biological perspective essentially depends on
3 an organism's (or, more fundamentally, a gene's) ability to successfully reproduce.
4 Traits that are genetically determined (i.e., passed from one generation to the next) and
5 promote reproductive success prevail in subsequent generations, while traits that inhibit
6 reproductive success dwindle and die out as the organisms who carry them fall behind
7 in sprouting offspring.

8 **[00539]** Similarly, an implementation of the system may employ various standards
9 of performance success for evaluating Ad efficacy. The system may be configured to
10 rank the Ads based on one or more performance characteristics. The ranking, in turn
11 may be used to determine the propagation of particular Ad generation templates and/or
12 Ad generation template elements in subsequent generations of Ads. Possible definitions
13 of Ad success are widely varied and may differ depending on particular goals or
14 requirements of different embodiments of the CAN. Some examples of Ad success may
15 include Ad click-through numbers or rates (e.g., Ad click-throughs per impression, Ad
16 click-throughs per day, etc.). Additional Ad performance characteristics may be based
17 on Ad user ratings, Ad consummation numbers or rates (e.g., a purchase made based on
18 an Ad click-through, a job interview/hire based on a job listing click-through,
19 application submissions, new user registrations. Also, Ad performance characteristics
20 may include any other type of post-click response that may be correlated with web user
21 interactions with an Ad), mouse-overs (such as may be detected by an Ajax and/or
22 JavaScript enabled software module), clicking on interactive Ad elements, mouse
23 pointer tracking, head and/or eyeball movement tracking, time spent on an Ad, content
24 provider requests for particular Ad generation templates, and/or the like.

25 **[00540]** In one implementation, only positive user responses are registered while
26 in another implementation, both positive and negative user responses are registered,
27 while in still another implementation, only negative user responses are registered. An
28 example of a negative user response might be closing a window containing an Ad
29 quicker than a specified minimum time period, a low and/or negative user rating,
30 and/or the like. In one embodiment, only web user interactions with and/or responses
31 to whole Ads are registered by the Engine, while in another embodiment, only web user

1 interactions with and/or responses to elements within an Ad are registered by the
2 Engine, while in yet another embodiment, web user interactions with and/or responses
3 to both the entirety of Ads and to elements within Ads are registered by the Engine.

4 **[00541]** FIGURE 92a shows an embodiment of an Ad receiving and processing
5 passive performance metric registration based on web user click-throughs. A web user
6 viewing an Ad 9201 may decide to click 9205 an interactive widget 9210 in order to
7 acquire more information. The system may include a feedback module as part of the
8 landing page generation process. The feedback module creates a landing page providing
9 the requested additional information and transmits a feedback indicator back to the
10 system. As illustrated in Fig. 92a, the feedback indicator is configured as an increment
11 9215 to the Ad generation template score 9230 stored in the system. In one
12 embodiment, the increment may initially occur to a record of the Ad generation
13 template score in a cookie 9220 stored locally on the web user's computer; the cookie
14 may then upload an accumulated score 9225 to the system at some later time.

15 **[00542]** FIGURES 92b-c show embodiments of Ads admitting active performance
16 metric registration based on web user ratings of a whole Ad (Fig. 92b) or elements
17 within Ads (Figs. 92c and 92d). In Fig. 92b, a web user may rate their impression of the
18 overall "quality" of the Ad using a radio button widget 9235. A wide variety of other
19 rating interfaces and mechanisms may be employed in different embodiments,
20 including buttons, lists, scrollbars, sliders, text boxes, checkboxes, fields, menus, icons,
21 and/or the like. In Fig. 92c, a radio button widget is provided after each Ad element
22 (job title heading 9240, clip art motif 9245, education level requirement 9250, and
23 salary 9255). Since the presence of rating widgets can, themselves, affect the
24 appearance of an Ad, it may be desirable in some embodiments to hide the rating
25 mechanism from immediate view. Fig. 92d shows an embodiment wherein the rating
26 widget appears as a pop-up 9260 only when the mouse pointer 9265 is placed over an
27 Ad element (in this case, the star motif).

28 **[00543]** Web user responses may also be registered for whole Ads or Ad elements
29 within embodiments employing any other performance metrics. Ad element ratings
30 may be considered during creation of new Ad generation templates, whereby collections
31 of highly rated Ad generation template elements are compiled to create new Ad

1 generation templates. Details surrounding creation of new Ad generation templates,
2 including based on compilations of highly rated Ad generation template elements, are
3 discussed below.

4 **[00544]** In some system implementations, the pop-up rating widget 9260 may be
5 activated on a volunteer's local terminal. The volunteer may be a web user who has
6 agreed to provide feedback to distributed advertisements and identified as a volunteer
7 by processing cookie information stored on their local system. In one embodiment, the
8 Engine may be configured to restrict registration of web user responses and/or
9 interactions for a given, unique web user (e.g., such as may be designated by a unique IP
10 address). For example, the Engine may elect to accept a limited number of web user
11 responses from a particular web user, for a particular Ad, within a particular interval of
12 time, and/or the like.

13

Feedback evaluation (8860, 8920)

14 **[00545]** In one embodiment, web user responses and/or interactions registered by
15 the Engine are directed to the Feedback Evaluator module 8860 for subsequent
16 processing. The Feedback Evaluator module may track web user responses/interactions
17 and, in turn, convert them into scores or rankings for Ads and/or Ad elements.

18 **[00546]** FIGURE 93a shows high-level logic flow for system driven analysis of
19 passive performance metrics. At 9301, Ads are widely distributed and/or targeted to
20 particular web users. At 9302, the system creates Ad feedback records corresponding to
21 the distributed Ads that may be configured to track Ad performance metrics. At 9303,
22 the system monitors web user interactions with Ads and/or collects information
23 regarding web user interactions with Ads that may be stored in corresponding Ad
24 feedback records. In one implementation, the system may install cookies on web user
25 computers that are capable of tracking and/or monitoring web user interactions with
26 Ads and relaying interaction information back to the system. At 9304, the system may
27 create a performance hierarchy of distributed Ads based on performance metrics
28 collected in Ad feedback records. At 9305, the system preferentially selects Ad
29 generation templates and/or generates new Ad generation templates based on collected
30 web user interaction information and/or the performance hierarchy created at 9304.

1 The Ad generation templates selected and/or generated at 9305 are employed at 9306
2 to create new Ads that are then re-distributed to web users.

3 **[00547]** FIGURE 93b shows detailed logic flow for system driven registration of
4 passive performance metrics in according to an embodiment of the system. Ads are
5 distributed to web users at 9307 and the system passively monitors web user
6 interactions with the Ads 9308. Thereafter, the system may receive a notice of a web
7 user interaction whenever such an interaction occurs or at regular intervals. For
8 example, the system may install a cookie on the web user's computer that collects web
9 user-ad interaction information and periodically uploads the information to the system.

10 **[00548]** Received interactions are evaluated to determine whether they are valid at
11 9309. An example of an invalid interaction may be a second click-through indication
12 associated with an Ad for which the same web user has registered a previous click-
13 through within a predetermined duration of time (e.g., the last 10 minutes). Invalid
14 interactions are disregarded at 9310, while Ad rating records are retrieved for valid
15 interactions at 9311. The valid interactions are evaluated to determine whether they are
16 positive or negative 9312 and the corresponding Ad generation template score is
17 incremented 9315 or decremented 9320.

18 **[00549]** In another embodiment, the received interactions may pertain to elements
19 of the Ad generation template rather than the Ad generation template as a whole.
20 Consequently, Ad generation template elements may be scored separately. The system
21 determines at 9325 if the present round of persistent monitoring should continue. If
22 the system determines the round is complete, the Ad generation template scores are
23 persisted 9330. For example, the system may update and save the performance data
24 associated with the Ad generation template in the Ad generation template database.

25 **[00550]** FIGURE 93c shows detailed logic flow for web user driven registration of
26 active performance metrics in one embodiment that admits evaluation of both whole
27 Ads and Ad elements. Initial processing of web user feedback is performed at 9335, and
28 a determination is made as to whether the response is valid at 9340. An example of an
29 invalid web user response may be the entry of unacceptable characters in a text-box
30 based web user rating interface. Another example of an invalid web user response may

1 be a response from the same web user for the same Ad within a predetermined duration
2 of time.

3 **[00551]** Invalid web user responses are disregarded 9345, while Ad rating records
4 are retrieved for valid web user responses at 9347. Valid responses are evaluated at
5 9350 to determine whether they are responses to a whole Ad or to Ad parts/elements.
6 In another embodiment, only whole Ad responses are allowed, while in another
7 embodiment, only Ad element responses are allowed. For a whole Ad response, a
8 determination is made as to whether a positive or negative response has been registered
9 9355, and the score for the corresponding Ad generation template is incremented 9360
10 or decremented 9365 depending on whether the response is positive or negative
11 respectively. In another embodiment, only positive Ad responses are registered while,
12 in yet another embodiment, only negative Ad responses are registered. At 9395, the Ad
13 generation template scores are persisted.

14 **[00552]** In an implementation of the system, the Engine is configured with an Ad
15 scoring management module. The scoring module manages rankings of various
16 generation templates and/or template elements, depending on the implementation. If
17 the Engine determines at 9350 that Ad elements have been evaluated, then the response
18 for a given Ad element is queried at 9370. Depending on whether the score is positive or
19 negative 9375, the score corresponding to the Ad element in question is incremented
20 9380 or decremented 9385. If there are more Ad element responses, the Engine returns
21 to 9370. If not, then the Ad element scores are persisted at 9395. In one embodiment,
22 new Ad generation template and/or Ad generation template element scores replace
23 older Ad generation template and/or Ad generation template element scores, while in
24 another embodiment, new Ad generation template and/or Ad generation template
25 element scores are appended to a historical record of Ad generation template and/or Ad
26 generation template element scores. It is to be understood that the Ad scoring
27 management module may simply update Ad scores or maintain historical scoring
28 records for whole Ad generation templates as well.

29 **[00553]** Maintaining historical records of Ad scores enables the Engine to track Ad
30 success over successive Ad generations and through various adjustments and/or
31 variations in Ad style and/or design. Furthermore, Ad success may be resolved over a

1 variety of independent variables, including time, location, Ad style and/or design, Ad
2 characteristic(s) and/or element(s), and/or the like. Success records may be evaluated
3 by Engine administrators, content providers, and/or BDE providers to determine trends
4 and/or factors affecting the success of Ads and Ad generation templates.

5 **[00554]** There are a wide variety of possible scoring system implementations for
6 Ad generation templates and/or Ad generation template elements that may be applied
7 within various embodiments of the present invention, depending on specific goals
8 and/or requirements of a specific implementation. In one embodiment, a numerical
9 value associated with particular types of web user responses may be added to or
10 subtracted from the Ad generation template and/or Ad generation template element
11 score. For example, each click-through for a given Ad may increment the corresponding
12 Ad generation template score by +1 (see, e.g., Fig. 92a). For another example, values of -
13 2, -1, 0, +1, and +2 may be associated with each of the options between "Hate it!" and
14 "Love it" in the radio button embodiment shown at 9235 in Fig. 92b. In another
15 embodiment, a minimum number of responses of a particular type must be registered
16 before an Ad generation template score is adjusted (e.g. each time an Ad click-through is
17 registered).

18 **[00555]** In another embodiment of Ad generation template and/or template
19 element scoring, a score reflecting a probability (i.e., between 0 and 1) may be assigned
20 to each Ad generation template and/or Ad generation template element and increased
21 or decreased based on web user responses and/or interactions. For example, an initial
22 score of $S = 0.5$ may be assigned to a new Ad generation template. A negative web user
23 response, then, may change the Ad generation template score to S^N , where N is some
24 positive real number greater than 1.

25 **[00556]** For example, for $N = 1.1$, a single negative web user response would
26 change the Ad generation template score from $S = 0.5$ to $S = 0.467$. A positive web user
27 response, then, may change the Ad generation template score to the N -th root of S [or,
28 equivalently, $S^{(1/N)}$]. Thus, for $N = 1.1$, a single positive web user response would
29 change the Ad generation template score from $S = 0.5$ to $S = 0.533$. In the limit of a
30 large number of negative responses, the score will approach 0 while, in the limit of a
31 large number of positive responses, the score will approach 1. The rate at which these

1 limits are approached and that template scores will vary with each response in this
2 embodiment is determined by the exponent N. A particular use for an embodiment with
3 scores reflecting probabilities will be made apparent within embodiments of Ad
4 evolution discussed in the next section.

5 **[00557]** In one embodiment, a total Ad generation template score may be derived
6 from the scores for the Ad generation template elements of which the Ad generation
7 template is composed, such as by adding up all of the composite Ad generation template
8 element scores. In another embodiment, Ad generation template element scores may be
9 assigned based on the score of the overall Ad generation template of which they are a
10 part. For example, all of the elements of a particular Ad generation template may be
11 assigned the same score as the Ad generation template itself.

12

Ad Evolution (8875, 8925)

13 **[00558]** In one embodiment, scores assigned to Ad generation templates and/or
14 Ad generation template elements may be directed to the Ad Evolver module 8875. The
15 Ad Evolver module may be configured to process the scores in order to rank existing Ad
16 generation templates and/or to generate new Ad generation templates. Ad generation
17 templates and/or Ad generation template elements having higher scores may be
18 directed by the Ad Evolver module to be more prevalent in subsequent Ad distribution
19 than those having lower scores. This may be accomplished in a number of different
20 ways within various embodiments of the present invention.

21 **[00559]** FIGURE 94 provides an illustrative example of a logic flow in one
22 embodiment of Ad evolution. In this embodiment, referred to as "Ad duplication", web
23 user responses to whole Ads are registered and processed by the Feedback Evaluator
24 module 8860 and passed to the Ad Evolver module 8875. At 9401, the Ad Evolver
25 checks if the Ad generation template score is less than some minimum value. If so, the
26 particular Ad generation template is discarded from the Ad generation template DB
27 9405 and consequently disregarded in future distribution generations of Ads. If the Ad
28 generation template score is sufficiently high, a reproductive probability (RP) may be
29 assigned to the Ad generation template based on the score 9410. For example, a score S
30 may be translated to an RP by taking $RP = 0.5^{[N^{(-|S|)}]}$, where N is a positive integer

1 greater than 1. In an alternative embodiment, an RP score may be directly computed by
2 and passed from the Feedback Evaluator module to the Ad Evolver module. Once an RP
3 score has been established for the Ad generation template, the Ad Evolver module may
4 stochastically reproduce, duplicate, or copy Ad generation templates with likelihoods
5 based on their respective RP scores.

6 **[00560]** For example, the Ad Evolver may generate a random number (RAND)
7 uniformly distributed between 0 and 1 at 9415 and check whether the RP for a given Ad
8 generation template is greater than RAND 9420. If not, then the propagation of the Ad
9 generation template may be demoted or inhibited 9425 and, if $RP > RAND$, the
10 propagation of the Ad generation template may be promoted 9430. The probability that
11 $RP > RAND$ is equal to RP. Accordingly, this condition ties promotion of Ad template
12 propagation to the probability score established at 9410.

13 **[00561]** In alternative implementations, RAND may be taken to be a random
14 variable with an arbitrary and/or non-uniform distribution in order to further weight
15 the selection of Ad generation templates as desired for particular implementations of
16 the present invention. At 9435, a determination is made as to whether there are
17 additional Ads to evaluate, and if so, the process returns to 9401. In an alternative
18 implementation, a single RAND is selected for comparison with RPs for all Ads under
19 consideration at a given time. In another implementation, a single RAND is selected for
20 all identical or related copies of an Ad generation template in the Ad generation
21 template DB. In another alternate implementation, the selection probabilities may be
22 used in combination with Ad generation template selection weightings. The Ad
23 generation template selection weighting is a system tool used in distributing targeted
24 advertisements to particular web users.

25 **[00562]** Promotion and/or demotion of Ad propagation may proceed in a number
26 of different ways within various Ad duplication implementations of the present
27 invention. In one implementation, promotion of Ad propagation may entail duplication
28 of the Ad generation template into one or more identical copies in the Ad generation
29 template DB. The existence of multiple, identical copies of a particular Ad generation
30 template in the Ad generation template DB may increase the likelihood that the
31 particular Ad generation template will be selected for future Ad generation. In one

1 implementation, demotion of Ad propagation may entail deletion of one or more copies
2 of an Ad generation template from the Ad generation template DB. In another
3 implementation, the number of duplicate copies made or deleted for a given Ad
4 generation template is affected by the Ad generation template score, RP, and/or the like.
5 In an implementation, a particular Ad generation template score, RP, or difference
6 between RAND and RP entails that the Ad generation template is neither deleted nor
7 duplicated.

8 **[00563]** In another implementation, Ad generation template duplication and/or
9 deletion is constrained by a requirement that the total number of Ad generation
10 templates in the Ad generation template DB be equal to or within a specific range of a
11 specified quota. In yet another implementation, promotion or demotion of Ad
12 propagation may entail adjustment of a weighting factor associated with the Ad
13 generation template. For example, this may occur in an embodiment wherein selection
14 of an Ad generation template for Ad generation queries the Ad generation template
15 weighting factor and preferentially selects Ad generation templates based on their
16 corresponding weighting factors. In another implementation, the degree of adjustment
17 to the weighting factor made for a given Ad generation template is affected by the Ad
18 generation template score, RP, and/or the like.

19 **[00564]** FIGURE 95 exhibits a schematic, illustrative example of three generations
20 (9501, 9505, 9510) of Ad generation templates in one implementation of Ad duplication,
21 where each Ad generation template's characterizing features are symbolically
22 represented by a shape. For example, the features of Ad generation template 9515 are
23 represented by a triangle. The RP and RAND for the evolution of 9515 are indicated at
24 9520. In this illustrative implementation, an RP within 0.1 of RAND results in neither
25 duplication nor deletion, $RP < RAND - 0.1$ results in deletion, $RAND + 0.1 < RP <$
26 $RAND + 0.2$ results in single duplication (i.e., the original Ad generation template plus
27 one copy), $RAND + 0.2 < RP < RAND + 0.3$ results in double duplication, and so on.
28 Thus, 9515 is duplicated once, yielding second generation Ad generation templates 9525
29 and 9530. Ad generation template 9530 has also acquired a "mutation" 9535, or
30 variation, in its characterizing features. Mutations are discussed in greater detail below.
31 The RP and RAND for the evolution of 9525 and 9530 are indicated at 9540 and 9545

1 respectively (note the mutation 9535 has resulted in a higher RP for 9530 than for the
2 parent template 9515), and the resulting third generation Ad generation templates are
3 exhibited at 9550, 9555, and 9560.

4 **[00565]** The first generation Ad generation template 9565 (characterizing features
5 represented by a square) dies out in the first evolution stage due to its corresponding RP
6 and RAND values 9570. The first generation Ad generation template 9575
7 (characterizing features represented by a pentagon) has corresponding RP and RAND
8 values shown at 9580. Consequently, it yields three second generation Ad generation
9 template children (9585, 9590, and 9595) which, collectively, have RP and RAND values
10 shown at 95100 and, therefore, yield third generation children Ad generation templates
11 (95105, 95110, and 95115).

12 **[00566]** FIGURE 96 provides an illustrative example of a logic flow associated with
13 another embodiment of Ad evolution. In this embodiment, referred to as "Ad
14 recombination", web user responses to whole Ads are registered, processed by the
15 Feedback Evaluator module 8860 and passed to the Ad Evolver module 8875. At 9601,
16 the Ad Evolver checks if the Ad generation template score is less than some minimum
17 value. If so, the particular Ad generation template is discarded 9605. Otherwise, the
18 system assigns RPs to Ad generation templates based on the Ad generation template
19 scores 9610. Two Ad generation templates to be recombined by the system are then
20 selected with likelihood based on their RP values 9615. In an alternative
21 implementation, more than two Ad generation templates may be selected for
22 recombination at this stage. Selection of Ad generation templates may be accomplished,
23 in one illustrative example, by generating a value for a uniformly distributed random
24 variable RAND, separating Ad generation templates based on whether $RP < RAND$ or RP
25 $> RAND$, and selecting any two Ad generation templates randomly from the latter group
26 with all Ad generation templates in this group treated on an equal footing.

27 **[00567]** At 9620, the system checks whether the Ad generation templates are
28 compatible with each other. In one implementation, this may be accomplished by
29 querying a collection of Ad generation template recombination rules that specify which
30 Ad generation template elements and/or characteristics may render two Ads
31 incompatible (for example, an Ad generation template configured to yield a banner Ad

1 may be incompatible with an Ad generation template configured to yield memory-
2 intensive video content). If the Ad generation templates are incompatible, the system
3 returns to 9615 and selects new Ad generation template(s). Otherwise, complementary
4 features are randomly selected from the two Ad generation templates at 9625.

5 **[00568]** In one implementation, this may be accomplished by randomly selecting
6 half of the features in each Ad generation template. In another implementation, this
7 may be accomplished by associating similar features in the Ad generation templates
8 (e.g., headings, footings, design templates, size specifications, etc.) and then selecting
9 randomly between matched features with equal probability for the features originating
10 from each Ad generation template. Any leftover, unmatched features may, then, be
11 randomly included or excluded in the final Ad generation template with equal
12 probabilities. At 9630, the new Ad generation template is constructed from the Ad
13 generation template features selected at 9625.

14 **[00569]** Finally, the system decides whether or not there are further Ad generation
15 templates to consider at 9635. In one implementation, parent Ad generation templates
16 are returned to the Ad generation template DB with the children Ad generation
17 templates, while in another implementation, parent Ad generation templates may be
18 discarded from the Ad generation template DB after a finite lifetime and/or number of
19 generations. In one implementation, child Ad generation templates are made available
20 for display to system administrators who may decide whether they are suitable for
21 inclusion in the Ad generation template DB.

22 **[00570]** FIGURE 97 shows an illustrative example of Ad recombination in one
23 embodiment, whereby two parent Ad generation templates (9701 and 9705) are
24 recombined into a child Ad generation template 9708. Though representations of the
25 Ads themselves are shown in the figure for clarity, it may in fact be the underlying Ad
26 generation templates that are recombined. Each Ad generation template underlying the
27 Ads in Fig. 97 has a number of characterizing elements.

28 **[00571]** For example, Ad generation template 9701 specifies a star motif 9710; a
29 global Arial font; a left-justified, 14 pt., underlined job title heading 9715; a left-justified,
30 12 pt. education level with underlined portion 9720; and a 12 pt., italicized salary. Ad

1 generation template 9705, on the other hand, specifies a global Times New Roman font;
2 a center-justified, 14 pt., underlined job title heading; a left-justified, 12 pt. company; a
3 left-justified, 12 pt. job description; and a left-justified, 12 pt., underlined clickable link
4 to more information. The child Ad generation template 9708 takes the star motif 9755
5 and salary specification 9770 from Ad generation template 9701 and the job title
6 heading, company, and clickable link from Ad generation template 9705.

7 **[00572]** FIGURE 98 provides a visualization of three successive Ad generation
8 template generations in an Ad recombination embodiment. In this Ad generation
9 template "family tree", Ad generation template features are symbolized by four letters.
10 For example, Ad generation template 9801 has features ABCD. A hypothetical RP value
11 for each Ad generation template is also indicated (e.g., 9801 has $RP = 0.45$). Parent
12 templates (9801 with 9805 and 9810 with 9815) are recombined to form child templates
13 (respectively 9820/9825 and 9830/9835), each possessing different combinations of
14 parent template features. The combination of features in template 9820 apparently
15 yields poor performance, as its $RP = 0.17$ is low and, consequently, it yields no further
16 children.

17 **[00573]** Templates 9825 and 9830 recombine to yield child templates 9845 and
18 9850, and template 9835 recombines with a newly introduced template 9840 to yield
19 child templates 9855 and 9860. Both this family tree and the lineage chart in Fig. 95
20 provide visualizations of Ad generation template performance and evolution, and thus
21 facilitate Ad performance tracking. In an implementation of the system, system
22 administrators may access these types of evolutionary visualizations to assist in
23 determining system performance and efficacy. Further, the evolutionary visualizations
24 may be based on ad generation templates and/or distributed advertisements (e.g.,
25 illustrating ad templates 9060 and/or distributed ads 9070).

26 **[00574]** In one embodiment, Ad generation template evolution may be directed or
27 restricted by grouping similar, compatible, and/or common-purpose Ad generation
28 templates together based on Ad generation template characteristics and/or other
29 labeling metadata. For example, the system may elect to only recombine Ad generation
30 templates that generate pop-up Ads or to only recombine Ad generation templates
31 marked with a "job listing template" metadata label with each other. This type of

1 strategy respects the possibility that not all Ad types may be best served by the same Ad
2 generation templates (e.g., consumer product Ads may be more effective with one
3 template while job listings may be more effective with another).

4 **[00575]** In another embodiment, aspects of the Engine may be employed to target
5 Ads to particular web users. For example, the system may monitor and collect web user
6 interactions with a plurality of Ads, collect web user information, parse and/or separate
7 collected interactions based on web user group identifiers (e.g., demographics,
8 consumer behavior and/or shopping patterns, web surfing habits, address, education
9 level, etc.), and serve Ads to each web user group generated using the most successful
10 Ad generation templates and/or Ad generation routines within that group.

11 **[00576]** In another implementation, the system may elect to evolve Ad generation
12 routines separately for different web user group identifiers. For example, web users
13 with 2 or more children may have a wholly separate Ad evolution process from web
14 users who have less than 2 children, yielding separate Ad generation template databases
15 for each demographic group.

16 **[00577]** In another embodiment, particular Ad generation template elements
17 and/or characteristics may be tied together or linked in such a way that they tend to
18 carry over together from parent Ad generation templates into child Ad generation
19 templates. For example, if two Ad generation template elements and/or characteristics
20 (e.g., pop-up Ad and prominently featured salary) are correlated with a consistent
21 and/or high-degree of performance success, they may be automatically coalesced into a
22 single Ad generation template characteristic or brought to the attention of a system
23 administrator who may elect to coalesce them.

24 **[00578]** Ad generation template elements and/or characteristics that are good
25 candidates for coalescing may be identified heuristically by a system administrator using
26 evolutionary visualizations discussed above or may be automatically detected.
27 Automatic detection may proceed in a variety of different ways in various
28 implementations of the present invention. In one implementation, the system may
29 restrict its attention to Ad generation templates exhibiting the best performance (e.g.,
30 $RP > 0.9$) in a given generation and then coalescing Ad generation template elements

1 and/or characteristics that occur together most frequently in the set of best performing
2 Ad generation templates. For example, if a large majority of the best performing Ad
3 generation templates are pop-up Ads with a prominently featured salary, then these
4 characteristics may be tied together into a single characteristic in subsequent
5 generations.

6 **[00579]** In another embodiment, the system may admit restrictions on
7 combinations of Ad generation template characteristics, preventing them from existing
8 within a single Ad generation template in subsequent generations. Such restrictions
9 may be implemented manually by a system administrator or automatically by the
10 system itself. Ad generation template elements and/or characteristics that are good
11 candidates for restriction may be identified heuristically by a system administrator or
12 may be automatically detected.

13 **[00580]** Automatic detection may proceed in a variety of different ways in various
14 implementations of the present invention. In one implementation, the system may
15 restrict its attention to Ad generation templates exhibiting the worst performance (e.g.,
16 $RP < 0.1$) in a given generation and then restrict Ad generation template elements
17 and/or characteristics that occur together most frequently in the set of worst performing
18 Ad generation templates. For example, if a large majority of worst performing Ad
19 generation templates are banner Ads with a 300 x 600 size designation, then these
20 characteristics may be restricted from occurring together in subsequent Ad generation
21 templates. In one implementation, restrictions on Ad generation template element
22 and/or characteristic combinations may be stored in a set of Ad generation template
23 recombination rules.

24 **[00581]** In another embodiment, Ad generation template elements may be
25 spontaneously and/or randomly changed by the system. Such changes, analogous to
26 mutations in biological evolution, may yield novel Ad generation templates that may
27 otherwise not have formed by other Ad evolution processes. In one implementation,
28 child Ad generation templates may be randomly selected with a low probability to have
29 one or more of their Ad generation template elements altered by the Engine. Such
30 altering of Ad generation template elements may be accomplished, in one example, by
31 replacing an Ad generation template element in the mutating Ad generation template

1 with a randomly selected Ad generation template element from another randomly
2 selected Ad generation template. In yet another embodiment, the system may admit
3 manual changes to Ad generation templates and/or Ad generation template elements by
4 a system administrator.

5 **[00582]** The Engine provides an efficient and effective Ad performance tracking
6 and learning system that may be applied to a wide variety of marketing and information
7 dispensation applications. In one embodiment, the Engine may process advertisements
8 pertaining to job listings and/or opportunities. The same base job listing data entry may
9 be incorporated into a variety of different formats that exhibit varying degrees of
10 "attractiveness" and/or emphasize and/or de-emphasize various job listing elements.
11 The Engine may then monitor the performance of the job listings by any of the means
12 listed above and/or, in particular, by correlating web user applications for jobs, job
13 interviews, or job offers with advertisement interactions and/or impressions.
14 Subsequent job listing advertisements created by the system may be designed to exploit
15 the most effective Ad generation template characteristics.

16 **[00583]** In another embodiment, the Engine may be employed to improve job
17 seeker profiles and/or resume listings. A job seeker may submit characterizing
18 information, profiles, and/or resumes to the system, which may then parse and/or
19 incorporate the submission into a job seeker listing for display to possible employers.
20 Performance of a job seeker listing may be measured by employer ratings and/or
21 interactions with the listing, such as employer click-throughs, mouse-overs,
22 impressions, interview offers, job offers, and/or the like. The job listings may then be
23 refined based on the performance of various resume and/or job seeker listing generation
24 templates. In one embodiment, analysis of template performance may form the basis
25 for a resume and/or job seeker listing consultation service.

26 **[00584]** In another embodiment, the Engine may form the basis for a graded
27 advertisement pricing system. In tracking Ad generation template performance, the
28 Engine may determine the degree to which particular Ad generation templates and/or
29 Ad generation template elements contribute to Ad effectiveness. Consequently, a
30 graded advertisement pricing system may be established, whereby a graded premium
31 may be charged to companies supplying base data entries for inclusion in

1 advertisements for the use of Ad generation templates and/or Ad generation template
2 elements proven to be most attractive and/or effective in eliciting desired web user
3 responses. In another embodiment, Ad generation templates and/or Ad generation
4 template elements that yield particularly high performance metric scores may be
5 marked by the system as candidates for intellectual property protection. These marked
6 templates may be brought to the attention of a system administrator or automatically
7 submitted for legal consideration.

8

9

SMP Controller

10 **[00585]** FIGURE 99 shows a block diagram illustrating embodiments of a SMP
11 controller. In this embodiment, the SMP controller 9901 may serve to aggregate,
12 process, store, search, serve, identify, instruct, generate, match, and/or facilitate
13 interactions with a computer through information technologies, and/or other related
14 data.

15 **[00586]** Typically, users, which may be people and/or other systems, may engage
16 information technology systems (e.g., computers) to facilitate information processing.
17 In turn, computers employ processors to process information; such processors 9903
18 may be referred to as central processing units (CPU). One form of processor is referred
19 to as a microprocessor. CPUs use communicative circuits to pass binary encoded signals
20 acting as instructions to enable various operations. These instructions may be
21 operational and/or data instructions containing and/or referencing other instructions
22 and data in various processor accessible and operable areas of memory 9929 (e.g.,
23 registers, cache memory, random access memory, etc.). Such communicative
24 instructions may be stored and/or transmitted in batches (e.g., batches of instructions)
25 as programs and/or data components to facilitate desired operations. These stored
26 instruction codes, e.g., programs, may engage the CPU circuit components and other
27 motherboard and/or system components to perform desired operations. One type of
28 program is a computer operating system, which, may be executed by CPU on a
29 computer; the operating system enables and facilitates users to access and operate

1 computer information technology and resources. Some resources that may be employed
2 in information technology systems include: input and output mechanisms through
3 which data may pass into and out of a computer; memory storage into which data may
4 be saved; and processors by which information may be processed. These information
5 technology systems may be used to collect data for later retrieval, analysis, and
6 manipulation, which may be facilitated through a database program. These information
7 technology systems provide interfaces that allow users to access and operate various
8 system components.

9 **[00587]** In one embodiment, the SMP controller 9901 may be connected to and/or
10 communicate with entities such as, but not limited to: one or more users from user
11 input devices 9911; peripheral devices 9912; an optional cryptographic processor device
12 9928; and/or a communications network 9913.

13 **[00588]** Networks are commonly thought to comprise the interconnection and
14 interoperation of clients, servers, and intermediary nodes in a graph topology. It should
15 be noted that the term “server” as used throughout this application refers generally to a
16 computer, other device, program, or combination thereof that processes and responds to
17 the requests of remote users across a communications network. Servers serve their
18 information to requesting “clients.” The term “client” as used herein refers generally to a
19 computer, program, other device, user and/or combination thereof that is capable of
20 processing and making requests and obtaining and processing any responses from
21 servers across a communications network. A computer, other device, program, or
22 combination thereof that facilitates, processes information and requests, and/or
23 furthers the passage of information from a source user to a destination user is
24 commonly referred to as a “node.” Networks are generally thought to facilitate the
25 transfer of information from source points to destinations. A node specifically tasked
26 with furthering the passage of information from a source to a destination is commonly
27 called a “router.” There are many forms of networks such as Local Area Networks
28 (LANs), Pico networks, Wide Area Networks (WANs), Wireless Networks (WLANs), etc.
29 For example, the Internet is generally accepted as being an interconnection of a
30 multitude of networks whereby remote clients and servers may access and interoperate
31 with one another.

1 **[00589]** The SMP controller 9901 may be based on computer systems that may
2 comprise, but are not limited to, components such as: a computer systemization 9902
3 connected to memory 9929.

4 **Computer Systemization**

5 **[00590]** A computer systemization 9902 may comprise a clock 9930, central
6 processing unit (“CPU(s)” and/or “processor(s)” (these terms are used interchangeable
7 throughout the disclosure unless noted to the contrary)) 9903, a memory 9929 (e.g., a
8 read only memory (ROM) 9906, a random access memory (RAM) 9905, etc.), and/or an
9 interface bus 9907, and most frequently, although not necessarily, are all interconnected
10 and/or communicating through a system bus 9904 on one or more (mother)board(s)
11 9902 having conductive and/or otherwise transportive circuit pathways through which
12 instructions (e.g., binary encoded signals) may travel to effectuate communications,
13 operations, storage, etc. The computer systemization may be connected to a power
14 source 9986; e.g., optionally the power source may be internal. Optionally, a
15 cryptographic processor 9926 and/or transceivers (e.g., ICs) 9974 may be connected to
16 the system bus. In another embodiment, the cryptographic processor and/or
17 transceivers may be connected as either internal and/or external peripheral devices
18 9912 via the interface bus I/O. In turn, the transceivers may be connected to antenna(s)
19 9975, thereby effectuating wireless transmission and reception of various
20 communication and/or sensor protocols; for example the antenna(s) may connect to:
21 Texas Instruments WiLink WL1283 transceiver chip (e.g., providing 802.11n, Bluetooth
22 3.0, FM, global positioning system (GPS) (thereby allowing SMP controller to determine
23 its location)); Broadcom BCM4329FKUBG transceiver chip (e.g., providing 802.11n,
24 Bluetooth 2.1 + EDR, FM, etc.); a Broadcom BCM4750IUB8 receiver chip (e.g., GPS); an
25 Infineon Technologies X-Gold 618-PMB9800 (e.g., providing 2G/3G HSDPA/HSUPA
26 communications); and/or the like. The system clock typically has a crystal oscillator and
27 generates a base signal through the computer systemization’s circuit pathways. The
28 clock is typically coupled to the system bus and various clock multipliers that will
29 increase or decrease the base operating frequency for other components interconnected
30 in the computer systemization. The clock and various components in a computer

1 systemization drive signals embodying information throughout the system. Such
2 transmission and reception of instructions embodying information throughout a
3 computer systemization may be commonly referred to as communications. These
4 communicative instructions may further be transmitted, received, and the cause of
5 return and/or reply communications beyond the instant computer systemization to:
6 communications networks, input devices, other computer systemizations, peripheral
7 devices, and/or the like. It should be understood that in alternative embodiments, any
8 of the above components may be connected directly to one another, connected to the
9 CPU, and/or organized in numerous variations employed as exemplified by various
10 computer systems.

11 **[00591]** The CPU comprises at least one high-speed data processor adequate to
12 execute program components for executing user and/or system-generated requests.
13 Often, the processors themselves will incorporate various specialized processing units,
14 such as, but not limited to: integrated system (bus) controllers, memory management
15 control units, floating point units, and even specialized processing sub-units like
16 graphics processing units, digital signal processing units, and/or the like. Additionally,
17 processors may include internal fast access addressable memory, and be capable of
18 mapping and addressing memory 9929 beyond the processor itself; internal memory
19 may include, but is not limited to: fast registers, various levels of cache memory (e.g.,
20 level 1, 2, 3, etc.), RAM, etc. The processor may access this memory through the use of a
21 memory address space that is accessible via instruction address, which the processor
22 can construct and decode allowing it to access a circuit path to a specific memory
23 address space having a memory state. The CPU may be a microprocessor such as:
24 AMD's Athlon, Duron and/or Opteron; ARM's application, embedded and secure
25 processors; IBM and/or Motorola's DragonBall and PowerPC; IBM's and Sony's Cell
26 processor; Intel's Celeron, Core (2) Duo, Itanium, Pentium, Xeon, and/or XScale;
27 and/or the like processor(s). The CPU interacts with memory through instruction
28 passing through conductive and/or transportive conduits (e.g., (printed) electronic
29 and/or optic circuits) to execute stored instructions (i.e., program code) according to
30 conventional data processing techniques. Such instruction passing facilitates
31 communication within the SMP controller and beyond through various interfaces.

1 Should processing requirements dictate a greater amount speed and/or capacity,
2 distributed processors (e.g., Distributed SMP), mainframe, multi-core, parallel, and/or
3 super-computer architectures may similarly be employed. Alternatively, should
4 deployment requirements dictate greater portability, smaller Personal Digital Assistants
5 (PDAs) may be employed.

6 **[00592]** Depending on the particular implementation, features of the SMP may be
7 achieved by implementing a microcontroller such as CAST's R8051XC2 microcontroller;
8 Intel's MCS 51 (i.e., 8051 microcontroller); and/or the like. Also, to implement certain
9 features of the SMP, some feature implementations may rely on embedded components,
10 such as: Application-Specific Integrated Circuit ("ASIC"), Digital Signal Processing
11 ("DSP"), Field Programmable Gate Array ("FPGA"), and/or the like embedded
12 technology. For example, any of the SMP component collection (distributed or
13 otherwise) and/or features may be implemented via the microprocessor and/or via
14 embedded components; e.g., via ASIC, coprocessor, DSP, FPGA, and/or the like.
15 Alternately, some implementations of the SMP may be implemented with embedded
16 components that are configured and used to achieve a variety of features or signal
17 processing.

18 **[00593]** Depending on the particular implementation, the embedded components
19 may include software solutions, hardware solutions, and/or some combination of both
20 hardware/software solutions. For example, SMP features discussed herein may be
21 achieved through implementing FPGAs, which are a semiconductor devices containing
22 programmable logic components called "logic blocks", and programmable
23 interconnects, such as the high performance FPGA Virtex series and/or the low cost
24 Spartan series manufactured by Xilinx. Logic blocks and interconnects can be
25 programmed by the customer or designer, after the FPGA is manufactured, to
26 implement any of the SMP features. A hierarchy of programmable interconnects allow
27 logic blocks to be interconnected as needed by the SMP system designer/administrator,
28 somewhat like a one-chip programmable breadboard. An FPGA's logic blocks can be
29 programmed to perform the operation of basic logic gates such as AND, and XOR, or
30 more complex combinational operators such as decoders or mathematical operations. In
31 most FPGAs, the logic blocks also include memory elements, which may be circuit flip-

1 flops or more complete blocks of memory. In some circumstances, the SMP may be
2 developed on regular FPGAs and then migrated into a fixed version that more resembles
3 ASIC implementations. Alternate or coordinating implementations may migrate SMP
4 controller features to a final ASIC instead of or in addition to FPGAs. Depending on the
5 implementation all of the aforementioned embedded components and microprocessors
6 may be considered the “CPU” and/or “processor” for the SMP.

7 **Power Source**

8 **[00594]** The power source 9986 may be of any standard form for powering small
9 electronic circuit board devices such as the following power cells: alkaline, lithium
10 hydride, lithium ion, lithium polymer, nickel cadmium, solar cells, and/or the like.
11 Other types of AC or DC power sources may be used as well. In the case of solar cells, in
12 one embodiment, the case provides an aperture through which the solar cell may
13 capture photonic energy. The power cell 9986 is connected to at least one of the
14 interconnected subsequent components of the SMP thereby providing an electric
15 current to all subsequent components. In one example, the power source 9986 is
16 connected to the system bus component 9904. In an alternative embodiment, an outside
17 power source 9986 is provided through a connection across the I/O 9908 interface. For
18 example, a USB and/or IEEE 1394 connection carries both data and power across the
19 connection and is therefore a suitable source of power.

20 **Interface Adapters**

21 **[00595]** Interface bus(es) 9907 may accept, connect, and/or communicate to a
22 number of interface adapters, conventionally although not necessarily in the form of
23 adapter cards, such as but not limited to: input output interfaces (I/O) 9908, storage
24 interfaces 9909, network interfaces 9910, and/or the like. Optionally, cryptographic
25 processor interfaces 9927 similarly may be connected to the interface bus. The interface
26 bus provides for the communications of interface adapters with one another as well as
27 with other components of the computer systemization. Interface adapters are adapted
28 for a compatible interface bus. Interface adapters conventionally connect to the
29 interface bus via a slot architecture. Conventional slot architectures may be employed,

1 such as, but not limited to: Accelerated Graphics Port (AGP), Card Bus, (Extended)
2 Industry Standard Architecture ((E)ISA), Micro Channel Architecture (MCA), NuBus,
3 Peripheral Component Interconnect (Extended) (PCI(X)), PCI Express, Personal
4 Computer Memory Card International Association (PCMCIA), and/or the like.

5 **[00596]** Storage interfaces 9909 may accept, communicate, and/or connect to a
6 number of storage devices such as, but not limited to: storage devices 9914, removable
7 disc devices, and/or the like. Storage interfaces may employ connection protocols such
8 as, but not limited to: (Ultra) (Serial) Advanced Technology Attachment (Packet
9 Interface) ((Ultra) (Serial) ATA(PI)), (Enhanced) Integrated Drive Electronics ((E)IDE),
10 Institute of Electrical and Electronics Engineers (IEEE) 1394, fiber channel, Small
11 Computer Systems Interface (SCSI), Universal Serial Bus (USB), and/or the like.

12 **[00597]** Network interfaces 9910 may accept, communicate, and/or connect to a
13 communications network 9913. Through a communications network 9913, the SMP
14 controller is accessible through remote clients 9933b (e.g., computers with web
15 browsers) by users 9933a. Network interfaces may employ connection protocols such as,
16 but not limited to: direct connect, Ethernet (thick, thin, twisted pair 10/100/1000 Base
17 T, and/or the like), Token Ring, wireless connection such as IEEE 802.11a-x, and/or the
18 like. Should processing requirements dictate a greater amount speed and/or capacity,
19 distributed network controllers (e.g., Distributed SMP), architectures may similarly be
20 employed to pool, load balance, and/or otherwise increase the communicative
21 bandwidth required by the SMP controller. A communications network may be any one
22 and/or the combination of the following: a direct interconnection; the Internet; a Local
23 Area Network (LAN); a Metropolitan Area Network (MAN); an Operating Missions as
24 Nodes on the Internet (OMNI); a secured custom connection; a Wide Area Network
25 (WAN); a wireless network (e.g., employing protocols such as, but not limited to a
26 Wireless Application Protocol (WAP), I-mode, and/or the like); and/or the like. A
27 network interface may be regarded as a specialized form of an input output interface.
28 Further, multiple network interfaces 9910 may be used to engage with various
29 communications network types 9913. For example, multiple network interfaces may be
30 employed to allow for the communication over broadcast, multicast, and/or unicast
31 networks.

1 **[00598]** Input Output interfaces (I/O) 9908 may accept, communicate, and/or
2 connect to user input devices 9911, peripheral devices 9912, cryptographic processor
3 devices 9928, and/or the like. I/O may employ connection protocols such as, but not
4 limited to: audio: analog, digital, monaural, RCA, stereo, and/or the like; data: Apple
5 Desktop Bus (ADB), IEEE 1394a-b, serial, universal serial bus (USB); infrared; joystick;
6 keyboard; midi; optical; PC AT; PS/2; parallel; radio; video interface: Apple Desktop
7 Connector (ADC), BNC, coaxial, component, composite, digital, Digital Visual Interface
8 (DVI), high-definition multimedia interface (HDMI), RCA, RF antennae, S-Video, VGA,
9 and/or the like; wireless transceivers: 802.11a/b/g/n/x; Bluetooth; cellular (e.g., code
10 division multiple access (CDMA), high speed packet access (HSPA(+)), high-speed
11 downlink packet access (HSDPA), global system for mobile communications (GSM),
12 long term evolution (LTE), WiMax, etc.); and/or the like. One typical output device may
13 include a video display, which typically comprises a Cathode Ray Tube (CRT) or Liquid
14 Crystal Display (LCD) based monitor with an interface (e.g., DVI circuitry and cable)
15 that accepts signals from a video interface, may be used. The video interface composites
16 information generated by a computer systemization and generates video signals based
17 on the composited information in a video memory frame. Another output device is a
18 television set, which accepts signals from a video interface. Typically, the video interface
19 provides the composited video information through a video connection interface that
20 accepts a video display interface (e.g., an RCA composite video connector accepting an
21 RCA composite video cable; a DVI connector accepting a DVI display cable, etc.).

22 **[00599]** User input devices 9911 often are a type of peripheral device 512 (see
23 below) and may include: card readers, dongles, finger print readers, gloves, graphics
24 tablets, joysticks, keyboards, microphones, mouse (mice), remote controls, retina
25 readers, touch screens (e.g., capacitive, resistive, etc.), trackballs, trackpads, sensors
26 (e.g., accelerometers, ambient light, GPS, gyroscopes, proximity, etc.), styluses, and/or
27 the like.

28 **[00600]** Peripheral devices 9912 may be connected and/or communicate to I/O
29 and/or other facilities of the like such as network interfaces, storage interfaces, directly
30 to the interface bus, system bus, the CPU, and/or the like. Peripheral devices may be
31 external, internal and/or part of the SMP controller. Peripheral devices may include:

1 antenna, audio devices (e.g., line-in, line-out, microphone input, speakers, etc.),
2 cameras (e.g., still, video, webcam, etc.), dongles (e.g., for copy protection, ensuring
3 secure transactions with a digital signature, and/or the like), external processors (for
4 added capabilities; e.g., crypto devices 528), force-feedback devices (e.g., vibrating
5 motors), network interfaces, printers, scanners, storage devices, transceivers (e.g.,
6 cellular, GPS, etc.), video devices (e.g., goggles, monitors, etc.), video sources, visors,
7 and/or the like. Peripheral devices often include types of input devices (e.g., cameras).

8 **[00601]** It should be noted that although user input devices and peripheral devices
9 may be employed, the SMP controller may be embodied as an embedded, dedicated,
10 and/or monitor-less (i.e., headless) device, wherein access would be provided over a
11 network interface connection.

12 **[00602]** Cryptographic units such as, but not limited to, microcontrollers,
13 processors 9926, interfaces 9927, and/or devices 9928 may be attached, and/or
14 communicate with the SMP controller. A MC68HC16 microcontroller, manufactured by
15 Motorola Inc., may be used for and/or within cryptographic units. The MC68HC16
16 microcontroller utilizes a 16-bit multiply-and-accumulate instruction in the 16 MHz
17 configuration and requires less than one second to perform a 512-bit RSA private key
18 operation. Cryptographic units support the authentication of communications from
19 interacting agents, as well as allowing for anonymous transactions. Cryptographic units
20 may also be configured as part of the CPU. Equivalent microcontrollers and/or
21 processors may also be used. Other commercially available specialized cryptographic
22 processors include: Broadcom's CryptoNetX and other Security Processors; nCipher's
23 nShield; SafeNet's Luna PCI (e.g., 7100) series; Semaphore Communications' 40 MHz
24 Roadrunner 184; Sun's Cryptographic Accelerators (e.g., Accelerator 6000 PCIe Board,
25 Accelerator 500 Daughtercard); Via Nano Processor (e.g., L2100, L2200, U2400) line,
26 which is capable of performing 500+ MB/s of cryptographic instructions; VLSI
27 Technology's 33 MHz 6868; and/or the like.

28

Memory

29 **[00603]** Generally, any mechanization and/or embodiment allowing a processor to
30 affect the storage and/or retrieval of information is regarded as memory 9929. However,

1 memory is a fungible technology and resource, thus, any number of memory
2 embodiments may be employed in lieu of or in concert with one another. It is to be
3 understood that the SMP controller and/or a computer systemization may employ
4 various forms of memory 9929. For example, a computer systemization may be
5 configured wherein the operation of on-chip CPU memory (e.g., registers), RAM, ROM,
6 and any other storage devices are provided by a paper punch tape or paper punch card
7 mechanism; however, such an embodiment would result in an extremely slow rate of
8 operation. In a typical configuration, memory 9929 will include ROM 9906, RAM 9905,
9 and a storage device 9914. A storage device 9914 may be any conventional computer
10 system storage. Storage devices may include a drum; a (fixed and/or removable)
11 magnetic disk drive; a magneto-optical drive; an optical drive (i.e., Blu-ray, CD
12 ROM/RAM/Recordable (R)/ReWritable (RW), DVD R/RW, HD DVD R/RW etc.); an
13 array of devices (e.g., Redundant Array of Independent Disks (RAID)); solid state
14 memory devices (USB memory, solid state drives (SSD), etc.); other processor-readable
15 storage mediums; and/or other devices of the like. Thus, a computer systemization
16 generally requires and makes use of memory.

17

Component Collection

18 **[00604]** The memory 9929 may contain a collection of program and/or database
19 components and/or data such as, but not limited to: operating system component(s)
20 9915 (operating system); information server component(s) 9916 (information server);
21 user interface component(s) 9917 (user interface); Web browser component(s) 9918
22 (Web browser); database(s) 9919; mail server component(s) 9921; mail client
23 component(s) 9922; cryptographic server component(s) 9920 (cryptographic server);
24 the SMP component(s) 9935; and/or the like (i.e., collectively a component collection).
25 These components may be stored and accessed from the storage devices and/or from
26 storage devices accessible through an interface bus. Although non-conventional
27 program components such as those in the component collection, typically, are stored in
28 a local storage device 9914, they may also be loaded and/or stored in memory such as:
29 peripheral devices, RAM, remote storage facilities through a communications network,
30 ROM, various forms of memory, and/or the like.

Operating System

1
2 **[00605]** The operating system component 9915 is an executable program
3 component facilitating the operation of the SMP controller. Typically, the operating
4 system facilitates access of I/O, network interfaces, peripheral devices, storage devices,
5 and/or the like. The operating system may be a highly fault tolerant, scalable, and
6 secure system such as: Apple Macintosh OS X (Server); AT&T Plan 9; Be OS; Unix and
7 Unix-like system distributions (such as AT&T's UNIX; Berkley Software Distribution
8 (BSD) variations such as FreeBSD, NetBSD, OpenBSD, and/or the like; Linux
9 distributions such as Red Hat, Ubuntu, and/or the like); and/or the like operating
10 systems. However, more limited and/or less secure operating systems also may be
11 employed such as Apple Macintosh OS, IBM OS/2, Microsoft DOS, Microsoft Windows
12 2000/2003/3.1/95/98/CE/Millennium/NT/Vista/XP (Server), Palm OS, and/or the like.
13 An operating system may communicate to and/or with other components in a
14 component collection, including itself, and/or the like. Most frequently, the operating
15 system communicates with other program components, user interfaces, and/or the like.
16 For example, the operating system may contain, communicate, generate, obtain, and/or
17 provide program component, system, user, and/or data communications, requests,
18 and/or responses. The operating system, once executed by the CPU, may enable the
19 interaction with communications networks, data, I/O, peripheral devices, program
20 components, memory, user input devices, and/or the like. The operating system may
21 provide communications protocols that allow the SMP controller to communicate with
22 other entities through a communications network 9913. Various communication
23 protocols may be used by the SMP controller as a subcarrier transport mechanism for
24 interaction, such as, but not limited to: multicast, TCP/IP, UDP, unicast, and/or the
25 like.

Information Server

26
27 **[00606]** An information server component 9916 is a stored program component
28 that is executed by a CPU. The information server may be a conventional Internet
29 information server such as, but not limited to Apache Software Foundation's Apache,
30 Microsoft's Internet Information Server, and/or the like. The information server may

1 allow for the execution of program components through facilities such as Active Server
2 Page (ASP), ActiveX, (ANSI) (Objective-) C (++), C# and/or .NET, Common Gateway
3 Interface (CGI) scripts, dynamic (D) hypertext markup language (HTML), FLASH, Java,
4 JavaScript, Practical Extraction Report Language (PERL), Hypertext Pre-Processor
5 (PHP), pipes, Python, wireless application protocol (WAP), WebObjects, and/or the like.
6 The information server may support secure communications protocols such as, but not
7 limited to, File Transfer Protocol (FTP); HyperText Transfer Protocol (HTTP); Secure
8 Hypertext Transfer Protocol (HTTPS), Secure Socket Layer (SSL), messaging protocols
9 (e.g., America Online (AOL) Instant Messenger (AIM), Application Exchange (APEX),
10 ICQ, Internet Relay Chat (IRC), Microsoft Network (MSN) Messenger Service, Presence
11 and Instant Messaging Protocol (PRIM), Internet Engineering Task Force's (IETF's)
12 Session Initiation Protocol (SIP), SIP for Instant Messaging and Presence Leveraging
13 Extensions (SIMPLE), open XML-based Extensible Messaging and Presence Protocol
14 (XMPP) (i.e., Jabber or Open Mobile Alliance's (OMA's) Instant Messaging and
15 Presence Service (IMPS)), Yahoo! Instant Messenger Service, and/or the like. The
16 information server provides results in the form of Web pages to Web browsers, and
17 allows for the manipulated generation of the Web pages through interaction with other
18 program components. After a Domain Name System (DNS) resolution portion of an
19 HTTP request is resolved to a particular information server, the information server
20 resolves requests for information at specified locations on the SMP controller based on
21 the remainder of the HTTP request. For example, a request such as
22 `http://123.124.125.126/myInformation.html` might have the IP portion of the request
23 "123.124.125.126" resolved by a DNS server to an information server at that IP address;
24 that information server might in turn further parse the http request for the
25 `"/myInformation.html"` portion of the request and resolve it to a location in memory
26 containing the information "myInformation.html." Additionally, other information
27 serving protocols may be employed across various ports, e.g., FTP communications
28 across port 21, and/or the like. An information server may communicate to and/or with
29 other components in a component collection, including itself, and/or facilities of the
30 like. Most frequently, the information server communicates with the SMP database
31 9919, operating systems, other program components, user interfaces, Web browsers,
32 and/or the like.

1 **[00607]** Access to the SMP database may be achieved through a number of
2 database bridge mechanisms such as through scripting languages as enumerated below
3 (e.g., CGI) and through inter-application communication channels as enumerated below
4 (e.g., CORBA, WebObjects, etc.). Any data requests through a Web browser are parsed
5 through the bridge mechanism into appropriate grammars as required by the SMP. In
6 one embodiment, the information server would provide a Web form accessible by a Web
7 browser. Entries made into supplied fields in the Web form are tagged as having been
8 entered into the particular fields, and parsed as such. The entered terms are then passed
9 along with the field tags, which act to instruct the parser to generate queries directed to
10 appropriate tables and/or fields. In one embodiment, the parser may generate queries in
11 standard SQL by instantiating a search string with the proper join/select commands
12 based on the tagged text entries, wherein the resulting command is provided over the
13 bridge mechanism to the SMP as a query. Upon generating query results from the query,
14 the results are passed over the bridge mechanism, and may be parsed for formatting and
15 generation of a new results Web page by the bridge mechanism. Such a new results Web
16 page is then provided to the information server, which may supply it to the requesting
17 Web browser.

18 **[00608]** Also, an information server may contain, communicate, generate, obtain,
19 and/or provide program component, system, user, and/or data communications,
20 requests, and/or responses.

21

User Interface

22 **[00609]** Computer interfaces in some respects are similar to automobile operation
23 interfaces. Automobile operation interface elements such as steering wheels, gearshifts,
24 and speedometers facilitate the access, operation, and display of automobile resources,
25 and status. Computer interaction interface elements such as check boxes, cursors,
26 menus, scrollers, and windows (collectively and commonly referred to as widgets)
27 similarly facilitate the access, capabilities, operation, and display of data and computer
28 hardware and operating system resources, and status. Operation interfaces are
29 commonly called user interfaces. Graphical user interfaces (GUIs) such as the Apple
30 Macintosh Operating System's Aqua, IBM's OS/2, Microsoft's Windows

1 2000/2003/3.1/95/98/CE/Millennium/NT/XP/Vista/7 (i.e., Aero), Unix's X-Windows
2 (e.g., which may include additional Unix graphic interface libraries and layers such as K
3 Desktop Environment (KDE), mythTV and GNU Network Object Model Environment
4 (GNOME)), web interface libraries (e.g., ActiveX, AJAX, (D)HTML, FLASH, Java,
5 JavaScript, etc. interface libraries such as, but not limited to, Dojo, jQuery(UI),
6 MooTools, Prototype, script.aculo.us, SWFObject, Yahoo! User Interface, any of which
7 may be used and) provide a baseline and means of accessing and displaying information
8 graphically to users.

9 **[00610]** A user interface component 9917 is a stored program component that is
10 executed by a CPU. The user interface may be a conventional graphic user interface as
11 provided by, with, and/or atop operating systems and/or operating environments such
12 as already discussed. The user interface may allow for the display, execution,
13 interaction, manipulation, and/or operation of program components and/or system
14 facilities through textual and/or graphical facilities. The user interface provides a facility
15 through which users may affect, interact, and/or operate a computer system. A user
16 interface may communicate to and/or with other components in a component
17 collection, including itself, and/or facilities of the like. Most frequently, the user
18 interface communicates with operating systems, other program components, and/or the
19 like. The user interface may contain, communicate, generate, obtain, and/or provide
20 program component, system, user, and/or data communications, requests, and/or
21 responses.

22 **Web Browser**

23 **[00611]** A Web browser component 9918 is a stored program component that is
24 executed by a CPU. The Web browser may be a conventional hypertext viewing
25 application such as Microsoft Internet Explorer or Netscape Navigator. Secure Web
26 browsing may be supplied with 128bit (or greater) encryption by way of HTTPS, SSL,
27 and/or the like. Web browsers allowing for the execution of program components
28 through facilities such as ActiveX, AJAX, (D)HTML, FLASH, Java, JavaScript, web
29 browser plug-in APIs (e.g., FireFox, Safari Plug-in, and/or the like APIs), and/or the
30 like. Web browsers and like information access tools may be integrated into PDAs,

1 cellular telephones, and/or other mobile devices. A Web browser may communicate to
2 and/or with other components in a component collection, including itself, and/or
3 facilities of the like. Most frequently, the Web browser communicates with information
4 servers, operating systems, integrated program components (e.g., plug-ins), and/or the
5 like; e.g., it may contain, communicate, generate, obtain, and/or provide program
6 component, system, user, and/or data communications, requests, and/or responses.
7 Also, in place of a Web browser and information server, a combined application may be
8 developed to perform similar operations of both. The combined application would
9 similarly affect the obtaining and the provision of information to users, user agents,
10 and/or the like from the SMP enabled nodes. The combined application may be
11 nugatory on systems employing standard Web browsers.

12

Mail Server

13 **[00612]** A mail server component 9921 is a stored program component that is
14 executed by a CPU 9903. The mail server may be a conventional Internet mail server
15 such as, but not limited to sendmail, Microsoft Exchange, and/or the like. The mail
16 server may allow for the execution of program components through facilities such as
17 ASP, ActiveX, (ANSI) (Objective-) C (++), C# and/or .NET, CGI scripts, Java,
18 JavaScript, PERL, PHP, pipes, Python, WebObjects, and/or the like. The mail server
19 may support communications protocols such as, but not limited to: Internet message
20 access protocol (IMAP), Messaging Application Programming Interface
21 (MAPI)/Microsoft Exchange, post office protocol (POP3), simple mail transfer protocol
22 (SMTP), and/or the like. The mail server can route, forward, and process incoming and
23 outgoing mail messages that have been sent, relayed and/or otherwise traversing
24 through and/or to the SMP.

25 **[00613]** Access to the SMP mail may be achieved through a number of APIs offered
26 by the individual Web server components and/or the operating system.

27 **[00614]** Also, a mail server may contain, communicate, generate, obtain, and/or
28 provide program component, system, user, and/or data communications, requests,
29 information, and/or responses.

Mail Client

1
2 **[00615]** A mail client component 9922 is a stored program component that is
3 executed by a CPU 9903. The mail client may be a conventional mail viewing application
4 such as Apple Mail, Microsoft Entourage, Microsoft Outlook, Microsoft Outlook
5 Express, Mozilla, Thunderbird, and/or the like. Mail clients may support a number of
6 transfer protocols, such as: IMAP, Microsoft Exchange, POP3, SMTP, and/or the like. A
7 mail client may communicate to and/or with other components in a component
8 collection, including itself, and/or facilities of the like. Most frequently, the mail client
9 communicates with mail servers, operating systems, other mail clients, and/or the like;
10 e.g., it may contain, communicate, generate, obtain, and/or provide program
11 component, system, user, and/or data communications, requests, information, and/or
12 responses. Generally, the mail client provides a facility to compose and transmit
13 electronic mail messages.

Cryptographic Server

14
15 **[00616]** A cryptographic server component 9920 is a stored program component
16 that is executed by a CPU 9903, cryptographic processor 9926, cryptographic processor
17 interface 9927, cryptographic processor device 9928, and/or the like. Cryptographic
18 processor interfaces will allow for expedition of encryption and/or decryption requests
19 by the cryptographic component; however, the cryptographic component, alternatively,
20 may run on a conventional CPU. The cryptographic component allows for the
21 encryption and/or decryption of provided data. The cryptographic component allows for
22 both symmetric and asymmetric (e.g., Pretty Good Protection (PGP)) encryption and/or
23 decryption. The cryptographic component may employ cryptographic techniques such
24 as, but not limited to: digital certificates (e.g., X.509 authentication framework), digital
25 signatures, dual signatures, enveloping, password access protection, public key
26 management, and/or the like. The cryptographic component will facilitate numerous
27 (encryption and/or decryption) security protocols such as, but not limited to: checksum,
28 Data Encryption Standard (DES), Elliptical Curve Encryption (ECC), International Data
29 Encryption Algorithm (IDEA), Message Digest 5 (MD5, which is a one way hash
30 operation), passwords, Rivest Cipher (RC5), Rijndael, RSA (which is an Internet

1 encryption and authentication system that uses an algorithm developed in 1977 by Ron
2 Rivest, Adi Shamir, and Leonard Adleman), Secure Hash Algorithm (SHA), Secure
3 Socket Layer (SSL), Secure Hypertext Transfer Protocol (HTTPS), and/or the like.
4 Employing such encryption security protocols, the SMP may encrypt all incoming
5 and/or outgoing communications and may serve as node within a virtual private
6 network (VPN) with a wider communications network. The cryptographic component
7 facilitates the process of “security authorization” whereby access to a resource is
8 inhibited by a security protocol wherein the cryptographic component effects authorized
9 access to the secured resource. In addition, the cryptographic component may provide
10 unique identifiers of content, e.g., employing and MD5 hash to obtain a unique
11 signature for an digital audio file. A cryptographic component may communicate to
12 and/or with other components in a component collection, including itself, and/or
13 facilities of the like. The cryptographic component supports encryption schemes
14 allowing for the secure transmission of information across a communications network
15 to enable the SMP component to engage in secure transactions if so desired. The
16 cryptographic component facilitates the secure accessing of resources on the SMP and
17 facilitates the access of secured resources on remote systems; i.e., it may act as a client
18 and/or server of secured resources. Most frequently, the cryptographic component
19 communicates with information servers, operating systems, other program components,
20 and/or the like. The cryptographic component may contain, communicate, generate,
21 obtain, and/or provide program component, system, user, and/or data communications,
22 requests, and/or responses.

23

The SMP Database

24 **[00617]** The SMP database component 9919 may be embodied in a database and
25 its stored data. The database is a stored program component, which is executed by the
26 CPU; the stored program component portion configuring the CPU to process the stored
27 data. The database may be a conventional, fault tolerant, relational, scalable, secure
28 database such as Oracle or Sybase. Relational databases are an extension of a flat file.
29 Relational databases consist of a series of related tables. The tables are interconnected
30 via a key field. Use of the key field allows the combination of the tables by indexing

1 against the key field; i.e., the key fields act as dimensional pivot points for combining
2 information from various tables. Relationships generally identify links maintained
3 between tables by matching primary keys. Primary keys represent fields that uniquely
4 identify the rows of a table in a relational database. More precisely, they uniquely
5 identify rows of a table on the “one” side of a one-to-many relationship.

6 **[00618]** Alternatively, the SMP database may be implemented using various
7 standard data-structures, such as an array, hash, (linked) list, struct, structured text file
8 (e.g., XML), table, and/or the like. Such data-structures may be stored in memory
9 and/or in (structured) files. In another alternative, an object-oriented database may be
10 used, such as Frontier, ObjectStore, Poet, Zope, and/or the like. Object databases can
11 include a number of object collections that are grouped and/or linked together by
12 common attributes; they may be related to other object collections by some common
13 attributes. Object-oriented databases perform similarly to relational databases with the
14 exception that objects are not just pieces of data but may have other types of capabilities
15 encapsulated within a given object. If the SMP database is implemented as a data-
16 structure, the use of the SMP database 9919 may be integrated into another component
17 such as the SMP component 9935. Also, the database may be implemented as a mix of
18 data structures, objects, and relational structures. Databases may be consolidated
19 and/or distributed in countless variations through standard data processing techniques.
20 Portions of databases, e.g., tables, may be exported and/or imported and thus
21 decentralized and/or integrated.

22 **[00619]** In one embodiment, the database component 9919 includes several tables
23 9919a-u.

24 **[00620]** A *seeker_profiles table* 9919a may include fields such as, but not limited
25 to: user_ID, name, address, contact_info, preferences, friends, status, user_description,
26 attributes, experience_info_ID, path_ID(s), attribute_ID(s), and/or the like.

27 **[00621]** A *seeker_experience table* (aka “experience” or “resume” table) 9919b may
28 include fields such as, but not limited to: experience_info_ID, experience_item_ID(s),
29 education_item_ID(s), resume_data, skills, awards, honors, languages,
30 current_salary_preferences, user_ID(s), path_ID(s), and/or the like.

1 **[00622]** A *experience_item table* 9919c may include fields such as, but not limited
2 to: `experience_item_ID`, `institution_ID`, `job_title`, `job_description`, `job_dates`,
3 `job_salary`, `skills`, `awards`, `honors`, `satisfaction_ratings`, `state_ID`, `OC_code(s)`,
4 `attribute_ID(s)`, and/or the like.

5 **[00623]** A *education_item table* 9919d may include fields such as, but not limited
6 to: `education_item_ID`, `institution_ID`, `education_degree_subject_matter`,
7 `education_item_description`, `education_degree`, `education_item_dates`, `skills`, `awards`,
8 `honors`, `satisfaction_ratings`, `state_ID`, `attribute_ID(s)`, and/or the like.

9 **[00624]** A *state_structure table* 9919e may include fields such as, but not limited
10 to: `state_structure_ID`, `state_structure_data`, `state_ID(s)`, and/or the like.

11 **[00625]** A *states table* 9919f may include fields such as, but not limited to:
12 `state_ID`, `state_name`, `job_titles`, `topics`, `next_states_ID`, `previous_states_ID`,
13 `state_transition_probabilities`, `job_title_count`, `total_count`, `topic_count`,
14 `transition_weights`, `OC_code(s)`, `attribute_ID(s)`, `user_ID(s)`, and/or the like.

15 **[00626]** A *experience_structure table* 9919g may include fields such as, but not
16 limited to: `experience_structure_ID`, `experience_structure_data`, `OC_code(s)`, and/or
17 the like.

18 **[00627]** A *experiences table* (aka “OC” table) 9919h may include fields such as, but
19 not limited to: `OC_code`, `parent_OC_code`, `child_OC_code(s)`, `title(s)`,
20 `job_description(s)`, `educational_requirement(s)`, `experience_requirement(s)`,
21 `salary_range`, `tasks_work_activities`, `skills`, `category`, `keywords`, `related occupations`,
22 `state_ID(s)`, `attribute_ID(s)`, and/or the like.

23 **[00628]** An *attributes table* 9919i may include fields such as, but not limited to:
24 `attribute_ID`, `attribute_name`, `attribute_type`, `attribute_weight`, `attribute_keywords`,
25 `confidence_value`, `rating_value`, `comment`, `comment_thread_ID(s)`, `salary`,
26 `geographic_location`, `hours_of_work`, `vacation_days`, `benefits`,
27 `attribute_transition_value`, `attribute_transition_weight`, `education_level`, `degree`,
28 `years_of_experience`, `state_ID(s)`, `OC_code(s)`, `user_ID(s)`, and/or the like.

1 **[00629]** A *paths table* 9919j may include fields such as, but not limited to:
2 path_ID, state_path_sequence, state_ID(s), attribute_ID(s), user_ID(s),
3 attribute_key_terms, and/or the like.

4 **[00630]** A *templates table* 9919k may include fields such as, but not limited to:
5 template_ID, state_ID, job_ID, employer_ID, attribute_ID, template data, template
6 display name, template category (e.g., cover letter, resume, CV, etc.), template file
7 location, and/or the like.

8 **[00631]** A *job_listing table* 9919l may include fields such as, but not limited to:
9 job_listing_ID, institution_ID, job_title, job_description, educational_requirements,
10 experience_requirements, salary_range, tasks_work_activities, skills, category,
11 keywords, related occupations, OC_code, state_ID, attribute_ID(s), user_ID(s),
12 UI_ID(s), recruiter_ID, location, salary, and/or the like.

13 **[00632]** A *institution table* (aka “employer” table) 9919m may include fields such
14 as, but not limited to: institution_ID, name, address, contact_info, preferences, status,
15 industry_sector, description, experience_ID(s), template_ID(s), state_ID(s), attributes,
16 attribute_ID(s), and/or the like.

17 **[00633]** A *user table* 9919n includes fields such as, but not limited to: a userID,
18 screenName, address, social security number, e-mail address, education, job experience,
19 skills, references, honors and/or awards, publications, resume and/or CV, and/or the
20 like. The user table may support and/or track multiple entity accounts on a SMP.

21 **[00634]** An *offer table* 9919o includes fields such as, but not limited to: offerID,
22 offerDuration, targetDemographics, and/or the like.

23 **[00635]** A *content provider table* 9919p includes fields such as, but not limited to:
24 content provider ID, content provider name, AODSA module format restrictions,
25 AODSA module serving conditions, general content descriptors, provider content, web
26 user information, and/or the like.

27 **[00636]** A *base data entry table* 9919q includes fields such as, but not limited to:
28 sponsor ID, preferred content IDs, related base data entries, sponsor distribution

1 and/or advertisement subscription parameters, preferred distribution targets, and/or
2 the like.

3 **[00637]** A *generation rule set table* 9919r includes fields such as, but not limited
4 to: parsing priority, key term elements, location element, salary element, opportunity
5 type element, web user group labels, allowed template characteristic combinations,
6 restricted template characteristic combinations, and/or the like.

7 **[00638]** A *landing page table* 9919s includes fields such as, but not limited to: base
8 data entry, redacted base data entry, sponsor information, host data entity, additional
9 landing functionality, and/or the like.

10 **[00639]** A *web user table* 9919t includes fields such as, but not limited to: web user
11 ID, dynamic web user interaction data records, stored web user interaction data records,
12 content provider registration data, salary data, opportunity type element, performance
13 history, and/or the like.

14 **[00640]** An *Ad generation template table* 9919u includes fields such as, but not
15 limited to: Ad generation template ID, parsing priority, key term elements, location
16 element, salary element, opportunity type element, performance history, and/or the
17 like.

18 **[00641]** In one embodiment, the SMP database may interact with other database
19 systems. For example, employing a distributed database system, queries and data access
20 by search SMP component may treat the combination of the SMP database, an
21 integrated data security layer database as a single database entity.

22 **[00642]** In one embodiment, user programs may contain various user interface
23 primitives, which may serve to update the SMP. Also, various accounts may require
24 custom database tables depending upon the environments and the types of clients the
25 SMP may need to serve. It should be noted that any unique fields may be designated as a
26 key field throughout. In an alternative embodiment, these tables have been
27 decentralized into their own databases and their respective database controllers (i.e.,
28 individual database controllers for each of the above tables). Employing standard data
29 processing techniques, one may further distribute the databases over several computer
30 systemizations and/or storage devices. Similarly, configurations of the decentralized

1 database controllers may be varied by consolidating and/or distributing the various
2 database components 9919a-u. The SMP may be configured to keep track of various
3 settings, inputs, and parameters via database controllers.

4 **[00643]** The SMP database may communicate to and/or with other components in
5 a component collection, including itself, and/or facilities of the like. Most frequently, the
6 SMP database communicates with the SMP component, other program components,
7 and/or the like. The database may contain, retain, and provide information regarding
8 other nodes and data.

9

The SMPs

10 **[00644]** The SMP component 9935 is a stored program component that is executed
11 by a CPU. In one embodiment, the SMP component incorporates any and/or all
12 combinations of the aspects of the SMP that was discussed in the previous figures. As
13 such, the SMP affects accessing, obtaining and the provision of information, services,
14 transactions, and/or the like across various communications networks. The features and
15 embodiments of the SMP discussed herein increase network efficiency by reducing data
16 transfer requirements the use of more efficient data structures and mechanisms for their
17 transfer and storage. As a consequence, more data may be transferred in less time, and
18 latencies with regard to transactions, are also reduced. In many cases, such reduction in
19 storage, transfer time, bandwidth requirements, latencies, etc., will reduce the capacity
20 and structural infrastructure requirements to support the SMP's features and facilities,
21 and in many cases reduce the costs, energy consumption/requirements, and extend the
22 life of SMP's underlying infrastructure; this has the added benefit of making the SMP
23 more reliable. Similarly, many of the features and mechanisms are designed to be easier
24 for users to use and access, thereby broadening the audience that may enjoy/employ
25 and exploit the feature sets of the SMP; such ease of use also helps to increase the
26 reliability of the SMP. In addition, the feature sets include heightened security as noted
27 via the Cryptographic components 9920, 9926, 9928 and throughout, making access to
28 the features and data more reliable and secure

1 **[00645]** The SMP transforms platform join requests, social network info, and SMP
2 network info inputs via SMP components NJ, JIP, CIP, OP, CN-SGU and CN-UPSOG
3 into job info, candidate info, offer info, and social meetup info outputs.

4 **[00646]** The SMP component enabling access of information between nodes may
5 be developed by employing standard development tools and languages such as, but not
6 limited to: Apache components, Assembly, ActiveX, binary executables, (ANSI)
7 (Objective-) C (++), C# and/or .NET, database adapters, CGI scripts, Java, JavaScript,
8 mapping tools, procedural and object oriented development tools, PERL, PHP, Python,
9 shell scripts, SQL commands, web application server extensions, web development
10 environments and libraries (e.g., Microsoft's ActiveX; Adobe AIR, FLEX & FLASH;
11 AJAX; (D)HTML; Dojo, Java; JavaScript; jQuery(UI); MooTools; Prototype;
12 script.aculo.us; Simple Object Access Protocol (SOAP); SWFObject; Yahoo! User
13 Interface; and/or the like), WebObjects, and/or the like. In one embodiment, the SMP
14 server employs a cryptographic server to encrypt and decrypt communications. The
15 SMP component may communicate to and/or with other components in a component
16 collection, including itself, and/or facilities of the like. Most frequently, the SMP
17 component communicates with the SMP database, operating systems, other program
18 components, and/or the like. The SMP may contain, communicate, generate, obtain,
19 and/or provide program component, system, user, and/or data communications,
20 requests, and/or responses.

21

Distributed SMPs

22 **[00647]** The structure and/or operation of any of the SMP node controller
23 components may be combined, consolidated, and/or distributed in any number of ways
24 to facilitate development and/or deployment. Similarly, the component collection may
25 be combined in any number of ways to facilitate deployment and/or development. To
26 accomplish this, one may integrate the components into a common code base or in a
27 facility that can dynamically load the components on demand in an integrated fashion.

28 **[00648]** The component collection may be consolidated and/or distributed in
29 countless variations through standard data processing and/or development techniques.
30 Multiple instances of any one of the program components in the program component

1 collection may be instantiated on a single node, and/or across numerous nodes to
2 improve performance through load-balancing and/or data-processing techniques.
3 Furthermore, single instances may also be distributed across multiple controllers
4 and/or storage devices; e.g., databases. All program component instances and
5 controllers working in concert may do so through standard data processing
6 communication techniques.

7 **[00649]** The configuration of the SMP controller will depend on the context of
8 system deployment. Factors such as, but not limited to, the budget, capacity, location,
9 and/or use of the underlying hardware resources may affect deployment requirements
10 and configuration. Regardless of if the configuration results in more consolidated
11 and/or integrated program components, results in a more distributed series of program
12 components, and/or results in some combination between a consolidated and
13 distributed configuration, data may be communicated, obtained, and/or provided.
14 Instances of components consolidated into a common code base from the program
15 component collection may communicate, obtain, and/or provide data. This may be
16 accomplished through intra-application data processing communication techniques
17 such as, but not limited to: data referencing (e.g., pointers), internal messaging, object
18 instance variable communication, shared memory space, variable passing, and/or the
19 like.

20 **[00650]** If component collection components are discrete, separate, and/or
21 external to one another, then communicating, obtaining, and/or providing data with
22 and/or to other component components may be accomplished through inter-application
23 data processing communication techniques such as, but not limited to: Application
24 Program Interfaces (API) information passage; (distributed) Component Object Model
25 ((D)COM), (Distributed) Object Linking and Embedding ((D)OLE), and/or the like),
26 Common Object Request Broker Architecture (CORBA), Jini local and remote
27 application program interfaces, JavaScript Object Notation (JSON), Remote Method
28 Invocation (RMI), SOAP, process pipes, shared files, and/or the like. Messages sent
29 between discrete component components for inter-application communication or within
30 memory spaces of a singular component for intra-application communication may be
31 facilitated through the creation and parsing of a grammar. A grammar may be

1 developed by using development tools such as lex, yacc, XML, and/or the like, which
2 allow for grammar generation and parsing capabilities, which in turn may form the basis
3 of communication messages within and between components.

4 **[00651]** For example, a grammar may be arranged to recognize the tokens of an
5 HTTP post command, e.g.:

```
6           w3c -post http://... Value1
```

7
8 **[00652]** where Value1 is discerned as being a parameter because “http://” is part of
9 the grammar syntax, and what follows is considered part of the post value. Similarly,
10 with such a grammar, a variable “Value1” may be inserted into an “http://” post
11 command and then sent. The grammar syntax itself may be presented as structured data
12 that is interpreted and/or otherwise used to generate the parsing mechanism (e.g., a
13 syntax description text file as processed by lex, yacc, etc.). Also, once the parsing
14 mechanism is generated and/or instantiated, it itself may process and/or parse
15 structured data such as, but not limited to: character (e.g., tab) delineated text, HTML,
16 structured text streams, XML, and/or the like structured data. In another embodiment,
17 inter-application data processing protocols themselves may have integrated and/or
18 readily available parsers (e.g., JSON, SOAP, and/or like parsers) that may be employed
19 to parse (e.g., communications) data. Further, the parsing grammar may be used
20 beyond message parsing, but may also be used to parse: databases, data collections, data
21 stores, structured data, and/or the like. Again, the desired configuration will depend
22 upon the context, environment, and requirements of system deployment.

23 **[00653]** For example, in some implementations, the SMP controller may be
24 executing a PHP script implementing a Secure Sockets Layer (“SSL”) socket server via
25 the information sherver, which listens to incoming communications on a server port to
26 which a client may send data, e.g., data encoded in JSON format. Upon identifying an
27 incoming communication, the PHP script may read the incoming message from the
28 client device, parse the received JSON-encoded text data to extract information from the
29 JSON-encoded text data into PHP script variables, and store the data (e.g., client
30 identifying information, etc.) and/or extracted information in a relational database
31 accessible using the Structured Query Language (“SQL”). An exemplary listing, written

1 substantially in the form of PHP/SQL commands, to accept JSON-encoded input data
 2 from a client device via a SSL connection, parse the data to extract variables, and store
 3 the data to a database, is provided below:

```

4  <?PHP
5  header('Content-Type: text/plain');
6
7  // set ip address and port to listen to for incoming data
8  $address = '192.168.0.100';
9  $port = 255;
10
11 // create a server-side SSL socket, listen for/accept incoming communication
12 $sock = socket_create(AF_INET, SOCK_STREAM, 0);
13 socket_bind($sock, $address, $port) or die('Could not bind to address');
14 socket_listen($sock);
15 $client = socket_accept($sock);
16
17 // read input data from client device in 1024 byte blocks until end of message
18 do {
19     $input = "";
20     $input = socket_read($client, 1024);
21     $data .= $input;
22 } while($input != "");
23
24 // parse data to extract variables
25 $obj = json_decode($data, true);
26
27 // store input data in a database
28 mysql_connect("201.408.185.132", $DBserver, $password); // access database server
29 mysql_select("CLIENT_DB.SQL"); // select database to append
30 mysql_query("INSERT INTO UserTable (transmission)
31 VALUES ($data)"); // add data to UserTable table in a CLIENT database
32 mysql_close("CLIENT_DB.SQL"); // close connection to database
33 ?>
34

```

35 **[00654]** Also, the following resources may be used to provide example
 36 embodiments regarding SOAP parser implementation:

```

37 http://www.xav.com/perl/site/lib/SOAP/Parser.html
38 http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm
39 .IBMDI.doc/referenceguide295.htm
40

```

41 **[00655]** and other parser implementations:

```

42 http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm
43 .IBMDI.doc/referenceguide259.htm
44

```

45 **[00656]** all of which are hereby expressly incorporated by reference.

Application Tracking and Semantic Search

1
2 **[00657]** Figure 100 discloses an embodiment of a process tracking and
3 management environment aspect of the disclosure. The Process Tracking and
4 Management ("PTaM") Environment enables an applicant to initiate, manage, and
5 monitor processes of interest to the applicant, for example, via an SMP social network
6 application. The user interface for the PTaM Environment may be a PTaM Viewer
7 10010. The PTaM Viewer can be implemented using a variety of software development
8 techniques know in the art. For example, the PTaM Viewer can be a stand-alone
9 application, a social network application, a web-based application, a mobile device
10 application, and/or integrated into another piece of software.

11 **[00658]** The PTaM Viewer allows the applicant to access the functions of the PTaM
12 Environment, such as integrating input data 10020 or conducting process tracking. The
13 PTaM Viewer may connect to a network 10040 to receive information concerning the
14 processes being tracked, such as a status update for a particular process.

15 **[00659]** The PTaM Viewer also interacts with the PTaM Plug-In 10050. The PTaM
16 Plug-In provides a mechanism for information to be transferred to and from the PTaM
17 Viewer, e.g., via SMP functionality and/or social and/or professional networking
18 systems. For example, a monitored process might be generated through the use of
19 another application (i.e., a host application) with which the PTaM Plug-In is integrated
20 in order to provide added functionality. In this way, when a new process is started in
21 the host application the PTaM Plug-In provides a user interface and communications
22 infrastructure to communicate the relevant information about the new process to the
23 PTaM Plug-In. The PTaM Plug-In can similarly make relevant information from the
24 PTaM Viewer, such as the input data, accessible to the host program.

25 **[00660]** Figure 101 depicts an embodiment of the disclosed system specifically used
26 to implement a Job Application Tracking and Management ("JATaM") Environment.
27 The JATaM Environment comprises a Job Tracking and Management Core ("JTM
28 Core") 10110 and a Job Listing Tracking Extension 10150. The JTM Core serves to
29 provide a primary user interface and coordination point for an applicant's use of the
30 JATaM Environment. The JATaM Environment can be implemented as an application

1 in a social network environment, as a mobile application, a stand-alone application
2 and/or a web based application. The JATaM Environment may be implemented using a
3 variety of software development techniques as required by the software and/or
4 hardware platform it runs upon, such as, for example, a particular social network, a
5 personal digital assistant or smart phone, tablet computer, etc.

6 **[00661]** An applicant may have a unique instance of the JATaM Environment for
7 his or her use, such as by having one local version of the JATaM Environment on the
8 applicant's smart phone. Applicants might alternately be provided a unique instance of
9 the JATaM Environment through the use of user accounts with usernames and
10 passwords. A unique instance of the JATaM Environment stores relevant personal
11 information about the applicant and his or her job search. In particular, the JTM Core
12 10110, for example, might take as input the applicant's resume and/or cover letter
13 10120, and/or profile data from an applicant's social network profile, job applicant
14 profile, professional network profile, etc. The profile data, resume and/or cover letter
15 can then be made available to aid in filing job applications for the applicant and/or in
16 finding job listings suitable to the interests and abilities of the applicant. The JTM Core
17 may, in some embodiments, keep a copy of the applicant's resume and cover letter or a
18 symbolic link to its location, and/or a set of profile data that could be used to populate a
19 resume and/or job application. In a web application embodiment the resume and cover
20 letter could be up-loaded to the web application's host server.

21 **[00662]** Once accessed by the JTM Core, the profile, resume and/or cover letter
22 information can be parsed by the JTM Core to extract relevant information about the
23 applicant, such as, the applicant's name, address, phone number, email address,
24 educational background, title, work history, social network data, connection
25 information, etc. This parsed information might be presented to the applicant for
26 confirmation and then used by the JTM Core to create a new profile associated with the
27 applicant. In a further embodiment, the applicant could be prompted to provide
28 additional information to expand the profile created by the JTM Core. Some profile
29 information could be extracted from a social network and used to populate a
30 professional network profile, separate and apart from the social network profile.
31 Alternately, elements of the applicant's profile could be entered by hand or imported

1 from some other source. In some embodiments, key terms associated with the applicant
2 may be extracted from a social network profile and/or resume during the
3 creation/upload process. The key terms may be used for as search terms in a job
4 matching application that will be described in greater detail below.

5 **[00663]** In the job application process, information about job listings may be
6 derived from multiple sources, *e. g.*, various job listing services and various employer
7 listings. Typically, these job listings will come to the attention of the applicant through
8 searches on the World Wide Web, or a similar information-gathering environment, or
9 they may be provided to the applicant based on cookies, profile information, etc. The
10 Job Listing Tracking Extension can be embodied by social match platform app in a
11 social network, on a mobile device, and/or as a plug-in, for example, with a browser
12 acting as the host application for the Job Listing Tracking Extension. Alternately, it
13 could be a separate utility, a network, *e.g.* web, interface, or an integrated function of the
14 host application.

15 **[00664]** In the embodiment of Figure 101, the Job Listing Tracking Extension
16 10150 integrates with the applicant's browser 10160 to provide a mechanism to transfer
17 information to and from the JATaM Environment and the browser. In order to allow
18 the JATaM Environment to record and transfer information when the JTM Core 10110
19 is not actively running, a background process 10111 may be provided to provide the Job
20 Listing Tracking Extension 10150 a connection interface. The background process could
21 be embodied, for example, as a daemon or windows background process. In the
22 alternative, the information used by the system can be stored on a network accessible
23 data storage device that is accessible by both the Job Listing Tracking Extension and the
24 JTM Core. The Job Listing Tracking Extension 10150 integrates with its host
25 application browser 10160, which connects to the internet 10140. One implementation
26 of the Job Listing Tracking Extension 10150 may be a web browser toolbar, while an
27 alternative extension embodiment might incorporate menu commands for
28 implementing Job Listing Tracking Extension functions to the host application browser.

29 **[00665]** Figure 102A discloses a flow diagram for a job listing search aspect of one
30 embodiment of the present disclosure. The method disclosed, which in some
31 embodiments is implemented on a processor and executed by the Job Listing Tracking

1 Extension 10150 and/or the JTM Core 10110, allows for receiving input 10205 relevant
2 to a job listing search such as, by way of non-limiting example, location, skill(s) and/or
3 education, experience level and compensation. In some embodiments, the input 10205
4 may come directly from an applicant, such as an applicant's entered job parameters, an
5 applicant's response to job related queries, and/or a submitted resume. In another
6 embodiment, the input 10205 could include information not specific to a particular
7 applicant, such as a job description or listing.

8 **[00666]** Relevant data is extracted 10215 from the received input, for example, if
9 the input were a resume, the extraction could include parsing and analyzing the resume
10 to distill the key data relevant to the job listing search. The relevant data is used to build
11 a search profile 10225 that may then be utilized, either in whole or in part, in generating
12 and conducting a job listing search or searches 10235. The results of the search or
13 searches are processed 10245, which in some embodiments may include a further
14 analysis and/or refinement of the search results, and from the processing, relevant
15 results (i.e., a plurality of job listings corresponding to the input 10205) are identified
16 10255.

17 **[00667]** In a further embodiment, shown in Figure 102B, the above method may be
18 iteratively refined. As in the previous figure, an initial search profile 10225 is utilized in
19 conducting job listing searches 10235, the results of the searches are processed 10245,
20 and relevant job listings are identified 10255. The identified job listings are then
21 presented to an applicant who may evaluate 10265 the job listings based on any number
22 of user-rating metrics. For example, the user may rate the job listings according to how
23 well suited the listings are to the applicant's employment interests, the listings meet the
24 applicant's qualifications or desired income goals. The applicant's feedback and/or
25 evaluation 10265 may then be used to refine the search profile 10275, which may be
26 used in place or in addition to the initial search profile 10225 in conducting subsequent
27 searches of job listings 10235. The above method may be iterated multiple times to
28 further refine the job listings identification of job listings relevant to an applicant.

29 **[00668]** Figure 103A discloses a flow diagram for data entry correlation according
30 to an embodiment of the disclosure. The method disclosed allows for receiving and
31 processing applicant designated base data entries and correlating the base data entries

1 with supplementary data entries, as well as generating and outputting a data entry
2 correlation report. In some embodiments, the cross correlation process 10310 is
3 implemented on a processor by the Job Listing Tracking Extension 10150 and/or the
4 JTM Core 10110. The data entry correlation report may be utilized in searching and
5 identifying supplemental data entries (e.g., job listings) matching an applicant. An
6 applicant designated base data entry is received and processed 10320 to determine base
7 data entry metrics 10330. User correlation parameters are retrieved 10340 that match
8 the base data entry metrics 10330, wherein matching user correlation parameters
9 correspond to the base data entry metrics 10330. From the retrieved user correlation
10 parameters 10340, a data entry correlation report is generated 10350. Some
11 embodiments may output the data entry correlation report 10360, and in a further
12 embodiment, the data entry correlation report may be associated with the applicant's
13 profile. In some embodiments, user correlation parameters may be retrieved from
14 internal and/or external data bases.

15 **[00669]** In some embodiments, the base data entry and/or user correlation
16 parameters may be direct input, such as an applicant's response(s) to one or more brief
17 and/or in-depth surveys as discussed in Figure 103B. Survey methodologies may
18 include multiple choice, short answer, rating, ranking, free response and/or the like. In
19 one embodiment, the surveys are directly related to job finding. In a further
20 embodiment, the surveys include additional matter (e.g., personality tests) that, after
21 being analyzed, may be useful in finding job listings that are especially well matched to
22 an applicant.

23 **[00670]** In one embodiment, the relevant user correlation parameters may include
24 an applicant's historical information. In some embodiments, the historical information
25 may include an applicant's search for job listings, viewing of job listings, responses to
26 job listings, aspects of the applicant's profile, application submission to job listings,
27 and/or saved job listings. In another embodiment, the historical information includes
28 the applicant's web site viewing and/or web search history.

29 **[00671]** In one implementation, shown in Figure 103B, the applicant designated
30 base data entry or entries may include an applicant's resume 10321, cover letter 10322,
31 survey response(s) 10323 and/or profile 10324. The base data entry or entries are

1 received and processed to extract base data entry metrics 10330. User correlation
2 parameters including the applicant's job search history 10341 and/or internet/web
3 browsing history 10342 are retrieved and correlated with the base data entry metrics
4 10330 and a data entry correlation report is generated 10350. In a further embodiment,
5 the Job Listing Tracking Extension and/or the JTM Core utilizes a data entry correlation
6 rating module for an applicant to refine and target searches for job listings matching the
7 applicant.

8 **[00672]** For example, in one embodiment, the analysis of a particular applicant's
9 base data entry (in this case the applicant's resume) may yield base data entry
10 correlation metrics that correspond to skills, education, work history and other details
11 relevant to software engineering. User correlation metrics corresponding to the
12 applicant (in this case the job search terms used by the applicant on a job listing site,
13 such as job locations searched) are retrieved and correlated to the base data entry
14 metrics and a data entry correlation report is generated. In this case, the metrics may
15 indicate that the applicant has a certain level of experience in software engineering and
16 has previously worked in the southwest (from analysis of the base data entry), but has
17 been searching for jobs in the northwest (from the user correlation parameters). This
18 data entry correlation report may then be utilized to better target and serve the
19 applicant's interest by, for example, providing matching software engineering jobs in the
20 northwest.

21 **[00673]** In another embodiment, the base data entry may be selected by the
22 applicant, but is not necessarily applicant specific, including but not limited to a job
23 listing. For example, if a job listing is the base data entry, the job listing would be
24 processed to extract the relevant base data entry metrics. In some embodiments, the
25 correlation metrics may include the metrics mentioned above, and may be extracted
26 from the base data entries previously listed (e.g., resumes, cover letters, survey
27 responses, profiles). In one embodiment, the data entry correlation report may be
28 useful in finding job listings similar to the job listing used as the base data entry that are
29 especially well suited to an applicant.

30 **[00674]** For example, for the case where a job listing is the base data entry, the
31 correlation metrics may be based on the applicant's web or internet browsing history. If

1 the job listing selected is for a general software engineer with C/C++ programming
2 experience, and the applicant's browsing history shows indicates an interest in video
3 game development as well as a particular geographic region, the data entry correlation
4 report would reflect the potential correlation. The data entry correlation metrics could
5 then be used in finding and/or evaluating job listings for the applicant, i.e., a job listing
6 for a video game programmer requiring C++ programming abilities in applicant's
7 geographic region of interest would be identified as being a good match for the
8 applicant.

9 **[00675]** Figure 3C illustrates one embodiment in which the cross correlation
10 process 10310 is incorporated in the listing search method. As detailed in the above
11 discussions of Figures 102A and 102B, the listing search method may receive input
12 10205 relevant to searching job listings and extract relevant data 10215 from the
13 received input. The extracted data may then be used as the applicant designated base
14 data entry in the cross correlation process 10310 and the resulting data entry correlation
15 report 10350 is utilized in constructing the search profiles 10225 used to conduct job
16 listing searches 10235. As before, the results of the searches are processed 10245,
17 relevant job listings are identified 10255, and the identified job listings are presented to
18 the applicant for evaluation 10265. In some embodiments the applicant's evaluation
19 10265 may be incorporated into the cross correlation process 10310 to refine the
20 correlation of the base data entry metrics with the supplemental data entries and in a
21 further embodiment, is process is iterated to progressively enhance cross correlation
22 and the job listing search.

23 **[00676]** One implementation of the Job Listing Tracking Extension 10150 may be a
24 web browser toolbar. An implementation of a Job Listing Tracking Extension toolbar
25 10400 is shown in Figure 104A. In operation, the Job Listing Tracking Extension
26 toolbar of Figure 104A provides user accessible controls to implement job application
27 tracking functions and retrieve information from the JATaM Environment. For
28 example, if the applicant encounters a web page containing a job listing that the
29 applicant is interested in, the Save Job control 10415 can be activated. Activating the
30 Save Job control will cause the Job Listing Tracking Extension to transfer information
31 about the job listing to the JTM Core where it is stored or to a shared data repository

1 accessible by the JTM Core. In particular, the JTM Core might save the URL of the web
2 page containing the job listing. In the alternative, or in addition, information about the
3 job might be extracted from the web page and saved in a database accessible by JTM
4 Core and the Job Tracking extension.

5 **[00677]** The Job Listing Tracking Extension can also retrieve information from the
6 JTM Core for use in its host program (e.g., a web browser). For example, if the
7 applicant discovers a listing for a job of interest while browsing, he or she can use the
8 Apply To Job control 10425, to initiate the application process. The Apply To Job
9 control can display a menu with a number of different options for application purposes.
10 For example, if the on-line job application has an on-line web form, selecting the Apply
11 To Job option can use information from the JTM Core to auto-populate the web form. If
12 the job listing provides an email address for application submissions, the Choose
13 Resume And Apply option can open the applicant's email client and format an email
14 with the applicant is cover letter content in the body of the email and the applicant's
15 resume attached to the email. In the alternative, the Apply To Job control might
16 generate a cover letter document for transmission to the prospective employer. The
17 Apply To Job control can also be configured to provide a one-click apply feature. For
18 example, if the applicant has provided sufficient information to the system and the
19 currently viewed job listing is in a known format. The process could be configured to
20 effectuate application with only one user action, such as clicking on an automatic apply
21 button.

22 **[00678]** In some embodiments, the Job Listing Tracking Extension may also
23 submit information to the JTM Core from its host program. For example, if the
24 applicant is browsing a web site, particularly a web site listing jobs, the Job Listing
25 Tracking Extension can communicate this information to the JTM Core, which may
26 store the information and in a further embodiment, use it to modify the applicant's
27 profile. For example, if an applicant frequently goes to a particular company's job site,
28 that information could be sent from the Job Listing Tracking Extension to the JTM
29 Core, which could indicate the applicant's interest in employment at the particular
30 company.

1 **[00679]** The Figure 104A embodiment of the Job Listing Tracking Extension also
2 includes a Match control 10435 that provides job listing search assistance actions. For
3 example, in one embodiment, if the applicant discovers a job listing of interest, the
4 Show Jobs Matching Current 10436 option can be activated. In a further embodiment,
5 the Show Jobs Matching Current may be provided to the user as a one-click function. In
6 one embodiment, the Job Listing Tracking Extension will then activate processes to
7 analyze the current page and extract relevant job description terms. The Job Listing
8 Tracking Extension uses the extracted terms to initiate a search for similar jobs, the
9 results of which are provided to the applicant, for example, jobs from the same
10 employer, same area, same job description, or a combination of these. In a further
11 embodiment, the identification of similar jobs can be enhanced if express categorization
12 information is provided on the current page in a way that is easily manipulated by the
13 Job Listing Tracking Extension. One such mechanism would be to format the data
14 using an XML markup and associated schema that provides tags identifying the relevant
15 information.

16 **[00680]** Another option provided by the Match control 10435 in some
17 embodiments is Show Jobs Matching Saved 10437 option, which allows the applicant to
18 find jobs similar to jobs previously saved using the Save Job control 10415. The Job
19 Listing Tracking Extension will then activate processes to analyze the plurality of saved
20 jobs and extract the common and relevant job description terms. The Job Listing
21 Tracking Extension uses the extracted terms to initiate a search for similar jobs, the
22 results of which are provided to the applicant, by way of non-limiting example, jobs
23 from the same area, jobs with the same skills and/or educational requirements, same
24 job description, or a combination of these.

25 **[00681]** In another embodiment the Match control 10435 provides an option to
26 Score This Job 10438, which in some embodiments may be provided as a one-click
27 function. In one embodiment, shown in Figure 4B, when an applicant selects the Score
28 This Job option, the Job Listing Tracking Extension and/or JTM Core receive(s) the
29 score request 10410, parses and extracts the job listing data on the current page 10420
30 and retrieves analysis criteria from the applicant's profile 10430. The Job Listing
31 Tracking Extension and/or JTM Core analyzes and compares 10440 the extracted job

1 listing data to the criteria from the applicant's profile. For example, in the case of an
2 online job listing, contents of the web page (e.g., HTML page) are parsed and the parsed
3 contents of the job listing are compared to criteria from the applicant's profile and/or
4 resume. The correspondence between the job listing and the applicant's profile can then
5 be used to generate a score 10450 (a number or other indicia) of the job's relative
6 merits. For example, in one embodiment, if the job listing and the applicant's profile
7 have matching data elements such as skills, education, location, experience and the like,
8 such correspondence would be reflected in the score.

9 **[00682]** In one embodiment, the results of selecting Score This Job 10438 option
10 can be provided to the applicant via the Job Listing Tracking Extension toolbar 10400,
11 as a number or some other relative indicia of the job's suitability for the applicant (e.g.,
12 the job listing is an 86% match for the applicant). The applicant can score other job
13 listings and use the resulting indicia to compare different jobs and/or further refine the
14 applicant's profile. In some embodiments, the JTM Core may further enhance the
15 process of creating the score by receiving, storing and analyzing the applicant's goals,
16 criteria, and preferences. Some embodiments enhance the scoring process by analyzing
17 aspects of an applicants browsing or surfing history, such as job listings viewed as well
18 as companies sites, news sites and retailer sites visited. In one embodiment, a data
19 entry correlation report related to an applicant may be utilized in the scoring process.

20 **[00683]** This information could then be provided to the Job Listing Tracking
21 Extension and taken into account when performing the job scoring routine. In one
22 embodiment, the scoring process is iterative and may be refined by receiving additional
23 information and/or feedback from the applicant. For example, based on the applicant's
24 interests, the applicant may raise or lower the provided score of a particular job, and
25 this adjustment may then be incorporated when performing the job scoring routine. In
26 terms of implementation, the actual job scoring processes could reside in the JTM Core
27 with the Job Listing Tracking Extension passing the JTM Core the relevant data and the
28 JTM Core providing the score back to the Job Listing Tracking Extension and/or storing
29 for later reference by the applicant. The function could also be implemented on a
30 remote server that processes the job scoring request.

1 **[00684]** Another option provided by the Match control 10435 is Show Jobs
2 Matching Me 10439, shown in Figure 104A. Like the Score This Job option, the Show
3 Jobs Matching Me option uses information about the applicant indeed much of the
4 information used may be identical. However, rather than using the applicant's
5 information to score a pre-determined job listing, the applicant's information is used to
6 find job listings. In other words, this option finds high scoring jobs, and in one
7 embodiment, jobs scoring over a certain threshold (e.g., jobs scoring 90 or more out of
8 100). The applicant's information is used to create a search query or series of search
9 queries that will locate job listings relevant to the applicant. In one embodiment, the
10 Show Jobs Matching Me option is provided to the user as a one-click function. The
11 Show Jobs Matching Me 10439 option can cause searches to be performed upon
12 multiple different job listing databases and/or via a general internet search. The search
13 function can be implemented by either the Job Listing Tracking Extension or the JTM
14 Core.

15 **[00685]** The Figure 104A embodiment of the Job Listing Tracking Extension also
16 includes a Rate Job control 10445 that facilitates user feedback and response actions.
17 For example, if an applicant discovers a job listing of interest, the Rate Job control
18 10445 option can be activated allowing the applicant to submit feedback in the form of
19 some relative indicia, such as a "star" rating, indicating the applicant's interest in the
20 listing and/or particular aspects of the listing. The applicant can rate multiple job
21 listings and use the resulting indicia to compare different jobs and in a further
22 embodiment, the JTM Core may receive, store and analyze the applicant's ratings of job
23 listings to further refine the applicant's profile. In one embodiment, the job listing
24 ratings provided by an applicant may be incorporated in the generation of an associated
25 data entry correlation report.

26 **[00686]** Figure 4C shows a particular implementation of a Job Tracking And
27 Management Extension as a web browser toolbar 10400. This implementation includes
28 a key word search jobs interface 10405, a save jobs interface 10415, an apply to job
29 interface 10425 and a match job interface 10435. As shown in the figure, the save job
30 interface 10415 provides a drop down list of previously saved jobs. Also, the apply to job
31 interface 10425 allows the applicant to choose a resume and apply to a job. The resumes

1 that the applicant chooses could be stored locally on the applicant's machine or could be
2 stored on a network server. The create apply to job interface also includes a create
3 resume and create cover letter options, which allow the applicant to create resumes and
4 cover letters tailored to the job for which they are applying. The match user interface
5 10435 allows the applicant to perform a search for jobs matching the applicant, score
6 the job currently being viewed and search jobs similar to the current job.

7 **[00687]** In another embodiment, the Job Listing Tracking Extension and/or the
8 JTM Core may continue to search for matching job listings and notify an applicant if a
9 job listing is found that is especially well matched to the applicant (e.g., the job listing
10 represents the applicant's ideal job). Similarly, an applicant may be notified if a
11 particular listing has an especially high score according to the applicant's profile. In
12 some embodiments the system keeps searching for matching job listings for the
13 applicant even if the applicant has stopped actively looking for a job. In one
14 embodiment, the notification may be provided to an applicant regardless of whether or
15 not the applicant is currently searching for a job. For example, take an applicant who
16 had previously searched for jobs but was unable to find a job listing that was a perfect
17 match. The system could continue searching for matching listings, and if at some future
18 time the system found a job listing that was especially well-suited to the applicant
19 and/or the applicant's profile, the system would notify the applicant that a good match
20 had been found.

21 **[00688]** Figures 105A-105B provide flow diagrams representing overviews of
22 methods for dynamically providing matched job listings to an applicant in one
23 embodiment. In some embodiments, the methods disclosed may be implemented on a
24 processor by the Job Listing Tracking Extension 10150 and/or the JTM Core 10110, and
25 in further embodiments may utilize the Job Listing Tracking Extension toolbar 10400 as
26 disclosed in the above discussion of Figure 104A. In one embodiment, shown in Figure
27 105A, via the described method an applicant may browse job listings 10500 and view
28 10505, save 10515, rate 10545 and/or submit an application 10525 to job listing. When
29 an applicant views a job listing 10505, the listing, key terms from the listing, and the
30 amount of time it was viewed by the applicant are retained. In a further embodiment,
31 when saving 10515, rating 10545 and/or submitting an application 10525 to job listing,

1 the applicant may utilize the Save Job control 10415, Rate Job control 10435, and Apply
2 To Job control 10425 respectively, as disclosed in Figure 104A.

3 **[00689]** Each user initiated action yields a respective action group of listings from
4 which relevant data may be extracted 10510. The extracted data from individual listings
5 may be compared with extracted data from other listings in each group and/or, across
6 groups, to identify key common/distinguishing job listing characteristics 10520. In a
7 further embodiment, comparisons within and across sub-groupings, such as rated job
8 listings sub-grouped according to ratings or viewed job listings sub-grouped according
9 to view times, may be utilized in identifying key characteristics. The key characteristics
10 identified may then be used to build a search profile 10530 from which job listings
11 searches are conducted 10540. The job listings resulting from the searches 10550 may
12 then be presented to the applicant for browsing 10500, and in a further embodiment,
13 the process repeated to provide the applicant with progressively more relevant job
14 listings. In a further embodiment, the method disclosed is used to dynamically update
15 the job listings browsed by an applicant.

16 **[00690]** Figure 105B provides an overview of the group comparison process for one
17 embodiment. Rated job listings 10545 for an applicant may have the relevant data
18 extracted 10510, wherein the extracted data is associated with the respective individual
19 job listing. The rated job listings and associated data are grouped according to applicant
20 designated rating 10511: Group 5 for job listing the applicant has given a 5 star rating,
21 Group 4 for job listings the applicant has given a 4 star rating, and Group 3 for job
22 listings the applicant has given a 3 star rating. The extracted data from individual
23 listings is then analyzed within each group 10512 and/or between groups 10513 in order
24 to identify key characteristics 10520, which may be used to build a search profile. Other
25 embodiments may use the same or similar method in processing groups and/or
26 subgroups of job listings.

27 **[00691]** Figures 106A, 106B and 106C illustrate flow diagrams for the match
28 screening aspect of the matching process associated with some embodiments of the
29 disclosure. Certain embodiments may use these or similar processes in scoring jobs for
30 an applicant. The input for the match screening process is a Match request 10660 for a
31 plurality of job listings. The Match criteria 10661 for the job search may come from a

1 number of sources including but not limited to the following: analysis of the applicant's
2 profile, applicant's resume 10120, a particular job listing selected by the applicant, a
3 data entry correlation report, analysis of the plurality of job listings saved by the
4 applicant, job listings applied to by the applicant, analysis of the applicants browsing
5 history, applicant feedback and/or the like.

6 **[00692]** Next an iterative match screening of job listings is performed. The queries
7 discussed can be performed through a variety of methods. In some embodiments, the
8 iterative match screening begins with a primary criteria matching 10662 of the job
9 listings to obtain a first pool of potential matches for the applicant. The primary criteria
10 match screening 10662 uses match criteria that are essential to finding an appropriate
11 match for the applicant, such as skills and/or qualifications required. In some
12 embodiments, required skills and qualifications are fundamental criteria and provide a
13 solution for narrowing down the list of jobs (i.e., the job at least requires skills and
14 qualifications similar to the applicant's skills and qualifications).

15 **[00693]** In one embodiment, aspects of a data entry correlation report related to
16 an applicant may be included in the primary criteria and/or used in selecting primary
17 criteria. In one embodiment, the applicant may also be allowed to indicate required
18 criteria. For example, if an applicant is interested in finding a job with a particular
19 schedule, salary and/or location, the applicant might designate one of these parameters
20 as required criteria. Any such identified required criteria would be added to the primary
21 criteria matching 10662.

22 **[00694]** The results of the primary criteria matching 10662 are passed to a decision
23 block 10663 where a check is made to determine if any matching job listings were
24 identified. If no job listings meeting the primary criteria were found, a "no matches
25 found" error 10664 is generated and communicated. The criteria may then be adjusted
26 10671 to update the primary match criteria 10661 for the screen or start a new Match
27 request 10660, and in a further embodiment, update the search criteria. In a further
28 embodiment, the match screen and/or search criteria may be saved and the search be
29 re-run at a later date when the job listings information may have been supplemented or
30 modified. If the matching job listings are only a small number of entries (i.e., the
31 number of entries is $\leq k$, where k may be set by the applicant, the system, etc.), the

1 information for those job listings is output 10665 to the applicant. If the matching job
2 listings has many entries (number of entries > k), the process continues on to the
3 secondary criteria match screening.

4 **[00695]** A screen of the remaining job listings is performed to determine which job
5 listings meet one or more of the secondary criteria 10666. The results of this match
6 screen are passed to a decision block 10667. If none of the jobs meet any of the
7 secondary criteria, the job listings meeting the primary criteria may be reported to the
8 applicant 10668. If only a small number of job listings meet any of the secondary
9 criteria, the information for those job listings is output 10669 to the applicant. If many
10 job listings meet one or more of the secondary criteria further refinement of the screen
11 can be performed as follows.

12 **[00696]** In one embodiment, all job listings meeting at least a specified number of
13 secondary criteria can then be presented to the applicant for review 10670. This
14 approach works particularly well when the secondary criteria are not ordered in terms of
15 importance. In other words, if there are a number of secondary criteria without an
16 indication that some of those criteria are more preferred than others, it might be best to
17 give the applicant the information for all the job listings that meet their primary criteria
18 plus the specified number of their secondary criteria.

19 **[00697]** The applicant can then review the job listings to see how each meet their
20 secondary criteria and choose whichever they feel is the best match. In one
21 embodiment, the applicant may rate some or all of the provided job listings and/or
22 select job listings that the applicant feels are a good match and in a further embodiment,
23 the rated and/or selected job listings may be used to refine or adjust criteria 10671 and
24 update the match criteria 10661 used to select job listings for the applicant. For
25 example, if the applicant gives a low rating for a particular employer or type of job, the
26 criteria would be modified to screen against the particular employer or job type. In one
27 embodiment, an applicant's screen criteria and/or modified screen criteria may be
28 reflected in subsequent data entry correlation reports for an applicant.

29 **[00698]** The approach described above could be refined, as shown in Figure 106B,
30 by iterating over the job listings meeting one or more criteria to find the jobs meeting

1 the most categories of secondary criteria. This approach begins with the list of jobs
2 meeting one or more of the applicant's secondary criteria 10680. To determine if any of
3 the jobs meet more than one criteria the value n , starting at one, is incremented by
4 adding one to itself 10681. The list of jobs 10680 is screened for jobs that meet n criteria
5 10682 (i.e., two) during the first iteration of the process. The number of jobs remaining
6 after this matching is checked 10683. If no job listings remain, the list of jobs meeting
7 $n-1$ criteria is output 10684. If only a small number of job listings remain, those jobs'
8 information is output 10685. If more than a small number of job listings remain, a
9 determination is made to decide whether there are still more secondary applicant
10 criteria to check 10686. If there are not, the list of jobs meeting n criteria is output
11 10687. If there are secondary criteria remaining, the process increments n again 10681
12 and the filtering process continues.

13 **[00699]** The process as described above will result in job listings that meet the
14 highest number of the applicant's criteria. In other words, if twenty jobs meet any five
15 of the applicant's criteria, but none of the jobs meet six of the criteria, this match
16 process will identify those twenty job listings. This procedure, however, does not value
17 any of the secondary criteria more than the rest. Accordingly, if some of the applicant's
18 secondary criteria are more important than others a different process might be
19 advisable.

20 **[00700]** Figure 106C discloses a flow diagram for matching job listings such that
21 they are identified according to the importance of the secondary criteria. This flow
22 requires the secondary criteria to be ordered as to their importance, with the most
23 important criteria listed first, the second most important second and so on. This
24 ordering can either be identified by the applicant, the Job Listing Tracking Extension
25 10150, the JTM Core 10110, be inherently programmed into the system, generated using
26 other aspects of the disclosed embodiments, etc. For example, if the criteria for the
27 search were generated by analysis of the job listings saved by the applicant using the
28 save jobs interface 10415, the criteria could be ordered according to the which criteria
29 was most common to the saved job listings (e.g., if all the jobs require a particular skill,
30 that skill would be the first listed criteria; if most of the jobs were in a certain location,
31 that would be a subsequently listed criteria). In another embodiment, the criteria for

1 the search for a particular applicant may be identified and/or ordered according to a
2 data entry correlation report corresponding to the particular applicant.

3 **[00701]** The process begins with job listings meeting one or more of the secondary
4 criteria 10690. These are screened for job listings matching the (n)th most important
5 secondary criteria 10691, i.e., the first most important criteria in the first iteration of the
6 process, etc. The results derived from the match screen are checked to determine the
7 number of matching job listings 10692. If only a small number of job listings remain
8 after the matching, those job listings are output 10693. If no job listings remain after
9 the matching, a determination is made to discover whether there are other criteria to
10 check 10694. If not, the job listings found after the screen for matches for the (n-1)th
11 criteria are output 10695. If other criteria remain to be screened, the value of n is
12 incremented 10696. The job listings remaining after the previous matching are now
13 screened again at 10691 to determine if any of those job listings match the new (n)th
14 criteria and the iterative process continues. If multiple job listings are found to meet the
15 (n)th criteria at decision 10692, a determination 10697 is made to find whether there
16 are any criteria remaining to be screened. If no criteria remain, the job listings meeting
17 the (n)th criteria are output 10698. If other criteria remain to be screened, the value of
18 n is incremented 10699. The job listings remaining after the previous matching are now
19 screened again at 10691 to determine if any of those job listings match the new (n)th
20 criteria and the iterative process continues.

21 **[00702]** Alternatively, the matching systems disclosed in Figures 106A, 106B and
22 106C could be combined. For example, a hybrid approach might use the 106B approach
23 to narrow the field of job listings to job listings meeting at least three secondary criteria
24 and then use the 106C approach to test for secondary criteria in a preferred order. In
25 another embodiment, primary criteria are reused as secondary criteria. For example, it
26 might be advisable to use skills and/or responsibilities required as primary criteria to
27 exclude job listings outside the applicant's interest area. Then after narrowing down job
28 listings based upon a number of secondary criteria it might be beneficial to reuse the
29 skills and/or responsibilities required to find the job listings best suited to the applicant.

30 **[00703]** In addition to screening for matching job listings, in some embodiments
31 the above methods may be similarly utilized in scoring job listings for an applicant. For

1 example, the more criteria a job listing met, the higher the relative score of the job
2 listing would be. As discussed above, the scoring could similarly incorporate different
3 priority levels and/or weights for different criteria in calculating a job listing's score.
4 Also, some level of iterative screening may be implemented as part of the search
5 process.

6 **[00704]** Job listings saved and/or applied for by the applicant using the Job Listing
7 Tracking Extension are recorded by the JTM Core. The JTM Core allows the applicant
8 to access this information. For example, the JTM Core can provide a history screen
9 listing the applicant's saved and applied for jobs, along with relevant information about
10 each saved job. For example, the embodiment of Figure 107A lists job title 10701,
11 company 10702, location 10703, the date the job was applied for 10704, the number of
12 correspondence between the applicant and the employer 10705, the status of the
13 application 10706, and the job's score 10707. The date applied for field shows the date
14 applications were provided to the employer note that listings without dates indicate
15 saved job listings that have not yet been applied for. The correspondence field of the
16 history report can act as a control that when activated displays the history of
17 correspondence between the applicant and the employer. The job application history
18 screen might also provide a mechanism for highlighting applications in which some
19 activity has occurred.

20 **[00705]** In one embodiment, the applicant may rate the stored jobs according to
21 their preferences, as shown in Figure 107C. In one embodiment, the job ratings may be
22 incorporated into the applicant's profile and/or utilized when matching and/or scoring
23 additional jobs. In a further embodiment, the applicant may rate individual aspects of a
24 stored job, such as the job title 10701, company 10702, and/or location 10703. In some
25 embodiments, the ratings of the individual aspects may be associated with or
26 incorporated into the applicant's profile and utilized when matching and/or scoring
27 additional jobs.

28 **[00706]** The JTM Core can also provide a Job View screen that provides detailed
29 information about the applicant's saved jobs. This screen could, for example, be
30 accessed by selecting a job from the history screen. One embodiment of a Job View
31 screen is shown in Figure 108A. The Job View screen provides details about a particular

1 job. The job title 10801 and detailed description of the job 10802 can be provided, as
2 well as, the company 10803, location 10804, job type 10805 (full-time, part-time,
3 freelance, etc.), job category 10806, work experience required 10807, and education
4 level required 10808. The Job View screen might also display meta-data about the job
5 listing 10809, such as, how many times the same job has been posted in the last year or
6 how many days the job listing has been active. The Job View screen may also include a
7 note entry area 10810 for use by the applicant and controls to allow the applicant to
8 access data about the employer and find other similar jobs.

9 **[00707]** As it is advantageous for the applicant to know whether the same job is
10 listed elsewhere, the Job View screen can also include controls identifying other web
11 sites containing the same job listing 10811, which also allow the applicant to view the job
12 listing as it appears on that site. This could be accomplished by integrating a browser
13 window showing the job listing as it appears on the various listing sites or by retrieving
14 and reformatting the information from the various listing sites. The JTM Core can
15 independently search out these duplicate listings or they can be identified through the
16 applicant's search process, such the JTM Core recognizes when to retrieved jobs are in
17 fact identical.

18 **[00708]** Figure 108B shows a particular implementation of a Job View screen. This
19 particular Job View screen is implemented as part of a JTM Core embodied as stand-
20 alone application. As can be seen, other potential features of the toolbar are shown in
21 the Figure.

22 **[00709]** Additionally, as with the job history screen, in some embodiments the Job
23 View Screen may also allow the applicant to rate aspects of the job displayed on the Job
24 View Screen, as shown in Figure 108C. Aspects of the job listing such as the detailed
25 description of the job 10802, the company 10803, location 10804, qualifications
26 (including work experience 10807 and education 10808), starting salary 10812 and
27 benefits 10813. In a further embodiment, certain sub-categories of the provided
28 information, such as certain aspects of the detailed description of the job 10802 may
29 also be rated. For example, if a job description includes a list of technical skills desired,
30 an applicant might give one desired technical skill a high rating (indicating that it is a
31 skill the applicant is interested and/or proficient in), but give another desired technical

1 skill desired a low rating (indicating the applicant is not interested in performing and/or
2 lacks proficiency in that skill). The ratings may be associated with and/or incorporated
3 into the applicant's profile and utilized when matching and/or scoring additional jobs.

4 **[00710]** The JTM Core 10110 can also provide a Company Metrics screen such as is
5 shown in Figure 109A. The Company Metrics screen will allow the applicant to research
6 employers and, in a further embodiment, allow the applicant to rate employers. The
7 Company Metrics screen can include the company name 10901, the number of jobs
8 posted by the employer in the last year 10902, the average amount of time the
9 employer's job listings stay open 10903, other applicant's reviews of the employer and
10 its hiring process 10904, the top jobs posted by the employer 10905, the most common
11 locations of jobs posted by the employer 10906, links to information about the
12 employer's industry 10907, and seasonal hiring patterns for the employer 10908. Like
13 the Job View the Company Metrics screen might also include an area for entering and
14 saving notes made by the applicant 10909. In some embodiments, the Company
15 Metrics screen may also allow the applicant to rate each company as well as particular
16 aspects of the company. The applicant's ratings may be stored and/or incorporated into
17 the applicant's profile, and may further be utilized when matching and/or scoring
18 additional jobs.

19 **[00711]** Figure 109B shows a particular implementation of a company metrics
20 screen. A system user may access the company metrics module of the system in order to
21 conduct research on a company of interest. The module may be populated with
22 additional corporate information supplied by the company, extracted from corporate
23 documents and/or provided by other system users. This particular company metrics
24 screen is implemented as part of a JTM Core embodied as stand-alone application. As
25 can be seen, other potential features of the interface/toolbar are shown in the Figure.

26 **[00712]** According to some embodiments, applicants are provided with matching
27 job listings by accessing an applicant profile (where the applicant profile comprises
28 applicant specific information, such as information obtained from a social network
29 profile and/or a social match platform that includes professional network information),
30 searching job listings with search parameters based on the applicant profile and
31 applicant designated search input (such as prior search history, professional network

1 attributes, social network attributes, etc.), processing search results corresponding to
2 the search parameters and associating the search results with the applicant profile the
3 outputting the search results (e.g., displaying a job listing or ad to a user). In some
4 embodiments, a resume and/or other profile (social or professional network profile,
5 apart from a social match platform profile) may be processed to extract applicant-
6 specific information that is associated with the applicant profile. Prior history, including
7 search history, browsing history, social network information, jobs applied to history,
8 jobs saved history, jobs browsed history, may also be determined and/or associated with
9 a profile.

10 **[00713]** In some embodiments, a social match concept synonym matching engine
11 may be utilized, for example, as a component in a social match platform, to identify and
12 extract specific information from user profile information, for example, information
13 referenced in a selection of text (e.g. a wall post or other profile information), and
14 matches the information to defined concepts. In addition, this identification and
15 matching is performed in the presence of errors and misspellings in the input text or
16 variations in the input text (e.g., use of different words to describe the same entity,
17 variations in writing style and font, language/dialect differences, variations in
18 placement of text on a page, and the like).

19 **[00714]** As used herein, the term “concept” includes any type of information or
20 representation of an idea, topic, category, classification, group, term, unit of meaning
21 and so forth, expressed in any symbolic, graphical, textual, or other forms. For example,
22 concepts typically included in a profile include university names, companies, terms
23 identifying time (e.g., years), experiences, persons, connection, places, locations, names,
24 contact information, hobbies, publications, miscellaneous information, grade point
25 averages, honors, associations, clubs, teams, any type of entity, etc, or a collection of one
26 or more of these. A concept can also be represented by search terms that might be used
27 in a database search, web search, literature search, a search through statutes or case law,
28 a patent search, and the like.

29 **[00715]** While the social match concept synonym matching engine described
30 herein is described in relation to a profile/resume analysis application, and the
31 examples provided are those involving searching for concepts in a resume, the engine

1 can also be used for identifying concepts in other applications. For example, the engine
2 can be used in a word/phrase search or comparison search through documents, articles,
3 stories, publications, books, presentations, etc., in a search for court cases, medical
4 cases, etc., in a search for television programs, radio programs, etc., and many other
5 types of searches.

6 **[00716]** In an exemplary embodiment, the social match concept synonym
7 matching engine (SMCSME) reviews an input string of text from a source (e.g., from a
8 social match profile and/or resume provided by a job applicant, from a website, or any
9 other collection of information). The input string can be composed of letters, numbers,
10 punctuation, words, collections of words or a phrases, sentences, paragraphs, symbols
11 or other characters, any combination of these, etc. The SMCSME also receives the
12 specified concepts to be identified, including concepts which the SMCSME is to identify
13 in the input string. The set of specified concepts can include any information that is
14 being searched for within the input string, such as university attended, degrees attained,
15 skills, companies worked at, titles held, social network posts, interests, information in a
16 publication, on a website, and the like. The SMCSME may identify matches between the
17 specified concepts to be identified and the input string, and translates those matches to
18 one or more matched concepts. The matched concepts are concepts that the
19 SMCSME has identified to be present within the input string (e.g., the SMCSME has
20 found a match between the specified concepts and the text in the input string).

21 **[00717]** As stated above, a problem in concept matching systems is to identify a
22 particular concept cited in a selection of text (e.g., posts on a profile). There are variants
23 of this problem, for example, in one variant, the selection of text is short (e.g., citing at
24 most one entity or concept from a known domain of concepts, such as in a status update
25 or tweet (microblog)), or such as a search input string entered by a user, or a field in a
26 database.. In this variant, the SMCSME can determine if the text identifies a known
27 concept and, if so, the SMCSME can identify that concept. In another variant, the
28 selection of text is long (e.g., identifying more than one concept from one or more
29 known domains of concepts, such as free form history, personal description, browsed
30 communications/websites, etc.). In this variant, the SMCSME can identify the set of
31 concepts contained in the text, and the SMCSME may identify for each concept the sub-

1 string of the text that corresponds to the concept such that no two sub-selections share
2 the same word in the same location of the text. A correspondence between a concept or a
3 pattern and a sub-string or sub-sequence of text or between two tokens is referred to
4 herein as a matching or a match.

5 **[00718]** In one embodiment, usage of the SMCSME may be for a job search and/or
6 job serving application (or more general ad serving application, such as is described
7 with respect to Career Advertising Network). In such implementations, the
8 SMCSME identifies concepts referenced in selections of text in profiles and/or
9 resumes which serve as a source text. The SMCSME reviews an input string of text (e.g.,
10 “I attended Columbia University from 2003 to 2005”) from the profile or
11 resume document for user “Thomas R.” In this example, the SMCSME uses a collection
12 of concepts or a collection or list of terms, either of which serve as the specified
13 concepts to be searched in the user's profile. For example, the collection of concepts may
14 include a collection of universities (e.g., Columbia University, Cornell, Fordham, etc.), a
15 collection of skills (e.g. C++), a collection of companies (e.g., Monster.com), and/or the
16 like. Thus, the SMCSME would search in the profiles for each of these specified
17 concepts and try to identify the concepts named or referenced in the profile (e.g.,
18 matched concepts). Subsequently, these matched concepts can be used to determine if a
19 user, as represented by his/her profile, is a good match for a job position or a targeted
20 ad. As another example, an input string from a job description (e.g., describing desired
21 features in a candidate for a particular position of employment in a company) can be
22 input into the SMCSME along with the profile, and the SMCSME will match concepts
23 found in both input strings. In some embodiments, the collection of concepts is stored
24 in a knowledge base, and, in some embodiments, this knowledge base is included within
25 the SMCSME.

26 **[00719]** In one embodiment, the operation of the social match concept synonym
27 matching engine comprises phases, such by way of non-limiting example, the exemplary
28 phases: (a) Lexical Analysis/Tokenization - breaking up the input string into tokens; (b)
29 Token Matching - determining which tokens in the input string match which tokens in a
30 pattern (and how well they match); (c) Pattern Matching - selecting which patterns
31 match which sub-sequences or sub-strings of the input string; (d) Pattern Scoring -

1 evaluating how closely a sub-string of the input string matches a pattern; and (e) Pattern
2 Selection - selecting which patterns are contained in the input string.

3 **[00720]** The SMCSME may be configured to run on a Social Match Platform server
4 or like system storing and executing the SMCSME. Various modules may be utilized by
5 the SMCSME, including a tokenization module, a representation module, a token
6 matching module, a pattern matching module, a pattern scoring module, and a pattern
7 selection module. Various additional, alternative, and/or different modules may be
8 utilized. Various functionalities may be distributed among the modules in a manner
9 different than as described herein.

10 **[00721]** Lexical Analysis/Tokenization - In one embodiment, a tokenization
11 module divides the input string into one or more input tokens that represent one or
12 more sub-strings of text within the input string. In identifying concepts located in the
13 input string of text, the module decomposes the text into individual tokens, which are
14 components of the input string that together identify a lexical construct. Examples of
15 tokens are words, integers, real numbers, symbols, punctuation marks, numbers,
16 hyperlinks, email addresses, web addresses, etc. The sub-string of the input string of
17 text can be composed of one token or can be composed of more than one token.

18 **[00722]** The process of lexical analysis or tokenization (e.g., dividing text into
19 tokens) may be conducted by a standard lexical analyzer. Lexical analyzers may be hand
20 written, or may be automatically generated from a description of the lexical constructs.
21 For example, the Java programming language runtime environment has direct support
22 for a rudimentary form of lexical analysis (e.g., JLex).

23 **[00723]** In some embodiments, the SMCSME uses lexical constructs, including
24 words, symbols, integers, and real numbers, to represent concepts lexical constructs.
25 These lexical constructs are useful in the job search scenario for identifying of skills,
26 schools, product names, companies, job titles, and the like within profiles for users (e.g.,
27 job candidates). As used herein, the term “word” can include any sequence of alphabetic
28 characters, the term “integer” can include any sequence of numbers, the term “real
29 number” can include any sequence or numbers followed by or embedding a period, and
30 the term “symbol” can include any other character in the character set (e.g., the Unicode

1 character set can be used, which provides encodings for practically all written language).
2 To analyze an input string, the SMCSME can employ any type of lexical analyzer
3 generator (e.g., JLex, JFlex, Flex, Lex, etc.).

4 **[00724]** One example of the tokenization performed by the tokenization module
5 would be an input string such as one that might be found in a social/professional
6 network user profile for a job candidate or in a job description for an employer. For
7 example, the input string might identify the school attended by the job candidate, such
8 as an input string that states that “I attended Columbia University from 2003 to 2005.”
9 The module decomposes this input string into tokens as follows:

10 **[00725]** <I> <attended> <Columbia> <University> <from> <2003> <to >
11 <2005>

12 **[00726]** Thus, the string is broken down into eight separate tokens. A sub-string of
13 this input string could be “Columbia University,” which can be decomposed into tokens
14 as follows:

15 **[00727]** <Columbia> <University>

16 **[00728]** Thus, the sub-string of text within the input string is broken down into two
17 separate tokens that can then be used in identifying specific concepts within the input
18 string.

19 **[00729]** The representation module represents the concept to be identified with a
20 pattern that is divided into one or more pattern tokens. A user can choose certain
21 concepts that the user wishes to have the SMCSME identify in a document. For example,
22 a user can specify a collection of concepts that include schools, and degrees, or only
23 schools. The SMCSME can then be used to identify these concepts within profiles for job
24 candidates, or other text input strings. In some embodiments, the SMCSME can also
25 normalize the input string (possibly before tokenization) by mapping characters in the
26 string from one character set into characters in another character set (e.g. mapping
27 diacritic forms into non-diacritic forms, mapping for variations in case, such as lower
28 case or upper case characters, eliminating certain characters, such as punctuation, etc.).

1 **[00730]** Concepts with one or more patterns can then be decomposed into tokens,
2 according to some embodiments. A number of concepts, such as universities, locations,
3 etc., might be identified by a employer user as desired features in a job candidate. In
4 some embodiments, each of the concepts is represented by a pattern. In some
5 embodiments, the pattern text is the same as the concept text (e.g., the pattern “Cornell
6 University” is the same as the concept “Cornell University). In other embodiments, the
7 pattern differs somewhat from the concept, and can be an abbreviation of the concept or
8 a synonym for the concept. In some embodiments, the pattern is also broken down into
9 one or more tokens. The representation in the form of a pattern and tokens can be used
10 by the SMCSME to identify the associated concepts in the input string of text. The
11 collection of concepts with associated patterns and tokens can be stored in a knowledge
12 base, and the information in the knowledge base can be retrieved when needed to
13 conduct a particular matching.

14 **[00731]** Token Matching - A token matching module identifies a token match
15 between the one or more input tokens and the one or more pattern tokens. Thus, the
16 token matching module can identify possible correspondences between input tokens in
17 the input string of text and pattern tokens in the textual description of concepts.
18 According to the example above, the input string “I attended Columbia University from
19 2003 to 2005” is decomposed into tokens as follows:

20 **[00732]** <I> <attended> <Columbia> <University> <from> <2003> <to >
21 <2005>

22 **[00733]** The token matching module can then identify which patterns of which
23 concepts in a canonical set of concepts have tokens (e.g., pattern tokens) that
24 correspond to tokens in the input string (e.g., input tokens). In the above example, the
25 module might identify a correspondence between the third input token, <Columbia>,
26 and the input token of the pattern “Columbia University.” The token matching
27 module might also identify a correspondence between the fourth token, <University>,
28 and the second token of “Columbia University.” The module may further identify a
29 match between the input token <University> and pattern “Cornell University,” but
30 would not identify a match with the concept “Harvard University” since the textual
31 description or pattern representing this concept does not include the word “University.”

1 **[00734]** In some embodiments, the token matching module is configured to
2 identify an input string that is a synonym of a token by being a subordinate concept to a
3 parent concept represented by the token. For example, the input might include the input
4 string “I attended Radcliff College and Princetn University,” where Radcliff is a college
5 within Harvard University and “Princetn” is likely to be a misspelling of “Princeton.” In
6 these embodiments, the token matching module is configured to identify the input
7 string “Radcliff College” as a synonym for Harvard University, since Radcliff College can
8 be represented as a child concept of the concept Harvard University in the concept
9 hierarchy and it is possibly represented with pattern “Radcliff College.” Thus, when the
10 token <Radcliff> appears in the input string of text, the module can match the text to
11 the input token in the tokenization of “Radcliff College.” In this example, child concepts
12 inherit characteristics or attributes of the parent concept, hence, in this case, are “part-
13 of” the parent concept. As another example, since the concepts represented by the
14 representation module can have one or more patterns or synonyms, “Radcliff College”
15 can be included as a pattern or synonym of concept Harvard University.

16 **[00735]** In addition, in the example above, the token matching module can be
17 configured to identify misspellings, such as “Princetn” as a likely misspelling of
18 “Princeton.” This allows token matches between input tokens and pattern tokens that
19 are not exact. In some embodiments, this is accomplished by measuring the distance
20 between input tokens and pattern tokens using a string similarity metric (e.g., the Jaro-
21 Winkler metric or any other similarity metric). In some embodiments, the similarity
22 metric is scaled to provide a real number in the range 0.0 to 1.0, where 0.0 is perfect
23 mismatch between input and pattern tokens (e.g., no tokens are matched) and 1.0
24 represents a perfect match (e.g., all tokens are matched).

25 **[00736]** The token matching module can compute for each input token a set of
26 matching pattern tokens (e.g., pattern tokens that are either identical to or are similar
27 enough to the input tokens to be considered equivalent to the input tokens). In some
28 embodiments, the module does this by employing a token evaluation function (e.g.,
29 tokenCloseness(InputToken t1, PatternToken t2)) that evaluates the closeness of a input
30 token to a pattern token, producing a value in the range [0.0, 1.0] with 1.0 being a
31 perfect match and 0.0 being a perfect mismatch. The token matching module preferably

1 employs a thresholding function (e.g., tokenMatch (InputToken t1, PatternToken t2))
2 that returns values (e.g., TRUE (the tokens match), FALSE (the tokens do not match),
3 INDETERMINANT (cannot determine whether the tokens match or do not match)).
4 This function can be used to determine whether an input token in the input string
5 should be treated as a match for a pattern token in a pattern.

6 **[00737]** In addition, a number of modifiers can be employed in the SMCSME for
7 usage in matching tokens. One example is the “class modifier” that is used to modify a
8 particular class of lexical constructs. For example, to match the concept “Oracle
9 Database System” or “Oracle 8i,” a pattern such as “Oracle #8 i” could be used, which
10 can be decomposed into tokens to form the following token sequence: <Oracle> <#8>
11 <i>

12 **[00738]** In this example, the class modifier is represented by the pound sign, or
13 “#,” but the modifier can be represented by any symbol, letter, number, character,
14 sequence of characters, etc. The modifier signifies that any input term that is of the
15 same class as the token following the pound sign (e.g., an integer) matches the pattern.
16 In the Oracle example, the pound sign is actually a modifier for the term “8” that follows
17 it. Thus, the token matching module would also identify text “Oracle 7i” as a match for
18 the pattern “Oracle #8 i,” since “7” is an integer and in the same class as “8.” This class
19 modifier can also be used with words, real numbers, symbols, and the like.

20 **[00739]** An “exact modifier” is another example of a modifier that can be used, and
21 is here denoted as a single quote mark or “'”, but can be denoted by another character.
22 This modifier may appear in front of any word or associated with any token. For
23 example, when the exact modifier is used in front of a word, only input words that
24 exactly match the spelling of the token following the modifier are acknowledged as
25 matches. For example, the pattern “Oracle 8 i” would only be matched by text “Oracle 8
26 i” and not by “Oracel 8 i” or any other spelling. The exact modifier can be used to
27 override any default behavior that allows word tokens to match closely spelled words.

28 **[00740]** As still another example, a “stem modifier” can be used, and can be
29 denoted by a tilde symbol or “{tilde over ()},” or any other character. When matching
30 words, it can be beneficial to match all morphological variants of the verb in some cases.

1 In some embodiments, when this modifier appears in front of any pattern or pattern
2 token that is a word, input words or tokens whose stem matches the stem of the word
3 will be considered matches for the token. For example, the phrase “I {tilde over ()}
4 attend Columbia University” would be matched by “I attended Columbia University.”
5 The SMCSME can thus support a number of patterns, including some involving a
6 specific word, integer, symbol or real number, a class of words, integers, symbols, or real
7 numbers, a stemmed word, an exact word, and the like.

8 **[00741]** In some embodiments, the token matching module compares each input
9 token to each and every pattern token associated with a pattern of every concept when
10 performing a token match. In other embodiments, the token matching module creates a
11 hash table for each pattern type used by the SMCSME. A hash module can apply one or
12 more hash functions to each pattern of each concept, placing the resulting hash
13 value/concept combination in the appropriate hash table. To perform token matching,
14 the hash module can then hash each token using the same hash functions. The matching
15 module can then look up the hash values in the corresponding hash tables. When a
16 match or collision is found, it is possible to identify a token match, and thus identify the
17 pattern and the concept that contains that pattern using the hash table. In the case of
18 misspelled words, in some embodiments the token matching module measures the
19 similarity of the input token to the word in the pattern using a similarity metric (e.g., the
20 Jaro-Winkler similarity metric). In some embodiments, the input tokens or sequences of
21 tokens are mapped to other input tokens or sequences (e.g., mapping related terms,
22 such as “Software Developer” to “Software Engineer”). In this manner, it is possible to
23 account for possible variations in the different tokens.

24 **[00742]** When selecting a hash function to use for a pattern type, the hash
25 module can use numerous different types of functions. For example, for a “stemmed
26 word” pattern type, the hash function can be the morphological stem of the word, and a
27 stemming algorithm can be used for these hash functions (e.g., the Porter stemming
28 algorithm). As another example, for the patterns that are constructed from a class
29 modifier, it is possible to maintain a separate hash table for each pattern type. However,
30 in some embodiments, a single hash table is maintained for all class modifier pattern
31 types. For example, if there were only four basic patterns types, the hash function

1 produces one of four values, corresponding to the four basic pattern types. In this way,
2 all input tokens of a particular class can be matched to all patterns of that class.

3 **[00743]** In addition, other hash functions can be used in cases in which the hash
4 function will produce multiple values from a single pattern (e.g., in the case of the
5 potentially misspelled word). In some embodiments, to capture words that sound the
6 same when pronounced, the hash module hashes each word into a metaphone (e.g.,
7 Lawrence Philips' Metaphone Algorithm). Further, to capture input words that have
8 omitted a single character or phoneme, transposed a single character or phoneme, or
9 included an extraneous character or phoneme, the hash module can compute from the
10 metaphone every sub-string that is missing at most one character. This can result in a
11 set of strings that can be used as hash values for matching misspelled words.

12 **[00744]** Pattern Matching - A pattern matching module identifies a pattern match
13 between one of the one or more sub-strings and the pattern based on the token match.
14 In some embodiments, the pattern matching module receives patterns for matching
15 from the representation module or directly from the knowledge base. It is possible to
16 define two classes of evidence with regard to a match. Positive evidence can include
17 actual letters, numbers, words, and symbols in the input that are also contained exactly
18 in the pattern (e.g. letters of an input word that also occur in a particular pattern word
19 or words in an input string that also occur in a pattern). Negative evidence can include
20 actual letters, numbers, words, and symbols in the input that are not contained in the
21 pattern (e.g. extra letters in a misspelled word or words in an input string that match no
22 word in a pattern).

23 **[00745]** In some embodiments, the module identifies which concepts have patterns
24 that are likely to be matched and evaluates how closely various sub-strings of the input
25 string match the pattern. For example, with the input string "I attended Columbia
26 University from 2003 to 2005," there is a correspondence between the third token in the
27 string <Columbia> and the first token on the pattern "Columbia University," and
28 between the fourth token "University" and the second token on the pattern "Columbia
29 University." Therefore, based on these token matches, the module can identify the sub-
30 string "Columbia University" in the input string as a perfect match to the pattern
31 "Columbia University."

1 **[00746]** In some embodiments, the sub-string of the input string is composed of a
2 number of words, symbols, integers, real numbers etc., as shown in the example above.
3 However, in some embodiments, the sub-string being matched is composed of one
4 word, and the input token formed from this one word is matched with a pattern token of
5 a pattern. For example, in “I attended Harvard University,” the token <Harvard>
6 matches with the token on the pattern “Harvard,” so the sub-string “Harvard” matches
7 the pattern “Harvard.”

8 **[00747]** The SMCSME can support a number of basic patterns that can be the
9 fundamental building block for token matching. In some embodiments, a basic pattern
10 is composed of just one word, character, number, symbol, etc. (e.g., “Harvard”) or a
11 simple sequence (e.g., “Columbia University”). In some embodiments, a basic pattern
12 matches only a single token of an input string. Beyond these basic patterns, it is also
13 possible for a pattern to be composed of more than one word, a more complex sequence
14 of words, or a number of sub-patterns to form a compound pattern. A compound
15 pattern can include numerous words, characters, etc. In some embodiments, it is
16 possible for a compound pattern to match a sequence of tokens or a substring of text in
17 an input string.

18 **[00748]** The SMCSME can support a number of compound pattern types. For
19 example, a “set compound pattern” can be a composition of other patterns that is
20 matched if zero or more of its component patterns or sub-patterns (e.g., basic patterns)
21 is matched. An example of a compound pattern is the pattern comprising the basic
22 pattern “Princeton” and the basic pattern “University.” However, in some embodiments,
23 the general definition of the set pattern is recursive. Thus, sub-patterns may be any
24 other pattern, including other set patterns. In some embodiments, the only constraint
25 on matching the set pattern is that no two basic patterns match the same input token.

26 **[00749]** Another example of a compound pattern is the “sequence compound
27 pattern,” which is also a composition of other sub-patterns. In some embodiments, the
28 sequence pattern is identical to the set compound pattern, except that an additional
29 constraint is imposed where the sequence compound pattern is matched only if, for all
30 pairs of matched basic patterns, the order of appearance of the input tokens is the same
31 as the order of the appearance of the basic patterns in the target pattern. This pattern

1 can be used for distinguishing cases where word order is extremely important, such as
 2 the case of “University of Texas” and “Texas University,” two distinct institutions. Still
 3 another example of a compound pattern is the “alternative compound pattern.” In some
 4 embodiments, this pattern is matched if and only if one of more of its sub-patterns is
 5 matched. The alternative compound pattern can be useful for capturing lexical
 6 synonyms, abbreviations, and acronyms, such as “Microsoft Windows” or “Windows” or
 7 “WinXP” or Windows XP.”

8 **[00750]** As yet another example, it is also possible to use a concatenation
 9 constraint, where a sequence of concatenated patterns is applied. Multiple adjacent
 10 tokens can be concatenated in the input string and still be matched. In addition, it can
 11 be required that the matched tokens follow the order of the patterns.

12 **[00751]** The pattern matching module can also determine whether a particular
 13 matching of input tokens to basic patterns satisfies all of the constraints on the pattern.
 14 For example, if every input token is assigned to at most one basic pattern, the
 15 module can evaluate whether the assignment matches the pattern in time linear in the
 16 number of basic patterns. In some embodiments, the module constructs all valid
 17 assignments of input tokens to basic patterns simultaneously, using a recursive
 18 algorithm on the pattern. For example, the module can be used in an attempt to match
 19 the sequence pattern “Texas A&M University.” To determine possible matches of “I went
 20 to the University of Texas in Austin, Tex.,” the tokenization module would tokenize the
 21 input string into:

22 **[00752]** <I> <went> <to > <the> <University> <of > <Texas> <in > <Austin> <,>
 23 <Texas>

24 **[00753]** Both tokens matching “Texas” can be assigned to the basic pattern
 25 “Texas,” and the token <University> can be assigned to the basic pattern “University.”
 26 In some embodiments, the sub-patterns of the input string are recursively evaluated,
 27 and a set of correspondences between input tokens and basic patterns is produced. For
 28 the sequence pattern, the module can compute all possible sub-sets of the
 29 correspondences to the sub-patterns that may appear in the input string sequentially. In
 30 this example, it would be possible to generate four possible matchings: 1) the empty

1 matching, 2) the matching of the seventh input token, <Texas>, to the first basic pattern
2 “Texas,” 3) the matching of the 11th input token, <Texas>, to the first basic pattern
3 “Texas,” and, 4) the matching of the 5th input token, <University>, to the fifth basic
4 pattern, “University.” While each of these matches is valid, intuitively none of the
5 matches is correct, thus indicating the value of scoring the quality of a matching.

6 **[00754]** Pattern Scoring - A pattern scoring module scores the pattern match based
7 on the token match. In some embodiments, a pattern evaluation or scoring function is
8 used in scoring the match. The scoring function can take a matching (e.g., a
9 correspondence between an input token and a basic pattern), and compute a score. In
10 some embodiments, the higher the score received for a match, the more closely the
11 input string matches the pattern. In some embodiments, this input is taken under the
12 constraint that no matching may use the same token in the input or in the pattern (e.g.,
13 there can be no overlapped matchings among input tokens on the input string or among
14 pattern tokens on the pattern).

15 **[00755]** As one example, a scoring function could be used that produces a real
16 number in the range of 0.0 (for perfect mismatch) to 1.0 (for perfect match), analogous
17 to the token matching similarity metric. However, the output of the scoring function can
18 be represented in other manners, as well (e.g., as an integer). In the example described
19 above with the input string “I attended Columbia University from 2003 to 2005” that
20 resulted in identifying the sub-string “Columbia University” in the input string as
21 perfect match to the pattern “Columbia University,” the scoring function might return a
22 value of 1.0.

23 **[00756]** There are a number of different manners in which a score for a pattern
24 matching can be determined. In some embodiments, the pattern scoring module assigns
25 each basic pattern a weight. The weight assigned is drawn from a weight table that is
26 stored independently of the pattern. Weights in the weight table can be set by the user,
27 but a set of weights can also be pre-set by default in the system (e.g., all basic patterns
28 given a weight a 1). In some embodiments, the user can modify the weights as desired
29 (e.g., the employer can modify the weights associated with different desired features for
30 a job applicant to experiment with different weights to find an arrangement that
31 produces the best candidates).

1 **[00757]** As one example, a selection of text, “I went to Princeton University” can be
2 scored against two patterns “Harvard University” and “Princeton University.” If the first
3 pattern is scored on the input string, the module would return a positive score because
4 the input token “University” matches the basic word pattern “University.” In this
5 example, the contribution to the score of the input token “University” on the pattern
6 “Harvard University” is determined by the weight of “University” in the weight table. If
7 all of the basic patterns were given a weight of 1, then each of the three basic patterns,
8 “Harvard,” “University,” and “Princeton” would each get a weight of 1, affecting how
9 each basic pattern would contribute to the overall score for the matching. It is also
10 possible to use weights of patterns to distinguish more important patterns over less
11 important patterns, which is discussed in more detail below.

12 **[00758]** Examples of Scoring Functions - One example of a scoring function that
13 could be used would involve defining the score of a given matching (e.g., a set of
14 correspondences between an input token and a basic pattern) on a particular pattern as
15 the sum of the weights of the matched basic patterns divided by the sum of the weights
16 of all the basic patterns in the pattern. Thus, in the example above regarding Harvard
17 and Princeton, this scoring function would score the matching of the input token
18 “University” to the basic pattern “University” in the pattern “Harvard University” as 0.5,
19 since only one of two basic patterns is matched. For the pattern “Princeton University,”
20 the scoring function would return a score of 1.0, since all basic patterns were matched.

21 **[00759]** While the example of a scoring function described above did properly give
22 a higher score to the input string on the pattern representing Princeton University than
23 to the pattern representing Harvard University, this scoring function may not work as
24 well with some other input strings. For example, with the input string “I lived in
25 Princeton, N.J. while I attended Rutgers University,” the fourth input token,
26 <Princeton>, might be matched to the basic pattern of “Princeton University,” and the
27 eleventh input token <University> to the basic pattern of “Princeton University.”
28 Evaluating this matching using the scoring function described above might return a
29 perfect score of 1.0, but this matching would not be valid.

30 **[00760]** As another example, a different scoring function is used that takes into
31 consideration the input tokens that are not matched, particularly those that lie between

1 the first matched input token and the last matched input token. In the example above
 2 regarding Rutgers University, there are six input tokens that are not matched (e.g., <,>
 3 <NJ> <while> <I> <attended> <Rutgers>). Conceptually, the scoring function can
 4 score the sub-sequence of input tokens between the leftmost matched token in the input
 5 string and the rightmost matched token in the input string versus the pattern. The
 6 scoring function can involve two parts: a factor determined by which tokens in the input
 7 string are matched and not matched, and a factor determined by which basic patterns in
 8 the pattern are matched and not matched. This scoring function can construct a score as
 9 the product of a) the ratio of the weights of the matched input tokens to all tokens in the
 10 input string between and including the leftmost and rightmost matched input tokens,
 11 and b) the ratio of the weights of the matched basic patterns to all the basic patterns.
 12 When applied to the input string “I lived in Princeton, N.J. while I attended Rutgers
 13 University” on the pattern “Princeton University,” the result is the following:
 14 $(2/8)*(2/2)=0.25$.

15 Thus, this scoring function properly returns a much lower score for the match on the
 16 “Princeton University” pattern than the scoring function described in the previous
 17 example. In addition, distance constraints can be used to put a limit on the number of
 18 unmatched tokens between the first matched token and the last matched token.

19 Handling Mis-Ordered Words

20 **[00761]** In many scenarios, token order is also important. The scoring function can
 21 be optionally modified or augmented in a number of ways to account for ordering of
 22 tokens. As one example, the scoring function can be augmented with a penalty function if
 23 tokens are out of order. In any compound pattern used by the SMCSME that allows the
 24 input tokens to be presented in an order distinct from the order of basic patterns only in
 25 a single construct (e.g., the set compound pattern in the examples described above), this
 26 penalty function example can be useful for application to the portion of the score
 27 attributed to basic tokens contained in such a pattern when those basic tokens appear
 28 out of order. For example, the number of inversions present among the sub-patterns of a
 29 set compound pattern can be computed. If no inversions are present (e.g., none of the
 30 sub-patterns are out of order), then the penalty function returns a value of 0.0. If the
 31 sub-patterns in the set compound pattern are matched to input tokens that appear in

1 exactly the opposite order, then the penalty function returns a value of 1.0. Thus, in this
2 example, the score can be reduced by the product of the penalty function and a constant
3 penalty value. In some embodiments, this value ranges from 1% to 50% of the score.

4 **[00762]** Distinguishing Important Sub-Patterns - In some embodiments, it is
5 possible to distinguish that certain sub-patterns forming a compound pattern are more
6 or less important than other sub-patterns. In some embodiments, the scoring function
7 applied by the pattern scoring module assigns a higher score to important patterns that
8 are matched than to similar, but less important patterns that are matched. Conversely, a
9 lower score can be assigned to important patterns that are not matched compared to
10 similar, but less important patterns that are not matched.

11 **[00763]** In some embodiments, sub-pattern importance is conveyed by the pattern
12 scoring module using a weight function. The weight function can be computed by the
13 module using a representative sample of the texts that are likely to be input into the
14 system. From that sample (e.g., possibly matching thousands of concepts), it is possible
15 to construct the weight table based on the inverse of the frequencies of the occurrence of
16 the basic patterns in the sample set. This can be the same weight table as previously
17 described or it can be a separate weight table for storing weights based on the inverse of
18 frequencies of occurrence of patterns. For example, in an input consisting of a set of
19 10,000 names of schools of higher education, the importance of the basic pattern
20 “University” would be very low, since its frequency would be very high, whereas the
21 importance of the basic pattern “Princeton” would be very high, since very few schools
22 are named Princeton.

23 **[00764]** Distinguishing Optional and Required Patterns - In some embodiments,
24 certain basic patterns are entirely optional or absolutely required for the pattern to be
25 considered matched. In these embodiments, optional patterns contribute to the score of
26 a pattern match only when the contribution would be positive. Similarly, required
27 patterns are considered matched only when the score of the required sub-pattern
28 exceeds a minimum score threshold, in these embodiments. A “required” or an
29 “optional” constraint can be used with a pattern to designate that a particular pattern or
30 term is required or optional for a match to be found.

1 **[00765]** Pattern Selection - The pattern selection module determines whether the
2 concept is present in the input string based on the score, wherein which one of the one
3 or more sub-strings of text in the input string naming the concept is identified based on
4 the token match. In some embodiments, the pattern selection module selects from the
5 likely matches a set of non-conflicting matches (e.g., matches that do not overlap). The
6 module can select the pattern match with the total weight that is highest and that does
7 not overlap any other pattern matches for the input string. For example, the input string
8 “I went to Princeton, University of Southern California, and State University of New
9 York,” likely matches may include input substring “Princeton, University” for concept
10 Princeton University and input substring “California, and State University” for concept
11 California State University. However, this type of match would be taking the terms
12 matched out of context. Thus, the scoring function can seek select an optimal set of
13 concepts matches by maximizing the sum of the scores of the matched sub-strings under
14 the constraint that no two sub-strings overlap. In some embodiments, a dynamic
15 programming technique can be employed to perform the selection.

16 **[00766]** In some embodiments, a matching between input tokens and basic
17 patterns within a single pattern can be classified using a threshold function. The score
18 assigned to a matching can encode how well the matching fits the pattern. One method
19 of classifying whether the matching is a reasonable fit for the pattern is to compare the
20 score to a fixed threshold score. Matchings with scores exceeding the threshold value
21 can be considered plausible matches. Those with scores that do not meet the threshold
22 value can be considered implausible matches. For example, if the score is below a given
23 value, the input sub-string spanned by the first term in the matching (e.g., “Princeton”
24 in the above example) and the last term in the matching (e.g., “University”) according to
25 the order of input tokens is deemed to be an implausible match for the pattern (e.g., the
26 pattern “Princeton University”).

27 **[00767]** In one example, let $c[i]$ be the i -th character of the input text. For a
28 plausible match P , let $\text{interval}(P)$ be the smallest closed integer interval that contains the
29 indices of all characters in matched input tokens in P . In this example, two matches P
30 and Q are non-conflicting if and only if $\text{interval}(P)$ does not intersect $\text{interval}(Q)$.
31 Intuitively, P and Q are non-conflicting if and only if P and Q match different sub-

1 sequences of the input text. A weight for a matching P can be defined as $\text{weight}(P)$ or the
2 product of the square of the score of P and the sum of the weights of the basic patterns
3 in P. The pattern selection module can identify (e.g., using dynamic programming) a set
4 of mutually non-conflicting matches whose total weight (e.g., maximum weight) is
5 maximized. When selecting between two conflicting matchings, P and Q, the module can
6 select the one whose weight is larger. When combining non-conflicting matchings, the
7 weight of the combined set of matchings can be set as a) the sum of the weights of the
8 matchings where multiple matchings are sought, or b) the maximum of the weights of
9 the two sets of matchings where only a single matching is sought.

10 **[00768]** Importance Indexing - As discussed above, in some embodiments, for a
11 matching to be considered, its score must exceed a given threshold value. It is also
12 possible to eliminate a large number of patterns without scoring them. For example,
13 when trying to find schools in the text "I went to university at Princeton," conceivably
14 the fourth token of this text <university> matches every instance of the basic pattern
15 "University" in the collection of basic patterns in the knowledge base. Following the
16 algorithm described above, each one of those patterns would be scored, though it may
17 be discovered that the scores were too low to be used for the vast majority of concepts in
18 the knowledge base.

19 **[00769]** In some embodiments, to reduce the scoring effort, the pattern selection
20 module computes an importance value for each basic pattern within a pattern. In these
21 embodiments, if the importance value exceeds the scoring threshold value, then the
22 pattern is scored. The module maintains an importance index that includes, for each
23 basic pattern, the list of patterns that contain the basic pattern and the corresponding
24 importance values. The list can be sorted by importance values. The importance
25 index can be used to find an input token that corresponds to a basic pattern and to
26 identify all patterns whose importance value exceeds the threshold value. These patterns
27 exceeding the threshold can be added to a list of patterns that will be completely scored.
28 Thus, this importance indexing technique can dramatically reduce the number of
29 patterns that need be scored, and, consequently can result in a much faster SMCSME.

30 **[00770]** There are a number of ways to compute the importance values of the basic
31 patterns within a pattern. For example, the module can construct a total order on the set

1 of distinct basic patterns. The module can construct an input text whose score is
2 maximal (e.g., 1.0) and then for each basic pattern, the module can remove all instances
3 of that basic pattern from the input text. The module can also score the resulting input
4 text. The basic pattern that achieves the lowest score can be assigned the highest score
5 achieved when including that basic pattern. Then, that basic pattern can be removed
6 and the second step can be repeated until there are no basic patterns remaining.

7 **[00771]** Measure of Effectiveness - To measure the effectiveness of a method for
8 solving the problem of identifying concepts in a selection of text, it is possible to define a
9 measure of the accuracy of a solution in correctly identifying concepts. When given a
10 fixed set of concepts or a canonical set of concepts to be identified and an input string or
11 selection of text, T, it is possible to enumerate all the matchings, M, of sub-selections or
12 sub-strings of text with concepts in the canonical set. For example, given a set of
13 matchings H(T) that might be made manually by a human reviewing a string of text, it is
14 possible to measure the precision, recall, and accuracy of another method, C(T) (e.g., the
15 method used by the system), for generating matchings on T relative to M as follows:

16 **[00772]** Precision: The number of concepts matched in both H(T) and C(T) divided
17 by the number of concepts matched in C(T).

18 **[00773]** Recall: The number of concepts matched in both H(T) and C(T) divided by
19 the number of concepts matched in H(T).

20 **[00774]** Accuracy: The average of precision and recall.

21 **[00775]** F-Measure: $(2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$

22 **[00776]** The system can be configured to achieve the highest F-Measure or
23 Accuracy possible. To obtain this outcome, it is useful to define a measure of goodness
24 relating a sub-selection of text and a concept (e.g., how good or accurate the identified
25 match made by system is in comparison to the match a human might make through a
26 manual matching). The measure of goodness can correspond well to intuition that a
27 human being would apply when measuring goodness. For example, in a selection of text
28 assumed to name a single city or school, e.g. "Stanford," a resident of Northern
29 California may recognize this as an abbreviation of "Leland Stanford Jr. University,"
30 while a resident of Connecticut may recognize this as a misspelling of the city of

1 Stamford, Conn. Without further contextual information, it may be difficult to make a
2 determination of which concept is a closer or better match. Thus, a reasonable measure
3 of goodness or closeness can allow for abbreviations and allow for spelling errors.

4 **[00777]** In addition to abbreviations and spelling errors, a goodness measure can
5 also recognize synonyms of concepts, including both single word synonyms, e.g.
6 “taught” and “instructed” and multi-word synonyms, e.g. “developed software” and
7 “wrote code.” Other relationships between words can also be used to construct a
8 goodness measure, including the various relationships represented in the electronic
9 lexical database called WordNet. The system provides a class of methods for measuring
10 goodness that can be constructed using one or more hierarchies of concepts defined
11 with words and/or word synonyms, and it is possible to achieve very high levels of
12 accuracy when using methods of goodness constructed from such concept hierarchies.

13 **[00778]** In one embodiment, the SMCSME first receives an input string of text
14 (e.g., from a profile, resume, from a job description, etc.) and a defined set of concepts
15 (e.g., from a knowledgebase of concepts). In some embodiments, the input string is
16 normalized by mapping characters from one character set into characters in another
17 character set. The SMCSME tokenizes the input string, dividing the input string into one
18 or more input tokens, as described above. The SMCSME also represents a concept with
19 one or more patterns that can be divided into tokens. This representation is preferably
20 created in advance and lists a number of concepts to be identified in selections of text or
21 input strings. In some embodiments, the input tokens or token sequences are mapped to
22 other input tokens or sequences. The SMCSME applies hash functions to each pattern of
23 each concept and places the resulting hash value/concept combination in a hash table.
24 The SMCSME can also apply the hash functions to each input token to be matched to get
25 a resultant hash value. After applying the hash functions, the SMCSME can then look up
26 the hash value in the hash table to find a match in the hash table. Thus,
27 SMCSME identifies a token match between one or more input tokens and one or more
28 pattern tokens using the information stored in the hash table.

29 **[00779]** The SMCSME can also identify a pattern match between a sub-string (e.g.,
30 one or more of the input tokens) of the input string and a pattern associated with the
31 concept to be identified. The SMCSME then assigns a weight to the match that can be

1 used in the computation of a score for the match. The SMCSME can assign weights to
2 each basic pattern within a pattern that together equal the total weight for the pattern,
3 and the user can modify the weights as desired. The SMCSME can assign a lower weight
4 to less important of the basic patterns, and a higher weight to more important of the
5 basic patterns. Similarly, the SMCSME can assign a lower weight to more frequently
6 used patterns (e.g., University) and can assign a higher weight to less frequent patterns
7 (e.g., Harvard).

8 **[00780]** In some embodiments, the SMCSME provides pattern scoring and
9 determining whether a concept is present in a selection of text. The
10 SMCSME retrieves an importance value for each basic pattern within a pattern from an
11 importance index (e.g., containing, for each basic pattern, a list of patterns that contain
12 the basic pattern and the corresponding importance values). The importance index can
13 be used to find an input token that corresponds to a basic pattern and to identify all
14 patterns whose importance value exceeds the threshold value for a likely or plausible
15 match (as described above). The SMCSME next determines if the importance value
16 exceeds the threshold value. If the importance value does not exceed the threshold, the
17 SMCSME does not score the pattern match. If the value does exceed the threshold, the
18 SMCSME does score the pattern match. In some embodiments, the SMCSME conducts
19 some scoring of the pattern match before the retrieval of the importance value, but the
20 SMCSME completes the scoring **706** once the importance value is retrieved and is
21 determined to exceed the threshold. The SMCSME scores the pattern match based the
22 weights assigned to basic patterns making up the pattern that together equal the total
23 weight for the pattern.

24 **[00781]** The SMCSME next determines whether the concept is present in the input
25 string based on the score. In some embodiments, the SMCSME selects from the likely
26 matches a set of optimal, non-conflicting matches (e.g., matches that do not overlap).
27 Again, the SMCSME can compare the score to a threshold score to determine how
28 plausible or likely the match is and select the optimal match. The SMCSME selects the
29 pattern match with the total weight that is highest (e.g., the maximal total weight) and
30 where the pattern match does not overlap any other pattern matches for the input
31 string. The SMCSME can apply dynamic programming techniques or other methods to

1 select the matches that are non-conflicting or that include sub-strings that do not
2 overlap. Thus, using these methods, the SMCSME ultimately can automatically identify
3 from a potentially unstructured input string of text one or more specific concepts, where
4 the input string may include errors and/or variations in the text.

5 **[00782]**

6 **[00783]** aaa

7 **[00784]** In order to address various issues and advance the art, the entirety of this
8 application for SOCIAL MATCH PLATFORM APPARATUSES, METHODS AND
9 SYSTEMS (including the Cover Page, Title, Headings, Field, Background, Summary,
10 Brief Description of the Drawings, Detailed Description, Claims, Abstract, Figures,
11 Appendices, and otherwise) shows, by way of illustration, various embodiments in
12 which the claimed innovations may be practiced. The advantages and features of the
13 application are of a representative sample of embodiments only, and are not exhaustive
14 and/or exclusive. They are presented only to assist in understanding and teach the
15 claimed principles. It should be understood that they are not representative of all
16 claimed innovations. As such, certain aspects of the disclosure have not been discussed
17 herein. That alternate embodiments may not have been presented for a specific portion
18 of the innovations or that further undescribed alternate embodiments may be available
19 for a portion is not to be considered a disclaimer of those alternate embodiments. It will
20 be appreciated that many of those undescribed embodiments incorporate the same
21 principles of the innovations and others are equivalent. Thus, it is to be understood that
22 other embodiments may be utilized and functional, logical, operational, organizational,
23 structural and/or topological modifications may be made without departing from the
24 scope and/or spirit of the disclosure. As such, all examples and/or embodiments are
25 deemed to be non-limiting throughout this disclosure. Also, no inference should be
26 drawn regarding those embodiments discussed herein relative to those not discussed
27 herein other than it is as such for purposes of reducing space and repetition. For
28 instance, it is to be understood that the logical and/or topological structure of any
29 combination of any program components (a component collection), other components
30 and/or any present feature sets as described in the figures and/or throughout are not
31 limited to a fixed operating order and/or arrangement, but rather, any disclosed order is

1 exemplary and all equivalents, regardless of order, are contemplated by the disclosure.
2 Furthermore, it is to be understood that such features are not limited to serial execution,
3 but rather, any number of threads, processes, services, servers, and/or the like that may
4 execute asynchronously, concurrently, in parallel, simultaneously, synchronously,
5 and/or the like are contemplated by the disclosure. As such, some of these features may
6 be mutually contradictory, in that they cannot be simultaneously present in a single
7 embodiment. Similarly, some features are applicable to one aspect of the innovations,
8 and inapplicable to others. In addition, the disclosure includes other innovations not
9 presently claimed. Applicant reserves all rights in those presently unclaimed
10 innovations including the right to claim such innovations, file additional applications,
11 continuations, continuations in part, divisions, and/or the like thereof. As such, it
12 should be understood that advantages, embodiments, examples, functional, features,
13 logical, operational, organizational, structural, topological, and/or other aspects of the
14 disclosure are not to be considered limitations on the disclosure as defined by the claims
15 or limitations on equivalents to the claims. It is to be understood that, depending on the
16 particular needs and/or characteristics of a SMP individual and/or enterprise user,
17 database configuration and/or relational model, data type, data transmission and/or
18 network framework, syntax structure, and/or the like, various embodiments of the SMP,
19 may be implemented that enable a great deal of flexibility and customization. For
20 example, aspects of the SMP may be adapted for connecting players in multiplayer
21 online games. While various embodiments and discussions of the SMP have been
22 directed to connecting job candidates and recruiters using a social platform, however, it
23 is to be understood that the embodiments described herein may be readily configured
24 and/or customized for a wide variety of other applications and/or implementations.

25

CLAIMS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

What is claimed is:

1. A processor-implemented cross-social network search method, comprising:
 - obtaining user external social network access credentials;
 - obtaining user internal social network access credentials;
 - providing an external social network data request, using the user external social network access credentials;
 - obtaining, in response to the external social network data request, user external social network data;
 - storing the obtained external social network data in an external social network database table, wherein the external social network database table includes privacy enhanced access controls preventing intermingling of data with internal social network data;
 - storing user internal social network data in an internal social network database table, using the user internal social network access credentials;
 - modifying the user internal social network data using the user external social network data;
 - storing user internal social network data in an internal social network database table, using the user internal social network access credentials, after the modification of the user internal social network data using the user external social network data;
 - obtaining a cross-network search trigger including job search parameters;
 - extracting the job search parameters by parsing the cross-network search trigger, wherein the cross-network search trigger is one of:
 - a user search request;
 - a jobseeker search request;
 - a recruiter search request;
 - an employer search request;
 - a web service request; and

1 an advertisement service request;
2 querying the internal and external social network database tables, using
3 the extracted job search parameters, wherein the querying is network bandwidth
4 reducing via intra-server and not requiring external network queries;
5 obtaining initial query results including a plurality of information types,
6 based on querying the internal and external social network database tables;
7 providing the initial query results to a network reducing graph pathing
8 engine to generate additional cross-network search results and not requiring external
9 network queries;
10 obtaining the additional cross-network search results;
11 generating aggregated search results by aggregating the initial query
12 results and the additional cross-network search results, wherein the aggregated search
13 results are sorted by whether they are initial search results or additional cross-network
14 search results;
15 generating a user interface display data structure including the initial
16 search results and the additional cross-network search results placed within the data
17 structure according to the sorting of the aggregated search results; and
18 providing the aggregated search results in response to the cross-network
19 search trigger.

20
21 2. A processor-implemented cross-social network search method, comprising:
22 obtaining user external social network access credentials;
23 obtaining user internal social network access credentials;
24 providing an external social network data request, using the user external
25 social network access credentials;
26 obtaining, in response to the external social network data request, user
27 external social network data;
28 storing the obtained external social network data in an external social
29 network database table;
30 storing user internal social network data in an internal social network
31 database table, using the user internal social network access credentials;
32 obtaining a cross-network search trigger including job search parameters;

1 extracting the job search parameters by parsing the cross-network search
2 trigger;
3 querying the internal and external social network database tables, using
4 the extracted job search parameters;
5 obtaining initial query results including a plurality of information types,
6 based on querying the internal and external social network database tables;
7 providing the initial query results to a graph pathing engine to generate
8 additional cross-network search results;
9 obtaining the additional cross-network search results;
10 generating aggregated search results by aggregating the initial query
11 results and the additional cross-network search results; and
12 providing the aggregated search results in response to the cross-network
13 search trigger.

14
15 3. The method of claim 2, further comprising:
16 ranking the aggregated search results according to weights assigned to the
17 plurality of information types; and
18 providing the ranked initial query results and the ranked additional cross-
19 network search results to a requestor.

20
21 4. The method of claim 2, wherein the cross-network search trigger is one of: a
22 user search request; a jobseeker search request; a recruiter search request; an employer
23 search request; a web service request; and an advertisement service request.

24
25 5. The method of claim 2, where the job search parameters include one of: a job
26 type; a job location; a business entity; a salary range; an employment term or condition;
27 and an employee identification.

28
29 6. The method of claim 3, wherein the plurality of information types includes a
30 social graph; a person at a business entity; a job type; and a user interest.

31

1 7. The method of claim 6, wherein the weights assigned to the plurality of
2 information types are all distinct from each other.

3

4 8. The method of claim 3, wherein the requestor is one of: a job seeker; a
5 recruiter; a business entity; a web server; and an advertisement server.

6

7 9. The method of claim 2, wherein the aggregated search results are sorted by
8 whether they are initial search results or additional cross-network search results.

9

10 10. The method of claim 9, wherein the additional cross-network search results
11 are prioritized over the initial search results.

12

13 11. The method of claim 9, wherein the initial search results are prioritized over
14 the additional cross-network search results.

15

16 12. The method of claim 9, further comprising:
17 generating a user interface display data structure including the initial
18 search results and the additional cross-network search results placed within the data
19 structure according to the sorting of the aggregated search results.

20

21 13. The method of claim 2, further comprising:
22 generating an updated graph pathing graph based on the additional cross-
23 network search results.

24

25 14. The method of claim 2, wherein the external social network database table is
26 stored in a different database from the internal social network database table.

27

28 15. The method of claim 2, further comprising:
29 modifying the user internal social network data using the user external
30 social network data.

31

32 16. The method of claim 15, further comprising:

1 storing user internal social network data in an internal social network
2 database table, using the user internal social network access credentials, after the
3 modification of the user internal social network data using the user external social
4 network data.

5
6 17. The method of claim 2, further comprising:
7 generating a plurality of advertisements based on the aggregated search results;
8 and
9 distributing the plurality of advertisements to an at least one content provider.

10
11 18. A cross-network social search processor-implemented method, comprising:
12 obtaining user external social network data and user internal social
13 network data;
14 storing the obtained internal and external social network data in separate
15 database tables;
16 obtaining a cross-network job search trigger;
17 searching the internal and external social network data based on the cross-
18 network job search trigger;
19 obtaining initial query results including multiple information types;
20 generating, via a graph pathing engine, additional cross-network search
21 results using the initial query results;
22 aggregating the initial query results and the additional cross-network
23 search results; and
24 providing the aggregated results.

25
26 19. A processor-implemented method for developing a user profile,
27 comprising:
28 receiving an indication from a user to join a social match network;
29 creating a user page for the user;
30 sourcing, from a user profile on a social networking website, user
31 background information;
32 associating the sourced user profile data with the user;

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

populating the user page with the associated profile data; and
displaying the populated user page.

20. The method of claim 19, further comprising:
receiving an indication from the user that the displayed populated user
page is accurate; and
publishing the populated user page to the social match network.

21. The method of claim 19, further comprising:
receiving an indication from the user to update the displayed populated
user page;
displaying an editable version of the populated user page;
receiving an indication from the user that the user has completed updating
the editable version of the populated user page with updated information;
populating the user page with the updated information; and
publishing the populated user page with the updated information.

22. The method of claim 19, wherein the user background information is
sourced from at least two social networking websites.

23. A cross-social network search system, comprising:
means to obtain user external social network access credentials;
means to obtain user internal social network access credentials;
means to provide an external social network data request, using the user
external social network access credentials;
means to obtain, in response to the external social network data request,
user external social network data;
means to store the obtained external social network data in an external
social network database table, wherein the external social network database table
includes privacy enhanced access controls preventing intermingling of data with
internal social network data;

1 means to store user internal social network data in an internal social
2 network database table, using the user internal social network access credentials;

3 means to modify the user internal social network data using the user
4 external social network data;

5 means to store user internal social network data in an internal social
6 network database table, using the user internal social network access credentials, after
7 the modification of the user internal social network data using the user external social
8 network data;

9 means to obtain a cross-network search trigger including job search
10 parameters;

11 means to extract the job search parameters by parsing the cross-network
12 search trigger, wherein the cross-network search trigger is one of:

- 13 a user search request;
- 14 a jobseeker search request;
- 15 a recruiter search request;
- 16 an employer search request;
- 17 a web service request; and
- 18 an advertisement service request;

19 means to query the internal and external social network database tables,
20 using the extracted job search parameters, wherein the querying is network bandwidth
21 reducing via intra-server and not requiring external network queries;

22 means to obtain initial query results including a plurality of information
23 types, based on querying the internal and external social network database tables;

24 means to provide the initial query results to a network reducing graph
25 pathing engine to generate additional cross-network search results and not requiring
26 external network queries;

27 means to obtain the additional cross-network search results;

28 means to generate aggregated search results by aggregating the initial
29 query results and the additional cross-network search results, wherein the aggregated
30 search results are sorted by whether they are initial search results or additional cross-
31 network search results;

1 means to generate a user interface display data structure including the
2 initial search results and the additional cross-network search results placed within the
3 data structure according to the sorting of the aggregated search results; and

4 means to provide the aggregated search results in response to the cross-
5 network search trigger.

6

7 24. A cross-social network search system, comprising:

8 means to obtain user external social network access credentials;

9 means to obtain user internal social network access credentials;

10 means to provide an external social network data request, using the user
11 external social network access credentials;

12 means to obtain, in response to the external social network data request,
13 user external social network data;

14 means to store the obtained external social network data in an external
15 social network database table;

16 means to store user internal social network data in an internal social
17 network database table, using the user internal social network access credentials;

18 means to obtain a cross-network search trigger including job search
19 parameters;

20 means to extract the job search parameters by parsing the cross-network
21 search trigger;

22 means to query the internal and external social network database tables,
23 using the extracted job search parameters;

24 means to obtain initial query results including a plurality of information
25 types, based on querying the internal and external social network database tables;

26 means to provide the initial query results to a graph pathing engine to
27 generate additional cross-network search results;

28 means to obtain the additional cross-network search results;

29 means to generate aggregated search results by aggregating the initial
30 query results and the additional cross-network search results; and

31 means to provide the aggregated search results in response to the cross-
32 network search trigger.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

25. The system of claim 24, further comprising:
means to rank the aggregated search results according to weights assigned to the plurality of information types; and
means to provide the ranked initial query results and the ranked additional cross-network search results to a requestor.

26. The system of claim 24, wherein the cross-network search trigger is one of: a user search request; a jobseeker search request; a recruiter search request; an employer search request; a web service request; and an advertisement service request.

27. The system of claim 24, where the job search parameters include one of: a job type; a job location; a business entity; a salary range; an employment term or condition; and an employee identification.

28. The system of claim 25, wherein the plurality of information types includes a social graph; a person at a business entity; a job type; and a user interest.

29. The system of claim 28, wherein the weights assigned to the plurality of information types are all distinct from each other.

30. The system of claim 25, wherein the requestor is one of: a job seeker; a recruiter; a business entity; a web server; and an advertisement server.

31. The system of claim 24, wherein the aggregated search results are sorted by whether they are initial search results or additional cross-network search results.

32. The system of claim 31, wherein the additional cross-network search results are prioritized over the initial search results.

33. The system of claim 31, wherein the initial search results are prioritized over the additional cross-network search results.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

34. The system of claim 31, further comprising:
means to generate a user interface display data structure including the initial search results and the additional cross-network search results placed within the data structure according to the sorting of the aggregated search results.

35. The system of claim 24, further comprising:
means to generate an updated graph pathing graph based on the additional cross-network search results.

36. The system of claim 24, wherein the external social network database table is stored in a different database from the internal social network database table.

37. The system of claim 24, further comprising:
means to modify the user internal social network data using the user external social network data.

38. The system of claim 37, further comprising:
means to store user internal social network data in an internal social network database table, using the user internal social network access credentials, after the modification of the user internal social network data using the user external social network data.

39. The system of claim 24, further comprising:
means to generate a plurality of advertisements based on the aggregated search results; and
means to distribute the plurality of advertisements to an at least one content provider.

40. A cross-network social search system, comprising:
means to obtain user external social network data and user internal social network data;

1 means to store the obtained internal and external social network data in
2 separate database tables;

3 means to obtain a cross-network job search trigger;

4 means to search the internal and external social network data based on the
5 cross-network job search trigger;

6 means to obtain initial query results including multiple information types;

7 means to generate, via a graph pathing engine, additional cross-network
8 search results using the initial query results;

9 means to aggregate the initial query results and the additional cross-
10 network search results; and

11 means to provide the aggregated results.

12

13 41. A system for developing a user profile, comprising:

14 means to receive an indication from a user to join a social match network;

15 means to create a user page for the user;

16 means to source, from a user profile on a social networking website, user
17 background information;

18 means to associate the sourced user profile data with the user;

19 means to populate the user page with the associated profile data; and

20 means to display the populated user page.

21

22 42. The system of claim 41, further comprising:

23 means to receive an indication from the user that the displayed populated
24 user page is accurate; and

25 means to publish the populated user page to the social match network.

26

27 43. The system of claim 41, further comprising:

28 means to receive an indication from the user to update the displayed
29 populated user page;

30 means to display an editable version of the populated user page;

31 means to receive an indication from the user that the user has completed
32 updating the editable version of the populated user page with updated information;

1 means to populate the user page with the updated information; and
2 means to publish the populated user page with the updated information.

3
4 44. The system of claim 41, wherein the user background information is
5 sourced from at least two social networking websites.

6
7 45. A processor-readable medium storing processor-issuable instructions to:
8 obtain user external social network access credentials;
9 obtain user internal social network access credentials;
10 provide an external social network data request, using the user external
11 social network access credentials;
12 obtain, in response to the external social network data request, user
13 external social network data;
14 store the obtained external social network data in an external social
15 network database table, wherein the external social network database table includes
16 privacy enhanced access controls preventing intermingling of data with internal social
17 network data;
18 store user internal social network data in an internal social network
19 database table, using the user internal social network access credentials;
20 modify the user internal social network data using the user external social
21 network data;
22 store user internal social network data in an internal social network
23 database table, using the user internal social network access credentials, after the
24 modification of the user internal social network data using the user external social
25 network data;
26 obtain a cross-network search trigger including job search parameters;
27 extract the job search parameters by parsing the cross-network search
28 trigger, wherein the cross-network search trigger is one of:
29 a user search request;
30 a jobseeker search request;
31 a recruiter search request;
32 an employer search request;

1 a web service request; and
2 an advertisement service request;
3 query the internal and external social network database tables, using the
4 extracted job search parameters, wherein the querying is network bandwidth reducing
5 via intra-server and not requiring external network queries;
6 obtain initial query results including a plurality of information types,
7 based on querying the internal and external social network database tables;
8 provide the initial query results to a network reducing graph pathing
9 engine to generate additional cross-network search results and not requiring external
10 network queries;
11 obtain the additional cross-network search results;
12 generate aggregated search results by aggregating the initial query results
13 and the additional cross-network search results, wherein the aggregated search results
14 are sorted by whether they are initial search results or additional cross-network search
15 results;
16 generate a user interface display data structure including the initial search
17 results and the additional cross-network search results placed within the data structure
18 according to the sorting of the aggregated search results; and
19 provide the aggregated search results in response to the cross-network
20 search trigger.

21
22 46. A processor-readable medium storing processor-issuable instructions to:
23 obtain user external social network access credentials;
24 obtain user internal social network access credentials;
25 provide an external social network data request, using the user external
26 social network access credentials;
27 obtain, in response to the external social network data request, user
28 external social network data;
29 store the obtained external social network data in an external social
30 network database table;
31 store user internal social network data in an internal social network
32 database table, using the user internal social network access credentials;

1 obtain a cross-network search trigger including job search parameters;
2 extract the job search parameters by parsing the cross-network search
3 trigger;
4 query the internal and external social network database tables, using the
5 extracted job search parameters;
6 obtain initial query results including a plurality of information types,
7 based on querying the internal and external social network database tables;
8 provide the initial query results to a graph pathing engine to generate
9 additional cross-network search results;
10 obtain the additional cross-network search results;
11 generate aggregated search results by aggregating the initial query results
12 and the additional cross-network search results; and
13 provide the aggregated search results in response to the cross-network
14 search trigger.

15
16 47. The medium of claim 46, further comprising instructions to:
17 rank the aggregated search results according to weights assigned to the
18 plurality of information types; and
19 provide the ranked initial query results and the ranked additional cross-
20 network search results to a requestor.

21
22 48. The medium of claim 46, wherein the cross-network search trigger is one of: a
23 user search request; a jobseeker search request; a recruiter search request; an employer
24 search request; a web service request; and an advertisement service request.

25
26 49. The medium of claim 46, where the job search parameters include one of: a
27 job type; a job location; a business entity; a salary range; an employment term or
28 condition; and an employee identification.

29
30 50. The medium of claim 47, wherein the plurality of information types includes a
31 social graph; a person at a business entity; a job type; and a user interest.

32

1 51. The medium of claim 50, wherein the weights assigned to the plurality of
2 information types are all distinct from each other.

3

4 52. The medium of claim 47, wherein the requestor is one of: a job seeker; a
5 recruiter; a business entity; a web server; and an advertisement server.

6

7 53. The medium of claim 46, wherein the aggregated search results are sorted by
8 whether they are initial search results or additional cross-network search results.

9

10 54. The medium of claim 53, wherein the additional cross-network search results
11 are prioritized over the initial search results.

12

13 55. The medium of claim 53, wherein the initial search results are prioritized over
14 the additional cross-network search results.

15

16 56. The medium of claim 53, further comprising instructions to:
17 generate a user interface display data structure including the initial search
18 results and the additional cross-network search results placed within the data structure
19 according to the sorting of the aggregated search results.

20

21 57. The medium of claim 46, further comprising instructions to:
22 generate an updated graph pathing graph based on the additional cross-
23 network search results.

24

25 58. The medium of claim 46, wherein the external social network database table
26 is stored in a different database from the internal social network database table.

27

28 59. The medium of claim 46, further comprising instructions to:
29 modify the user internal social network data using the user external social
30 network data.

31

32 60. The medium of claim 59, further comprising instructions to:

1 store user internal social network data in an internal social network
2 database table, using the user internal social network access credentials, after the
3 modification of the user internal social network data using the user external social
4 network data.

5
6 61. The medium of claim 46, further comprising instructions to:
7 generate a plurality of advertisements based on the aggregated search results;
8 and
9 distribute the plurality of advertisements to an at least one content provider.

10
11 62. A processor-readable medium storing processor-issuable instructions to:
12 obtain user external social network data and user internal social network
13 data;
14 store the obtained internal and external social network data in separate
15 database tables;
16 obtain a cross-network job search trigger;
17 search the internal and external social network data based on the cross-
18 network job search trigger;
19 obtain initial query results including multiple information types;
20 generate, via a graph pathing engine, additional cross-network search
21 results using the initial query results;
22 aggregate the initial query results and the additional cross-network search
23 results; and
24 provide the aggregated results.

25
26 63. A processor-readable medium for developing a user profile storing
27 processor-issuable instructions to:
28 receive an indication from a user to join a social match network;
29 create a user page for the user;
30 source, from a user profile on a social networking website, user
31 background information;
32 associate the sourced user profile data with the user;

1 populate the user page with the associated profile data; and
2 display the populated user page.

3
4 64. The medium of claim 63, further comprising instructions to:
5 receive an indication from the user that the displayed populated user page
6 is accurate; and
7 publish the populated user page to the social match network.

8
9 65. The medium of claim 63, further comprising instructions to:
10 receive an indication from the user to update the displayed populated user
11 page;
12 display an editable version of the populated user page;
13 receive an indication from the user that the user has completed updating
14 the editable version of the populated user page with updated information;
15 populate the user page with the updated information; and
16 publish the populated user page with the updated information.

17
18 66. The medium of claim 63, wherein the user background information is
19 sourced from at least two social networking websites.

20
21 67. A cross-social network search apparatus, comprising:
22 a memory;
23 a processor disposed in communication with said memory, and configured
24 to issue a plurality of processing instructions stored in the memory, wherein the
25 processor issues instructions to:
26 obtain user external social network access credentials;
27 obtain user internal social network access credentials;
28 provide an external social network data request, using the user
29 external social network access credentials;
30 obtain, in response to the external social network data request, user
31 external social network data;

1 store the obtained external social network data in an external social
2 network database table, wherein the external social network database table includes
3 privacy enhanced access controls preventing intermingling of data with internal social
4 network data;

5 store user internal social network data in an internal social network
6 database table, using the user internal social network access credentials;

7 modify the user internal social network data using the user external
8 social network data;

9 store user internal social network data in an internal social network
10 database table, using the user internal social network access credentials, after the
11 modification of the user internal social network data using the user external social
12 network data;

13 obtain a cross-network search trigger including job search
14 parameters;

15 extract the job search parameters by parsing the cross-network
16 search trigger, wherein the cross-network search trigger is one of:

- 17 a user search request;
- 18 a jobseeker search request;
- 19 a recruiter search request;
- 20 an employer search request;
- 21 a web service request; and
- 22 an advertisement service request;

23 query the internal and external social network database tables,
24 using the extracted job search parameters, wherein the querying is network bandwidth
25 reducing via intra-server and not requiring external network queries;

26 obtain initial query results including a plurality of information
27 types, based on querying the internal and external social network database tables;

28 provide the initial query results to a network reducing graph
29 pathing engine to generate additional cross-network search results and not requiring
30 external network queries;

31 obtain the additional cross-network search results;

1 generate aggregated search results by aggregating the initial query
2 results and the additional cross-network search results, wherein the aggregated search
3 results are sorted by whether they are initial search results or additional cross-network
4 search results;

5 generate a user interface display data structure including the initial
6 search results and the additional cross-network search results placed within the data
7 structure according to the sorting of the aggregated search results; and

8 provide the aggregated search results in response to the cross-
9 network search trigger.

10

11 68. A cross-social network search apparatus, comprising:

12 a memory;

13 a processor disposed in communication with said memory, and configured
14 to issue a plurality of processing instructions stored in the memory, wherein the
15 processor issues instructions to:

16 obtain user external social network access credentials;

17 obtain user internal social network access credentials;

18 provide an external social network data request, using the user
19 external social network access credentials;

20 obtain, in response to the external social network data request, user
21 external social network data;

22 store the obtained external social network data in an external social
23 network database table;

24 store user internal social network data in an internal social network
25 database table, using the user internal social network access credentials;

26 obtain a cross-network search trigger including job search
27 parameters;

28 extract the job search parameters by parsing the cross-network
29 search trigger;

30 query the internal and external social network database tables,
31 using the extracted job search parameters;

1 obtain initial query results including a plurality of information
2 types, based on querying the internal and external social network database tables;
3 provide the initial query results to a graph pathing engine to
4 generate additional cross-network search results;
5 obtain the additional cross-network search results;
6 generate aggregated search results by aggregating the initial query
7 results and the additional cross-network search results; and
8 provide the aggregated search results in response to the cross-
9 network search trigger.

10

11 69. The apparatus of claim 68, wherein the processor further issues instructions
12 to:

13 rank the aggregated search results according to weights assigned to the
14 plurality of information types; and

15 provide the ranked initial query results and the ranked additional cross-
16 network search results to a requestor.

17

18 70. The apparatus of claim 68, wherein the cross-network search trigger is one of:
19 a user search request; a jobseeker search request; a recruiter search request; an
20 employer search request; a web service request; and an advertisement service request.

21

22 71. The apparatus of claim 68, where the job search parameters include one of: a
23 job type; a job location; a business entity; a salary range; an employment term or
24 condition; and an employee identification.

25

26 72. The apparatus of claim 69, wherein the plurality of information types includes
27 a social graph; a person at a business entity; a job type; and a user interest.

28

29 73. The apparatus of claim 72, wherein the weights assigned to the plurality of
30 information types are all distinct from each other.

31

1 74. The apparatus of claim 69, wherein the requestor is one of: a job seeker; a
2 recruiter; a business entity; a web server; and an advertisement server.

3

4 75. The apparatus of claim 68, wherein the aggregated search results are sorted
5 by whether they are initial search results or additional cross-network search results.

6

7 76. The apparatus of claim 75, wherein the additional cross-network search
8 results are prioritized over the initial search results.

9

10 77. The apparatus of claim 75, wherein the initial search results are prioritized
11 over the additional cross-network search results.

12

13 78. The apparatus of claim 75, wherein the processor further issues instructions
14 to:

15 generate a user interface display data structure including the initial search
16 results and the additional cross-network search results placed within the data structure
17 according to the sorting of the aggregated search results.

18

19 79. The apparatus of claim 68, wherein the processor further issues instructions
20 to:

21 generate an updated graph pathing graph based on the additional cross-
22 network search results.

23

24 80. The apparatus of claim 68, wherein the external social network database
25 table is stored in a different database from the internal social network database table.

26

27 81. The apparatus of claim 68, wherein the processor further issues instructions
28 to:

29 modify the user internal social network data using the user external social
30 network data.

31

1 82. The apparatus of claim 81, wherein the processor further issues instructions
2 to:
3 store user internal social network data in an internal social network
4 database table, using the user internal social network access credentials, after the
5 modification of the user internal social network data using the user external social
6 network data.

7
8 83. The apparatus of claim 68, wherein the processor further issues instructions
9 to:
10 generate a plurality of advertisements based on the aggregated search results;
11 and
12 distribute the plurality of advertisements to an at least one content provider.

13
14 84. A cross-network social search apparatus, comprising:
15 a memory;
16 a processor disposed in communication with said memory, and configured
17 to issue a plurality of processing instructions stored in the memory, wherein the
18 processor issues instructions to:
19 obtain user external social network data and user internal social
20 network data;
21 store the obtained internal and external social network data in
22 separate database tables;
23 obtain a cross-network job search trigger;
24 search the internal and external social network data based on the
25 cross-network job search trigger;
26 obtain initial query results including multiple information types;
27 generate, via a graph pathing engine, additional cross-network
28 search results using the initial query results;
29 aggregate the initial query results and the additional cross-network
30 search results; and
31 provide the aggregated results.

32

1 85. An apparatus for developing a user profile, comprising:
2 a memory;
3 a processor disposed in communication with said memory, and configured
4 to issue a plurality of processing instructions stored in the memory, wherein the
5 processor issues instructions to:
6 receive an indication from a user to join a social match network;
7 create a user page for the user;
8 source, from a user profile on a social networking website, user
9 background information;
10 associate the sourced user profile data with the user;
11 populate the user page with the associated profile data; and
12 display the populated user page.

13
14 86. The apparatus of claim 85, wherein the processor further issues
15 instructions to:
16 receive an indication from the user that the displayed populated user page
17 is accurate; and
18 publish the populated user page to the social match network.

19
20 87. The apparatus of claim 85, wherein the processor further issues
21 instructions to:
22 receive an indication from the user to update the displayed populated user
23 page;
24 display an editable version of the populated user page;
25 receive an indication from the user that the user has completed updating
26 the editable version of the populated user page with updated information;
27 populate the user page with the updated information; and
28 publish the populated user page with the updated information.

29
30 88. The apparatus of claim 85, wherein the user background information is
31 sourced from at least two social networking websites.

32

FIGURE 1

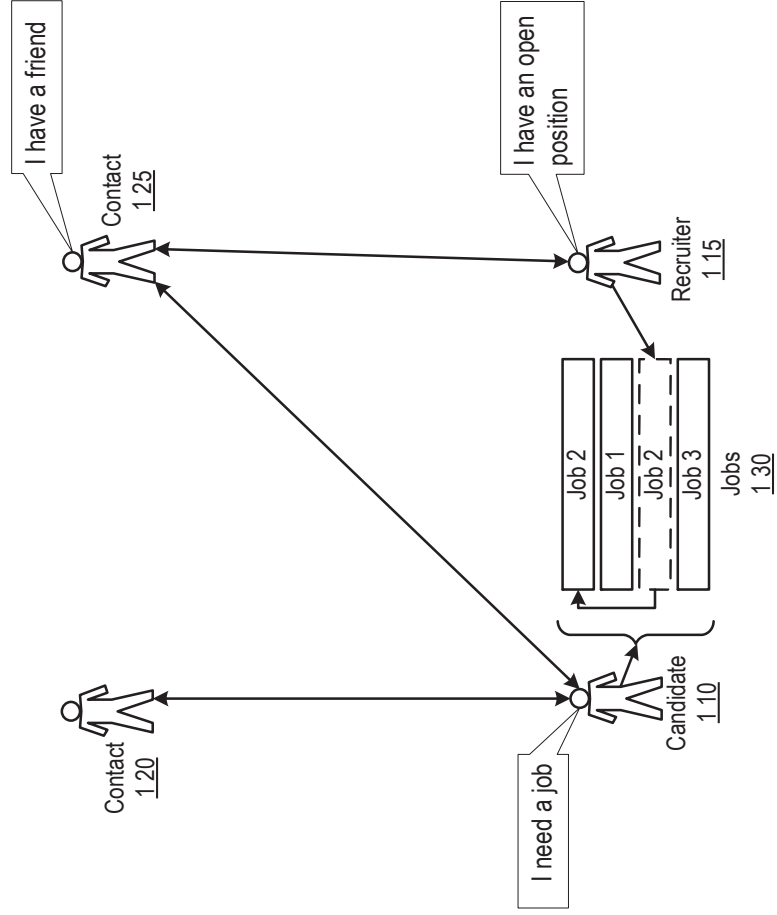
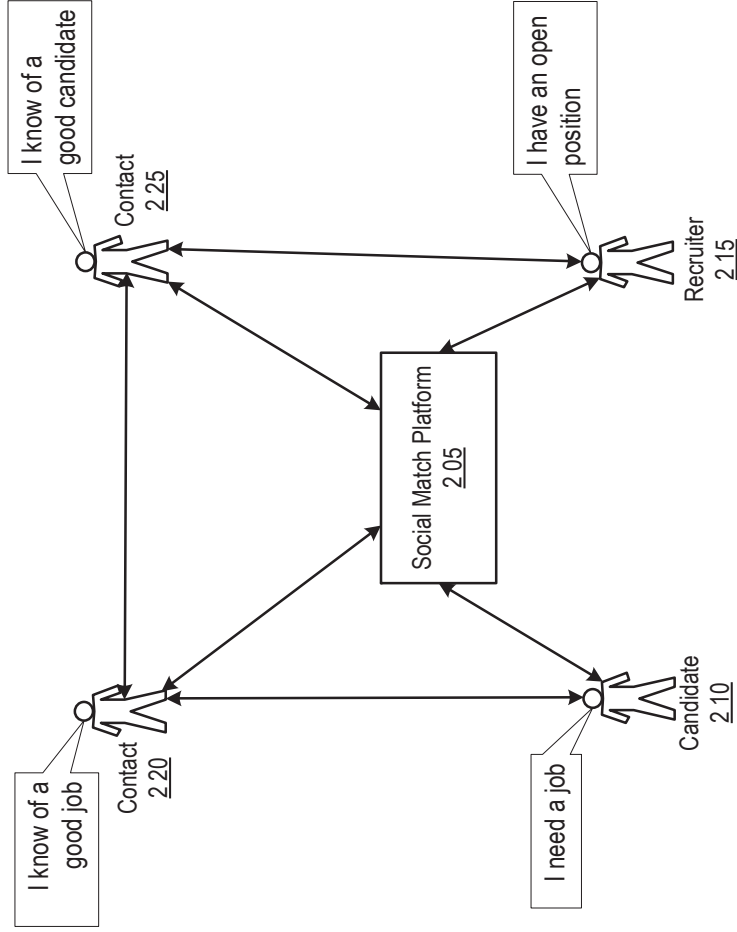


FIGURE 2A



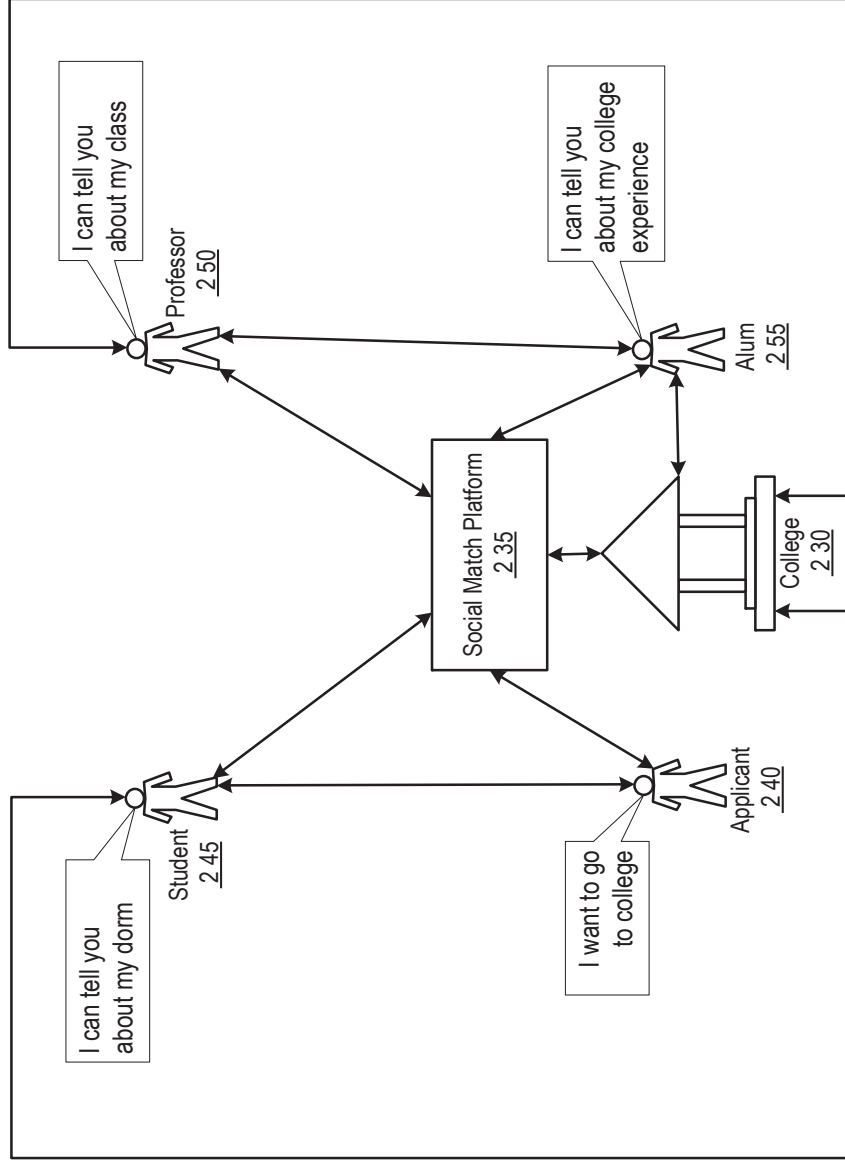


FIGURE 2B

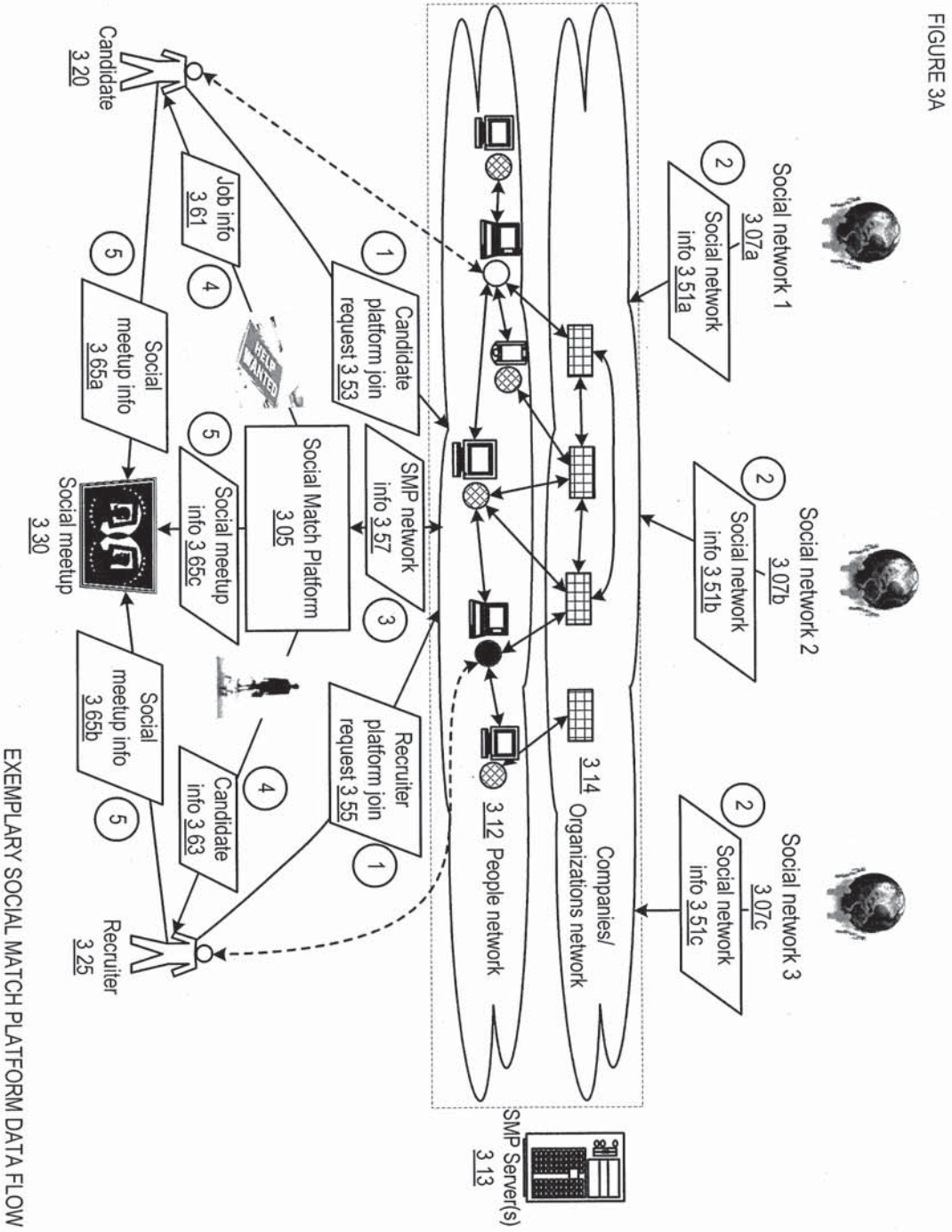


FIGURE 3A

EXEMPLARY SOCIAL MATCH PLATFORM DATA FLOW

FIGURE 3B

facebook Messages Profile Network Jobs Post a Job Search

5/167

Home Profile Find Friends Account

BeKnown.

David Kelly
Master Electrician at Morrissey Electric
Cambridge, MA

Your import from Monster was successful

PDF Add Profile Section Edit

Summary Edit

David is a highly accomplished, dedicated and disciplined individual recognized as a team player seeking to contribute to and grow with a dynamic, progressive, and innovative organization. Direct and decisive leader with a "hands-on" management style. Strong qualifications in personnel development.

Experience Edit

Master Electrician
Morrissey Electric
March 2007 - Present - Cambridge, MA

Journeyman Electrician
Norwood Electric
March 2005 - March 2007 (2 years) - Knoxville, TN

At CSC, we focus on one thing: servicing our customers. Through our people, passion and expertise, we'll gain a thorough understanding of your organization's needs and deliver best-in-class products and services. It's this guiding principle that has turned us into the fastest growing data communications and security products distributor in North America

Electrician Apprentice
Lamp Works
December 2000 - December 2005 (5 years) - Etchic, AK

- Assigned works to other staff as per requirement.
- Installed, repaired, maintained the electrical system, and electrical, electronic equipment.
- Repaired motors, other electrical and electronics equipment.
- Trained other electricians, laborers which are newly hired.
- Repaired or replaced fire alarm system, security alarm system.
- Established work priorities by considering other things.

Education Edit

University of Tennessee at Knoxville '03
Bachelor's Electrical Technology

Skills Edit

All Badges (7)

David Kelly
Master Electrician
Morrissey Electric
david.kelly@morrisseyelec.com

Professional Network Stats

- 522 Connections
- 43,456 2nd Degree Connections
- 86 Followed Companies
- 98,267 Companies

Profile Completeness 55%

What to do next?

- Complete Tasks (+45%)

Endorse Your Connections

Rick Steele Endorse

Games

Profile Network Jobs Post a Job Search

FIGURE 3C

6/167

Home Profile Find Friends Account ▾
facebook

Messages Profile Network Jobs Post a Job
BeKnown.

Rick Steele
General Contractor
Z Group Construction
rsteale@zgroupco.com

Professional Network Stats

- 1 Connections
- 15 2nd Degree Connections
- 0 Followed Companies
- 67 Companies











Profile Completeness 45%

What to do next?

- Add Interests (+5%)
- Complete Tasks (+50%)


Invite Friends


Connect Professionally With These Friends


 Christopher Leonard Invite	 Cody McManus Invite
 Erin Myers Invite	 Summer Sanders Invite
 Michael Norman Invite	 Brandi Henderson Invite
 Kyle Chafin Invite	 Brenden Anderson Invite
 Robert S. Smith Invite	 Bunthol Kim Invite

Find Friends:



Invite Connections From Other Accounts


 Twitter


 Gmail


 Yahoo!

Invite Your Facebook Friends

 Cissy Seeto Invite	 Jessie Dean Invite
--	--

Games

Interruption
 Moments Kaitiah to Move History
 Time is coming and its upon! The year has what it takes to be Mary's love (Contributor?)

Interruption Working
 Get paid with a job-winning resource crafted by me!

Real Time Controversy
 Comment with other people in your field and able to state an opinion and see developments.

Where a Job Offer is Paid on FBES
 Learn why offers get ahead and what you can do to get your offer back on track.

Create an Ad


Interruption Working

Real Time Controversy

Where a Job Offer is Paid on FBES

7/167

FIGURE 3D



facebook


Home Profile Find Friends Account

BeKnown.

Messages Profile Network Jobs Post a Job



Search

Company Search



David Kelly
Edit My Profile

All Badges (7)

David Kelly
Master Electrician
Morrissey Electric
david.kelly@morrisseyllc.com

Professional Network Stats

522 Connections
43,456 2nd Degree Connections
86 Followed Companies
98,267 Companies

Profile Completeness

55%

What to do next?

• Complete Tasks (+45%)


Endorse Your Connections

Rick Steele ✕
Endorse

Helena McDonel ✕
Endorse


Torsten Gottwald ✕

Badges




Established
You've worked at one place for 5+ years.
We gotta give you mad props, dude.
Unlocked May 18, 2011 at 10:13pm


All BeKnown Badges




Established
April 23, 2011




Loyal
May 18, 2011




25
First-Class Connector




500+
Epic Connector




250
Super Connector




100
Graduate Connector



Master's




Alumnus



Dedicated


Sponsored

Plano Butuh To Move Home? Time is running out to apply! Do you have what it takes to be Plano's new Media Production Coordinator?




Sponsored

Monster Resume Writing Get hired with a job-winning resume crafted by our experts.



Sponsored

Find Your Community Connect with other people in your field and stay up-to-date on acute and new developments.



Sponsored

When a Job Offer is Put on Hold Learn why offers get shelved and what you can do to get your offer back on track.




FIGURE 3E

8/167

Home Profile Find Friends Account
facebook

Messages 3 Profile Network Jobs Post a Job
BeKnown

Denise Silber
Marketing Manager
Z Group Construction
dsilber@zgroupco.com
555-555-5555

All Badges (3)





Professional Network Stats

- 426 Connections
- 34,124 2nd Degree Connections
- 25 Followed Companies
- 56,108 Companies

Profile Completeness

65%

What to do next?

- Complete Tasks (+35%)

Endorse Your Connections

David Kelly Endorse



Job Feed

All Jobs • Jobs for Friends • My Posts

Any Job Title in Any Location

Powered by **monster** Search

Electrician
Z Group Construction • Washington, DC 01223

Z Group Construction is an environmental friendly construction company. We are looking for a highly experienced electrician who has a positive can-do attitude, can work closely with...

Suggested Connections for this Job:



May 12 at 3:15pm • Share

Award: \$3,000

ELECTRICIAN- Commercial electrician Min. 3
Company Confidential • Riverside, CA 92503

ELECTRICIAN- Commercial electrician Min. 3 yr exp. Email: baxterelectric@verizon.com.

Looking for an experienced electrician who knows how to re-wire and entire house and...

Suggested Connections for this Job:



May 10 at 8:42pm • Share

Award: \$1,000

Electrician
DeWitt Electric • DeWitt, IA, 52742

Electrician JOURNEYMAN ELECTRICIAN Electrical contractor has an immediate opening for a Journeyman Electrician. This person must have a minimum of 4 years experience l...

Suggested Connections for this Job:



May 10 at 4:31pm • Share

Award: \$2,500

Games

Sponsored
Mariusz Szustak's Who's Missing



Time is running out! In search of the missing, you have 10 minutes to find 10 missing people. Can you solve the puzzle?

Missed! Reconnect Missing



Get tested with a missing person's name. Can you find the missing person?

Find Your Community



Connect with other people in your field and stay up to date on news and new developments.

When a Job Offer is Not an Offer



Learn why offers get declined and what you can do to get your offer back on track.

FIGURE 3F

9/167

facebook Messages Profile Network Jobs Post a Job Search Company Search Home Profile Find Friends Account



ZGROUP

Z Group Construction
www.zgroupco.com
Z Group Construction
Careers
Construction
Private Company
Founded: 1985
Size: 15+ employees

Network Stats
56 Followers
13 Employees
1 In Your Network

Flag Company


Z Group Construction
Headquarters in Washington D.C. 01223
Other locations Dallas, TX 75039

[Follow Company](#) [Claim Company](#) [Share](#)


Summary
Welcome to Z Group Construction where we care about our customers.

Our goal, with this site, is to provide insights on individual green construction methods which can be utilized on any residential or commercial project. Methods which can drastically improve energy efficiency, conserve natural resources, and preserve the environment. Learn key green home construction principles – from energy saving construction to solar energy producing to water conservation.... More

Your Connections at Z Group Construction



Rick Steele
General Contractor
April 1999 – Present (12 years)
[Send Message](#)

Updates
Electrician
Z Group Construction • Washington, DC 01223
Z Group Construction is an environmental friendly construction company. We are looking for a highly experienced electrician who has a positive can-do attitude, can work closely with the...

Rick Steele posted this job.
General Contractor
May 12 at 6:15pm • Share

Jobs

Sponsored [Create an Ad](#)

Plans Submit To New Listing
Time is running out to apply! Do you have what it takes to be Planer's new Profile Coordinator?

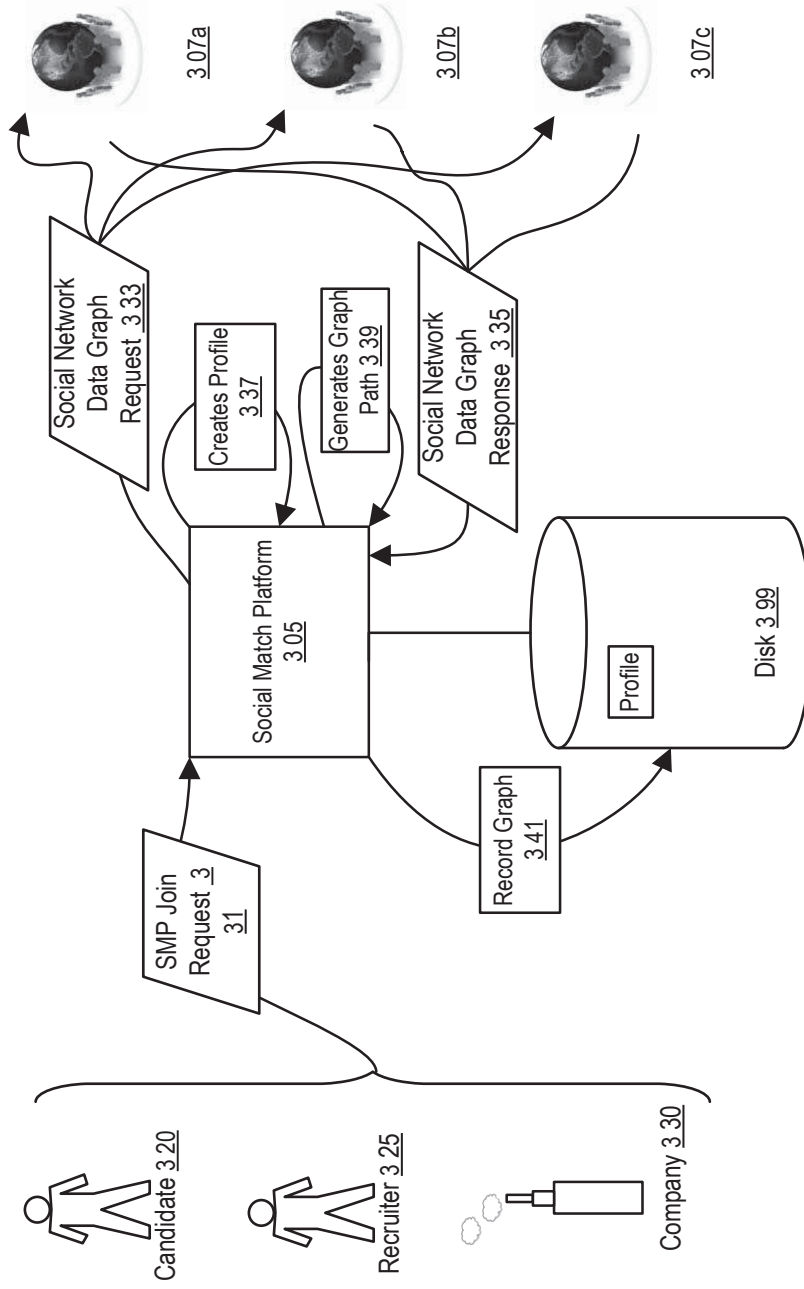

Member Resource Meeting
Get brief with a job-winning resource crafted by our experts.


Find Your Community
Connect with other people in your field and stay up-to-date on issues and new developments.


When a Job Offer is Not an Offer
Learn why offers get studied and what you can do to get your offer back on track.


10/167

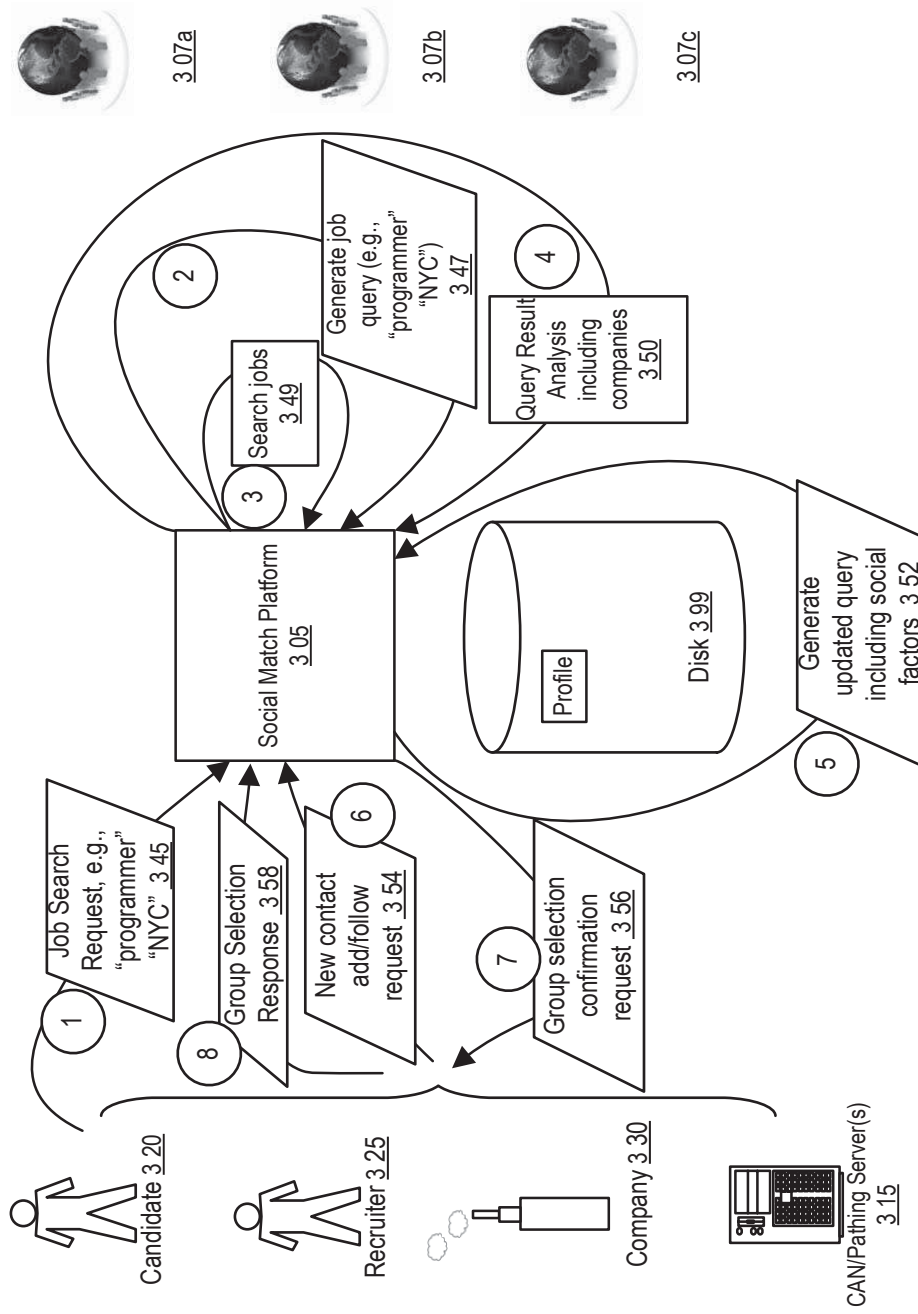
FIGURE 3G



EXEMPLARY JOIN REQUEST DATA FLOW

11/167

FIGURE 3H



EXEMPLARY SEARCH DATA FLOW

12/167

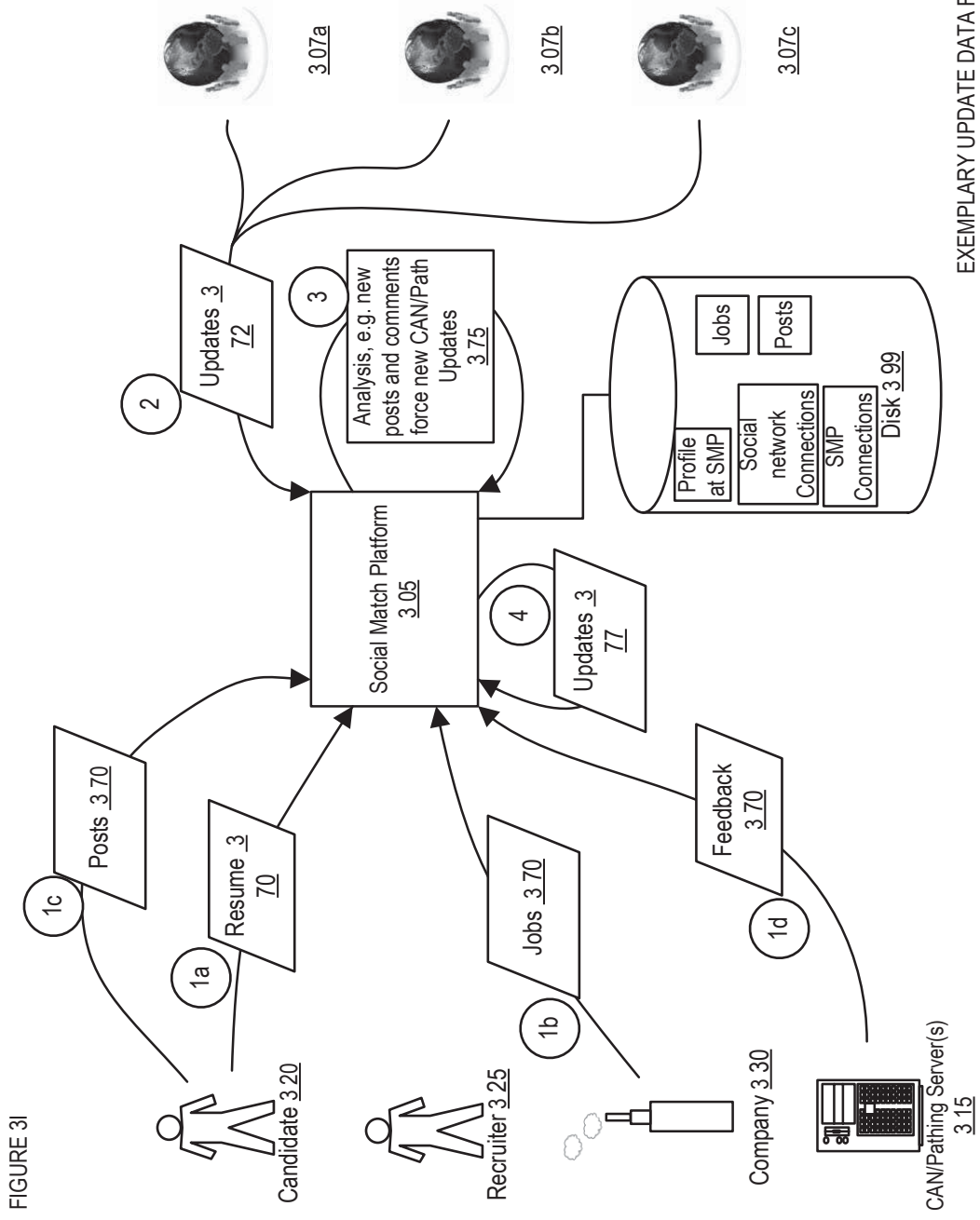


FIGURE 3I

EXEMPLARY UPDATE DATA FLOW

FIGURE 4

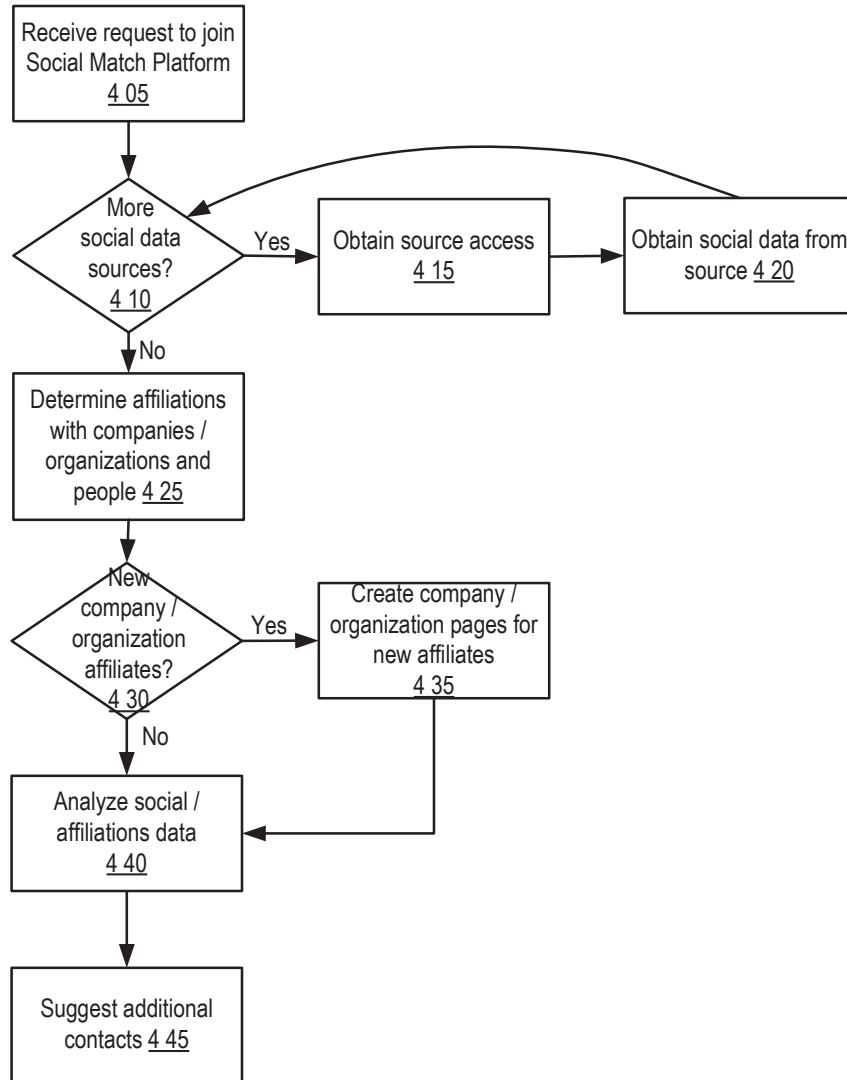
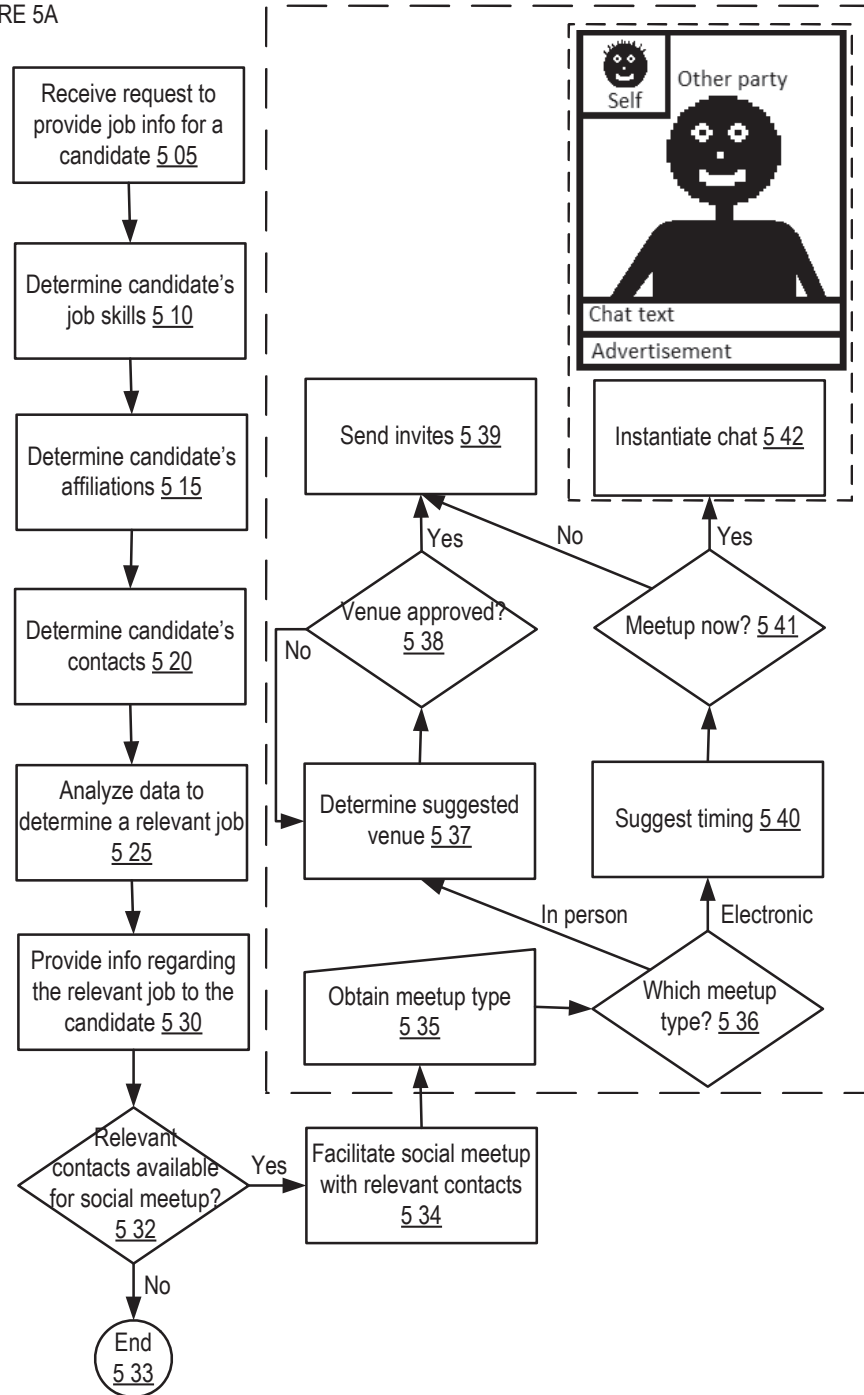
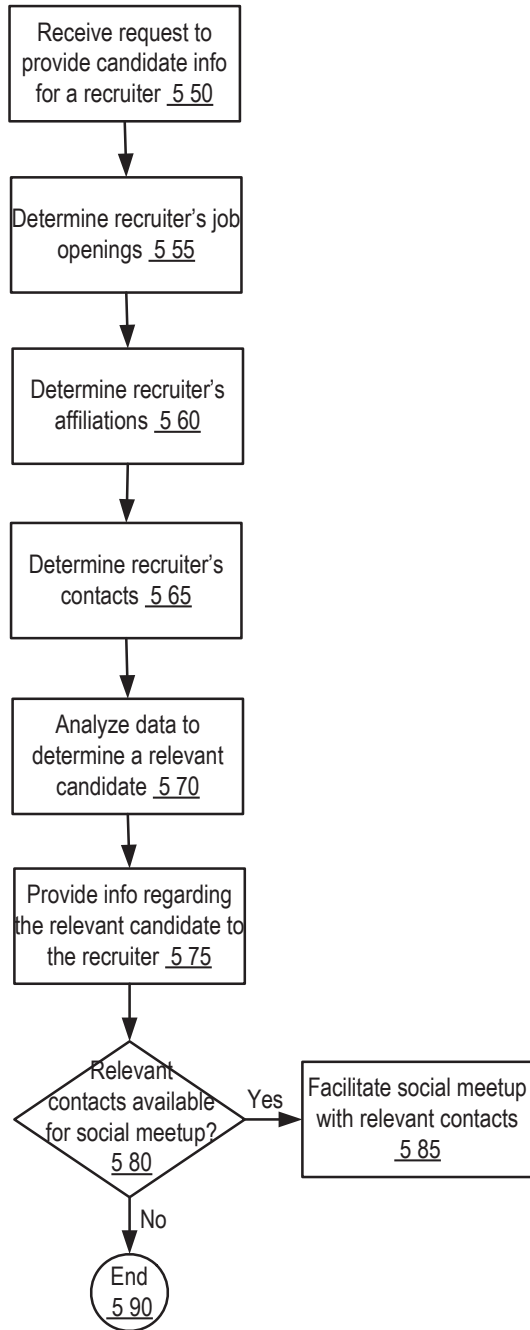


FIGURE 5A



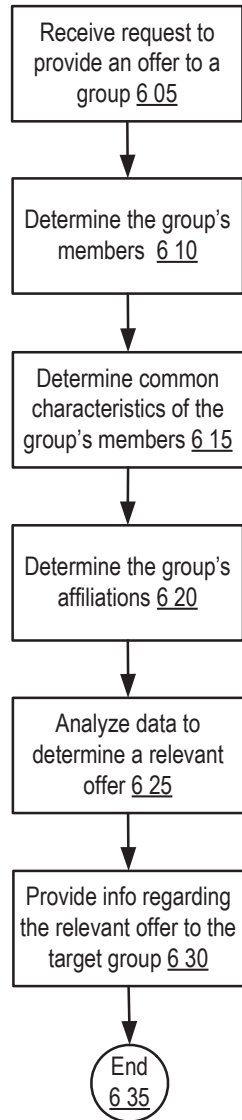
EXEMPLARY SOCIAL MATCH PLATFORM JOB INFO PROVIDING (JIP) COMPONENT

FIGURE 5B



EXEMPLARY SOCIAL MATCH PLATFORM CANDIDATE INFO PROVIDING (CIP) COMPONENT

FIGURE 6



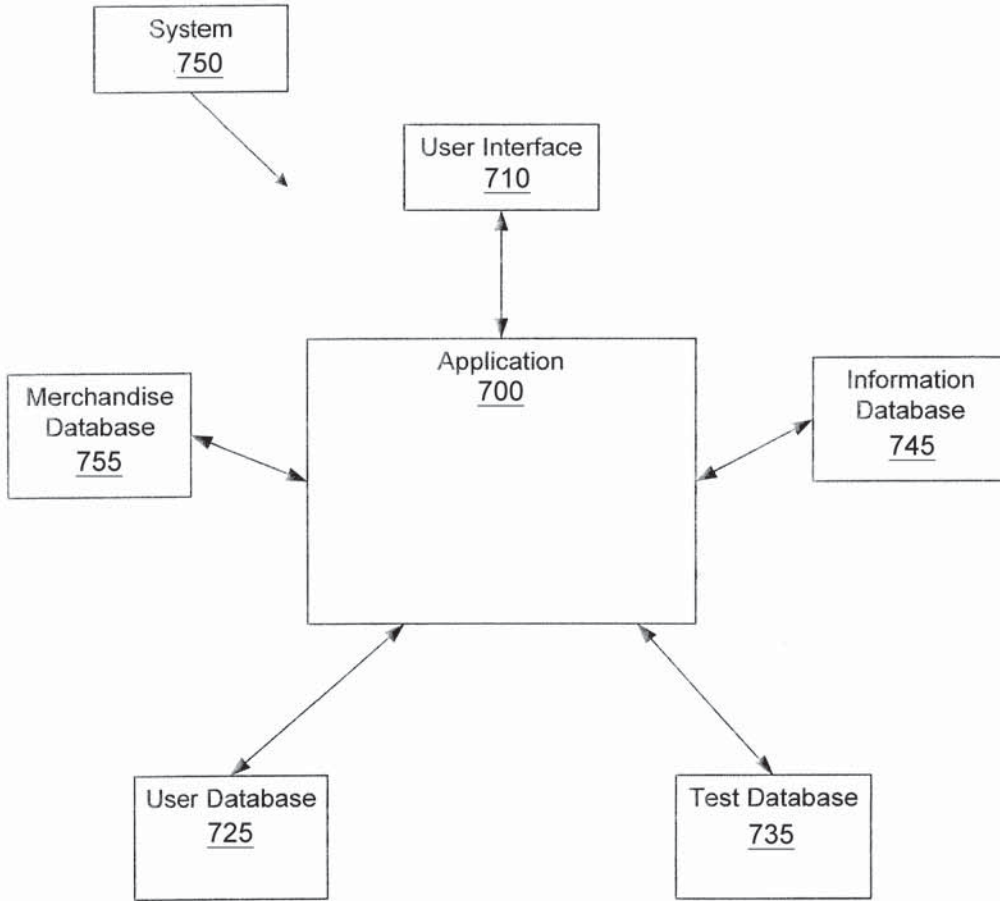


Fig. 7

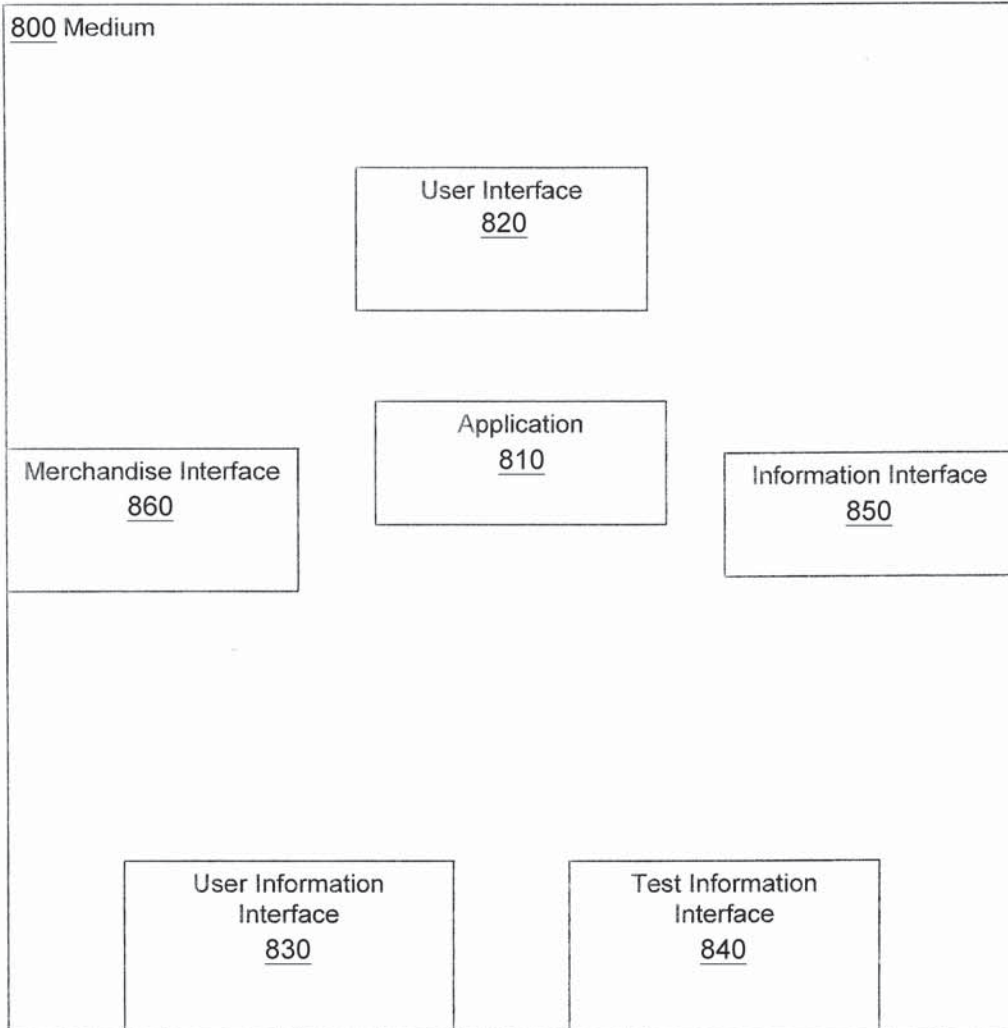


Fig. 8

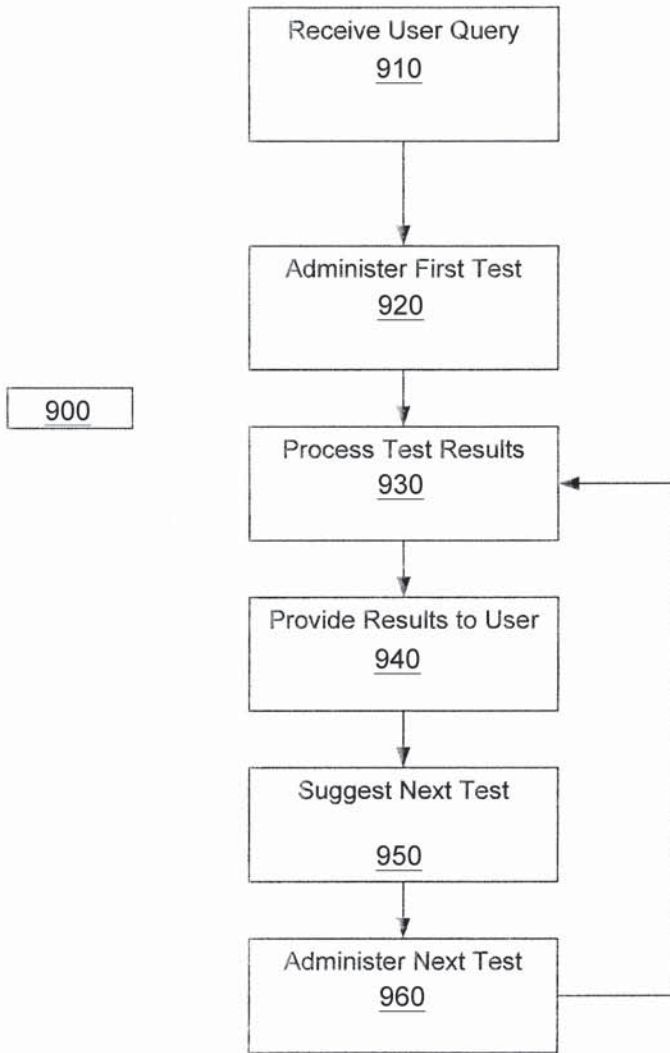
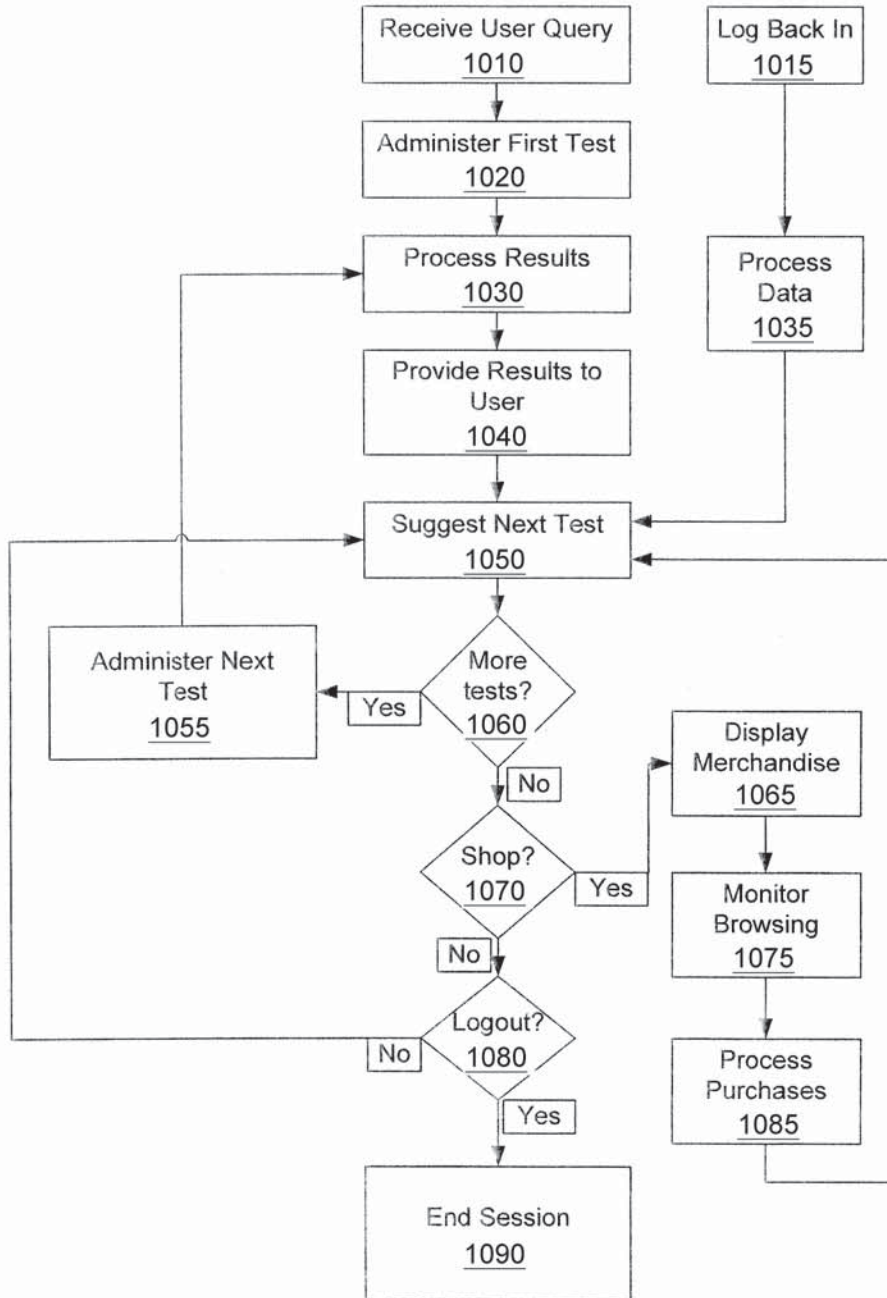
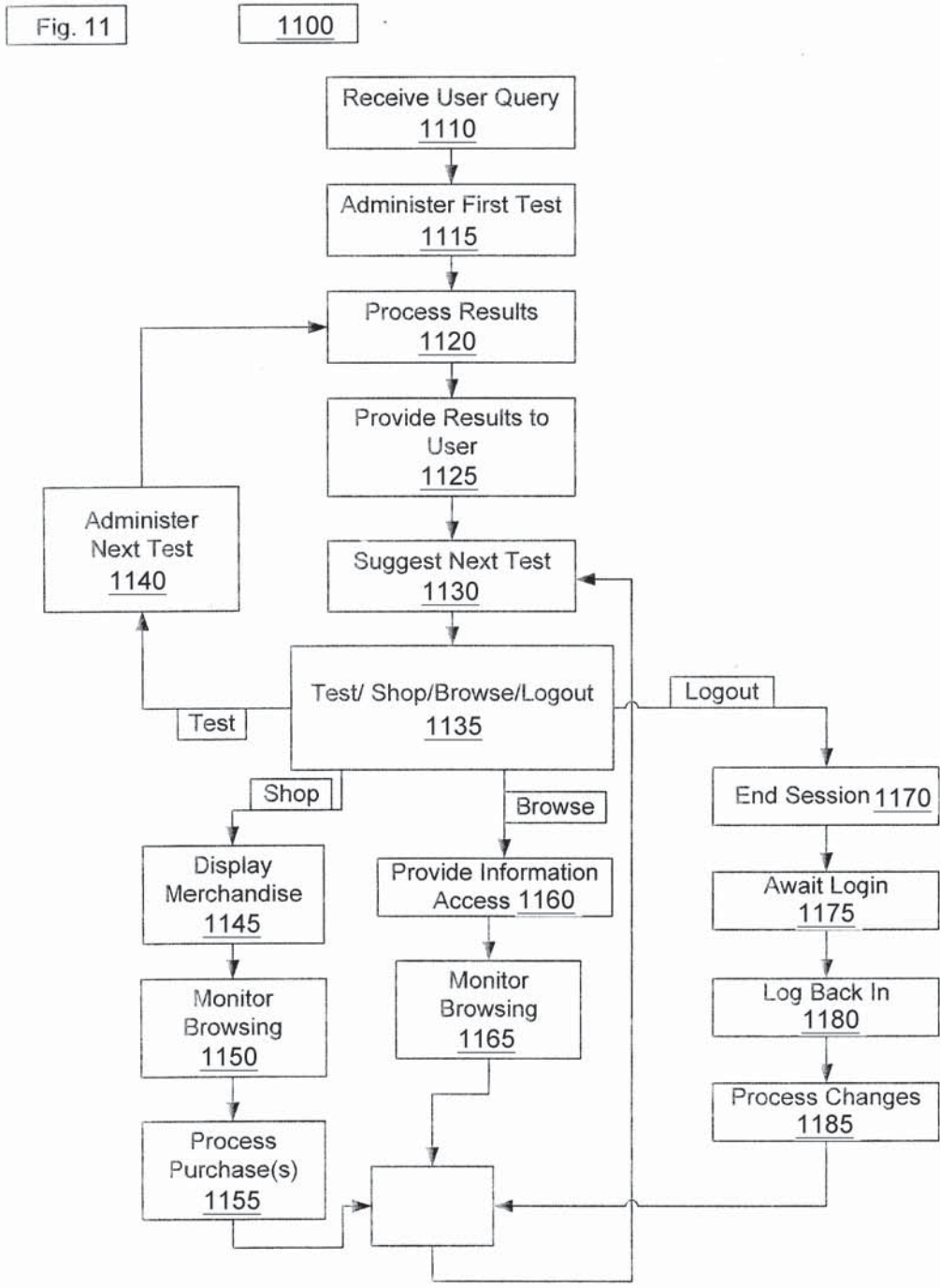


Fig. 9

Fig. 10

1000





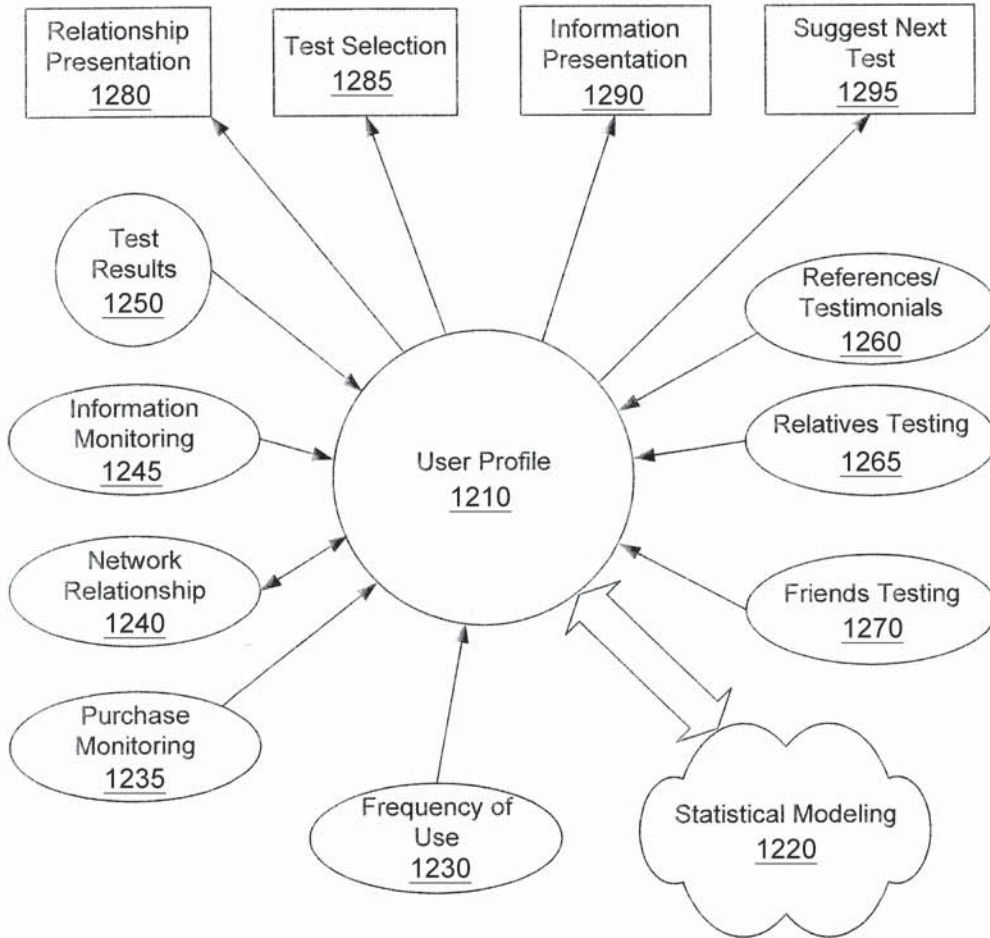
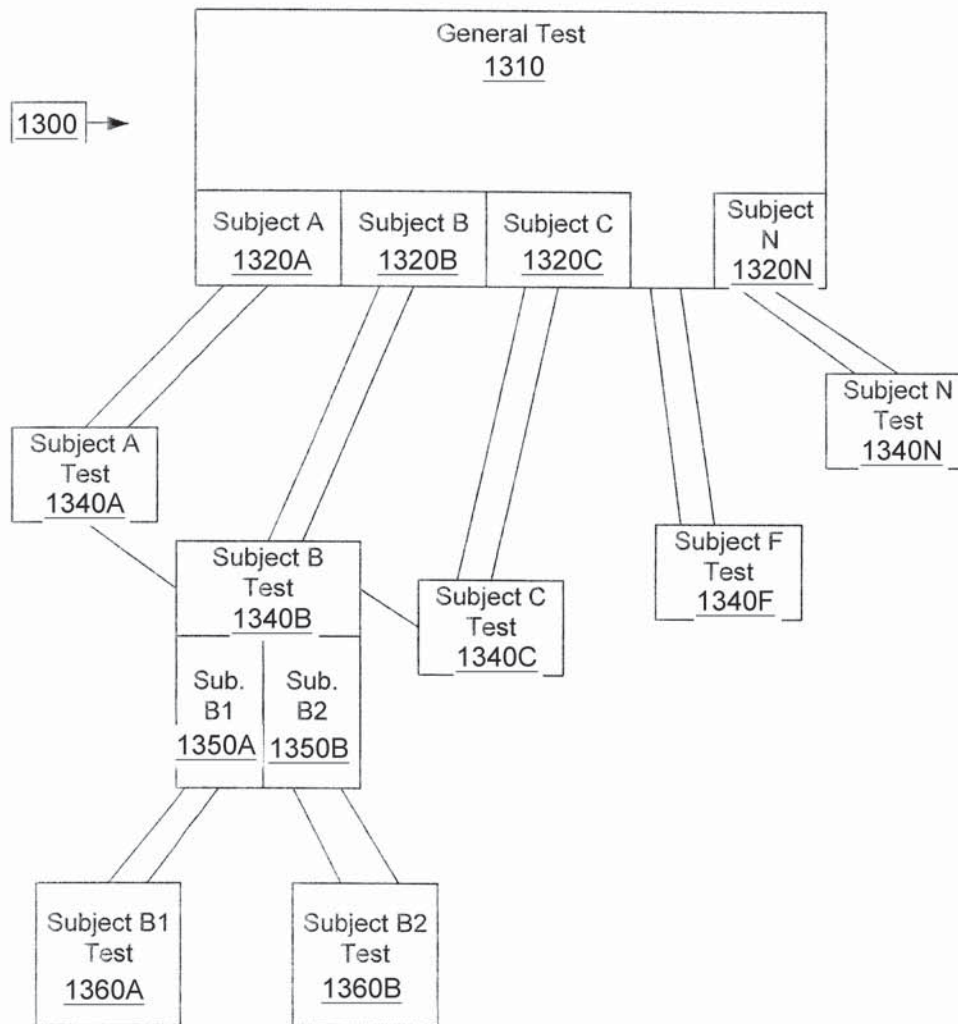


Fig. 12

Fig. 13



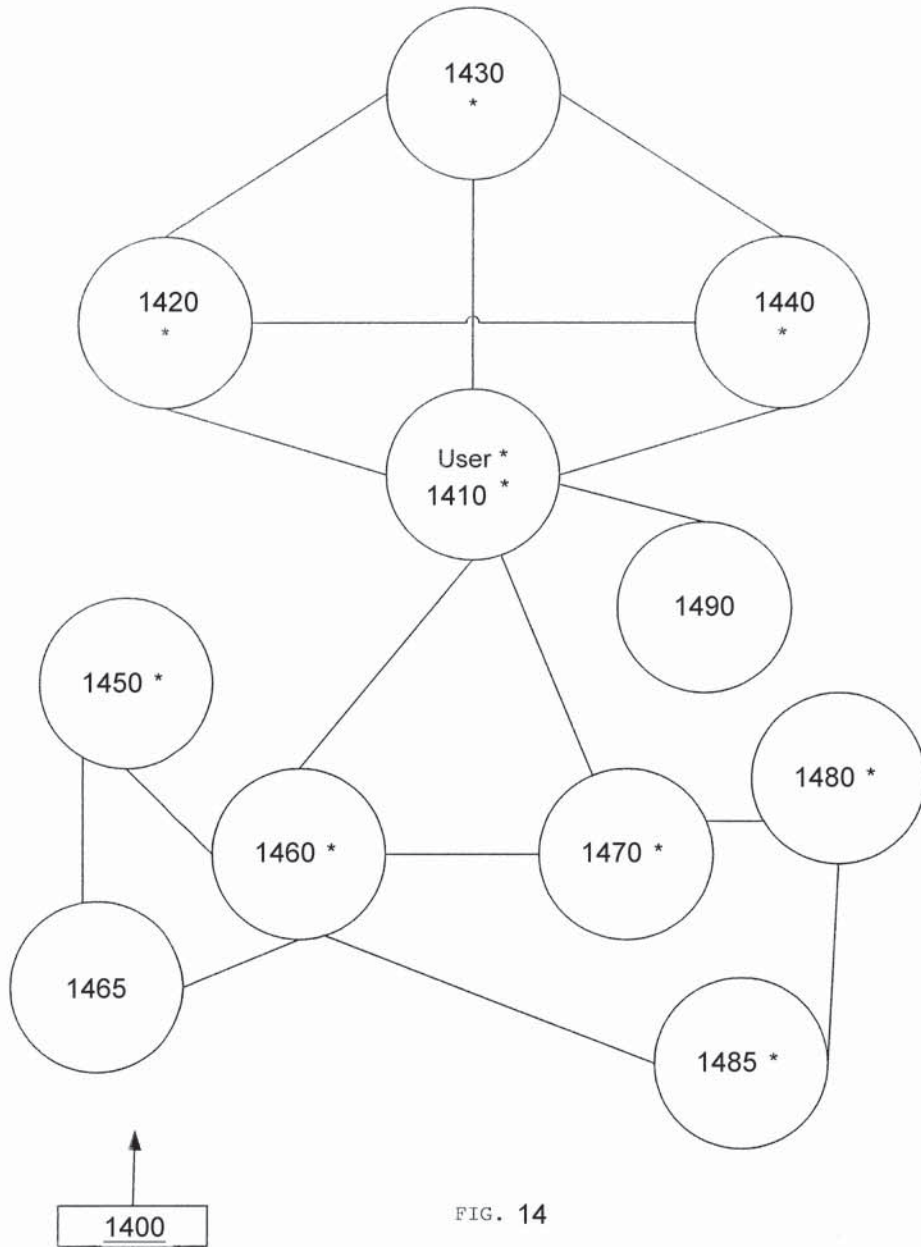


FIG. 14

Fig. 15

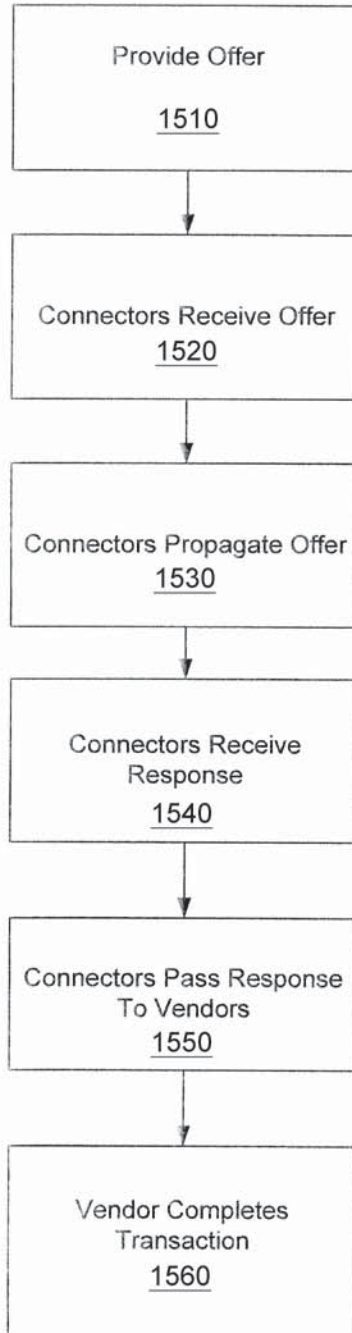
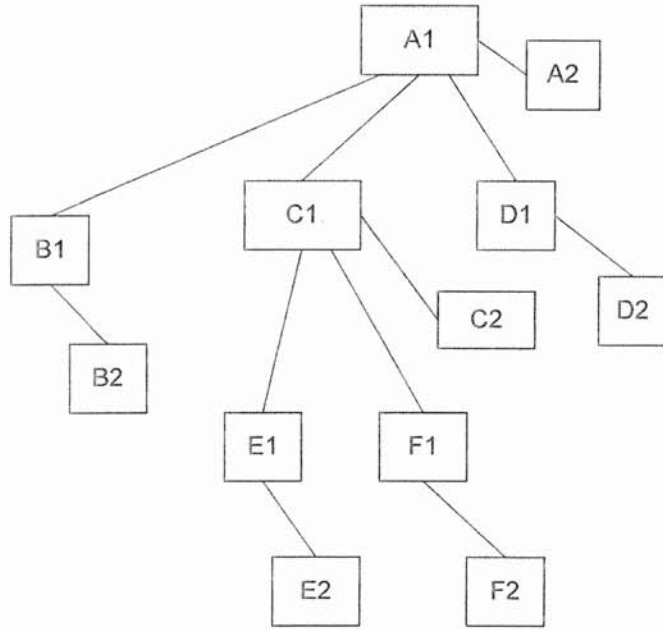


Fig. 16



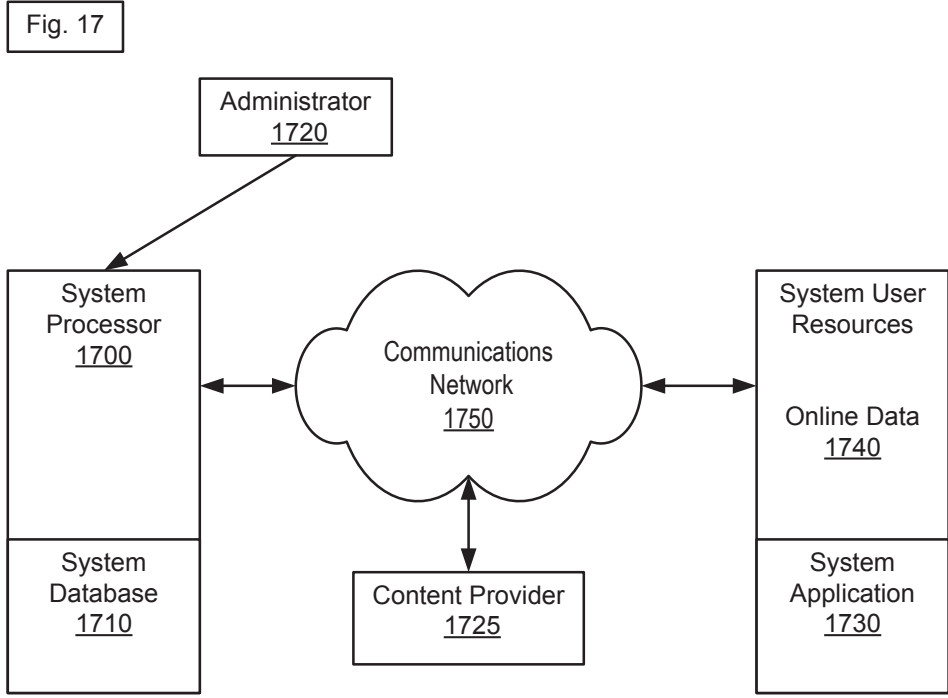
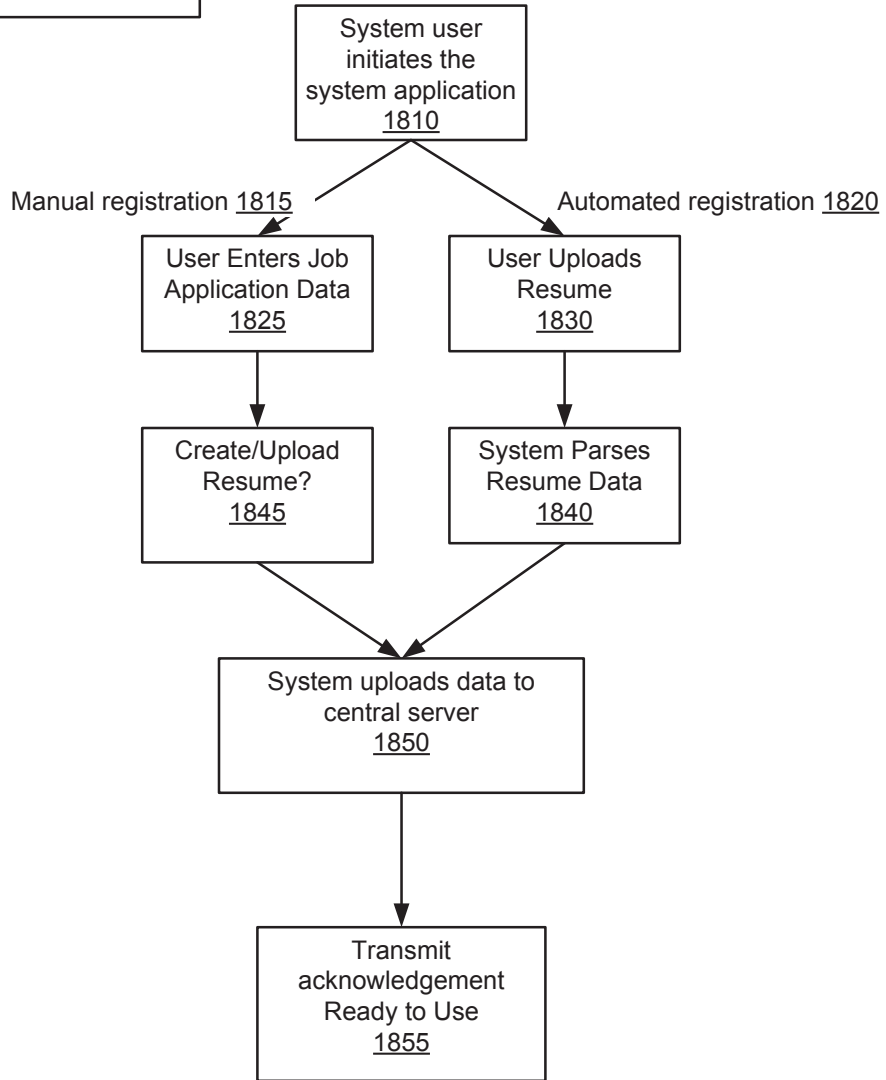
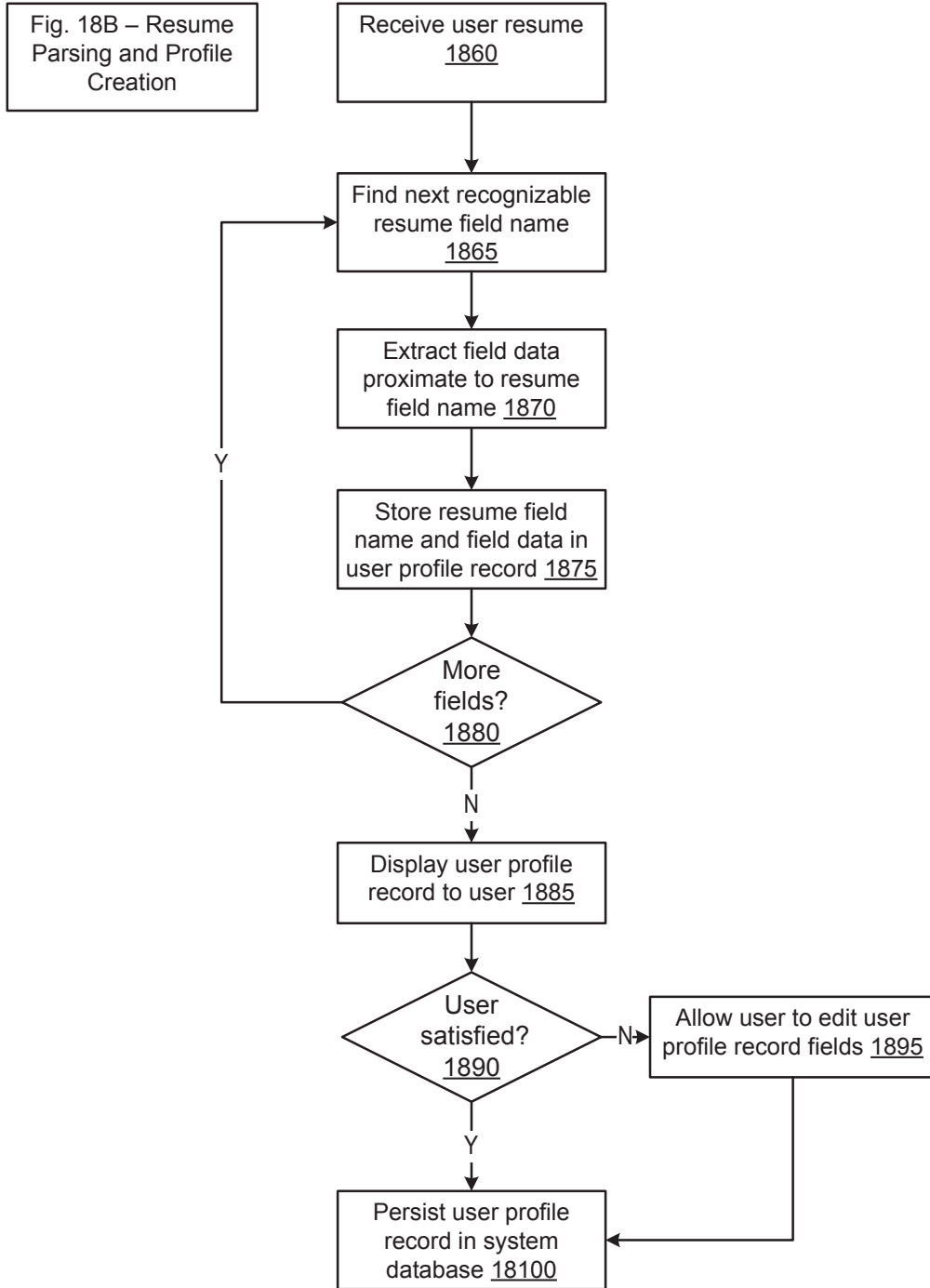


Fig. 18A - System User Data Registration





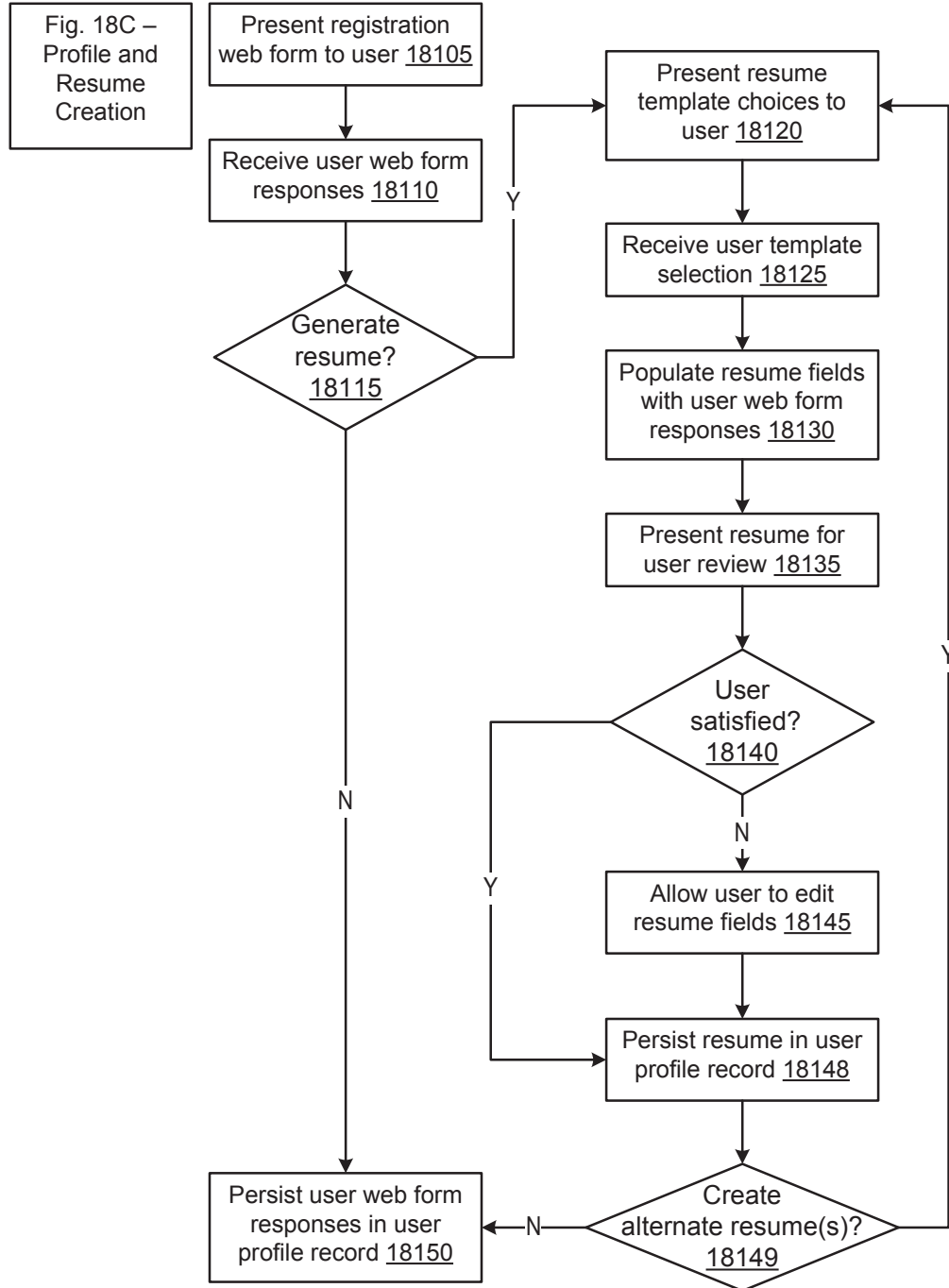
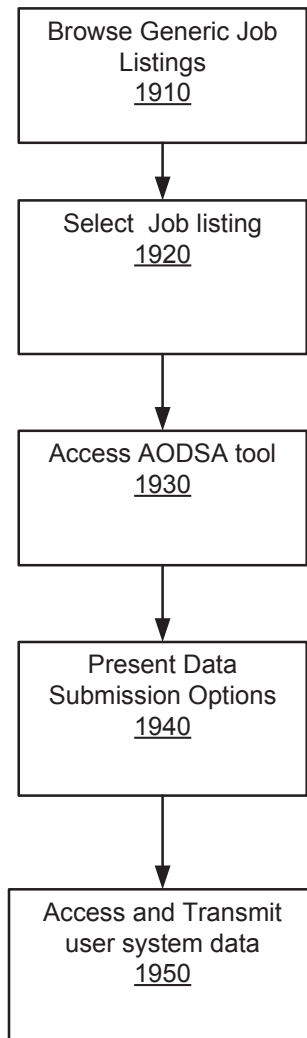
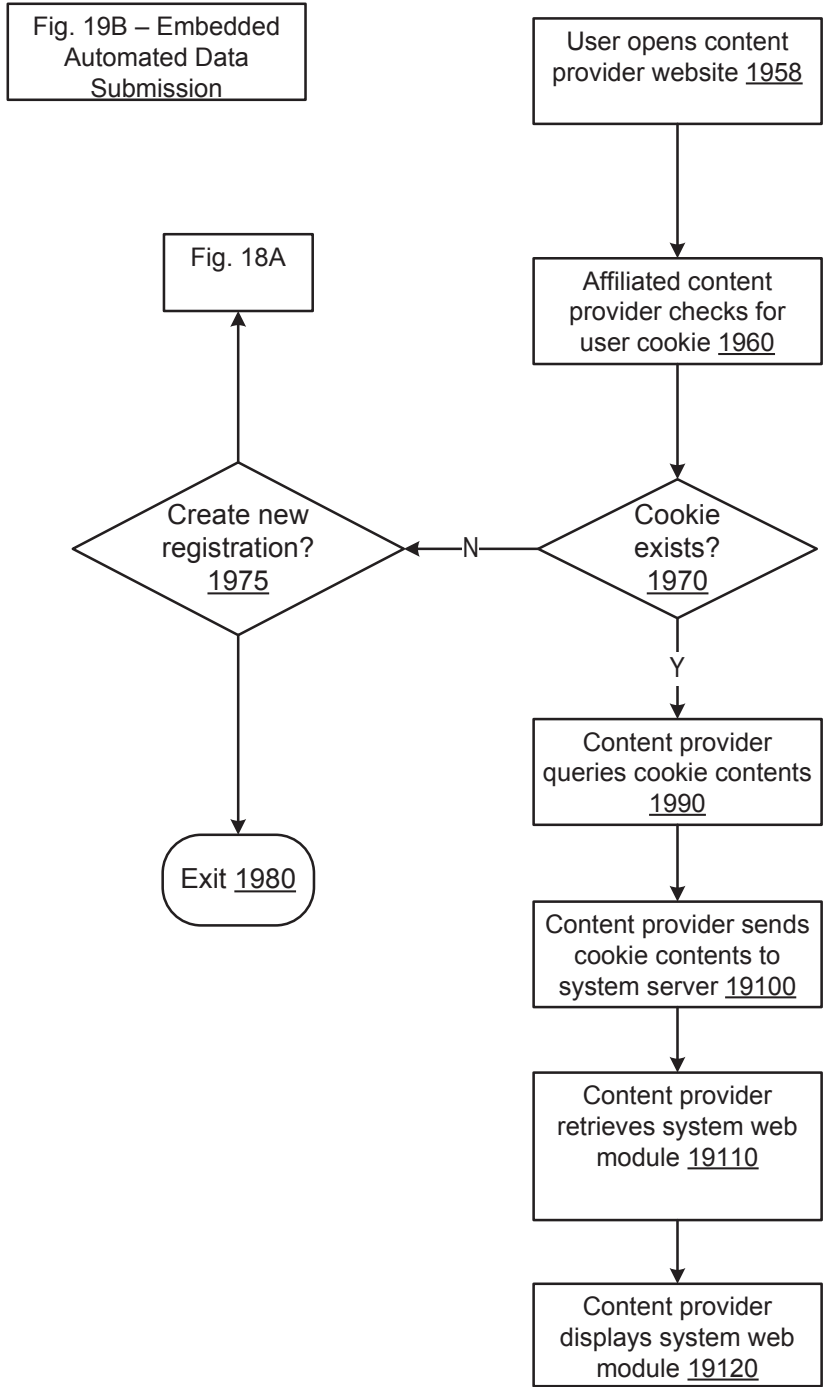
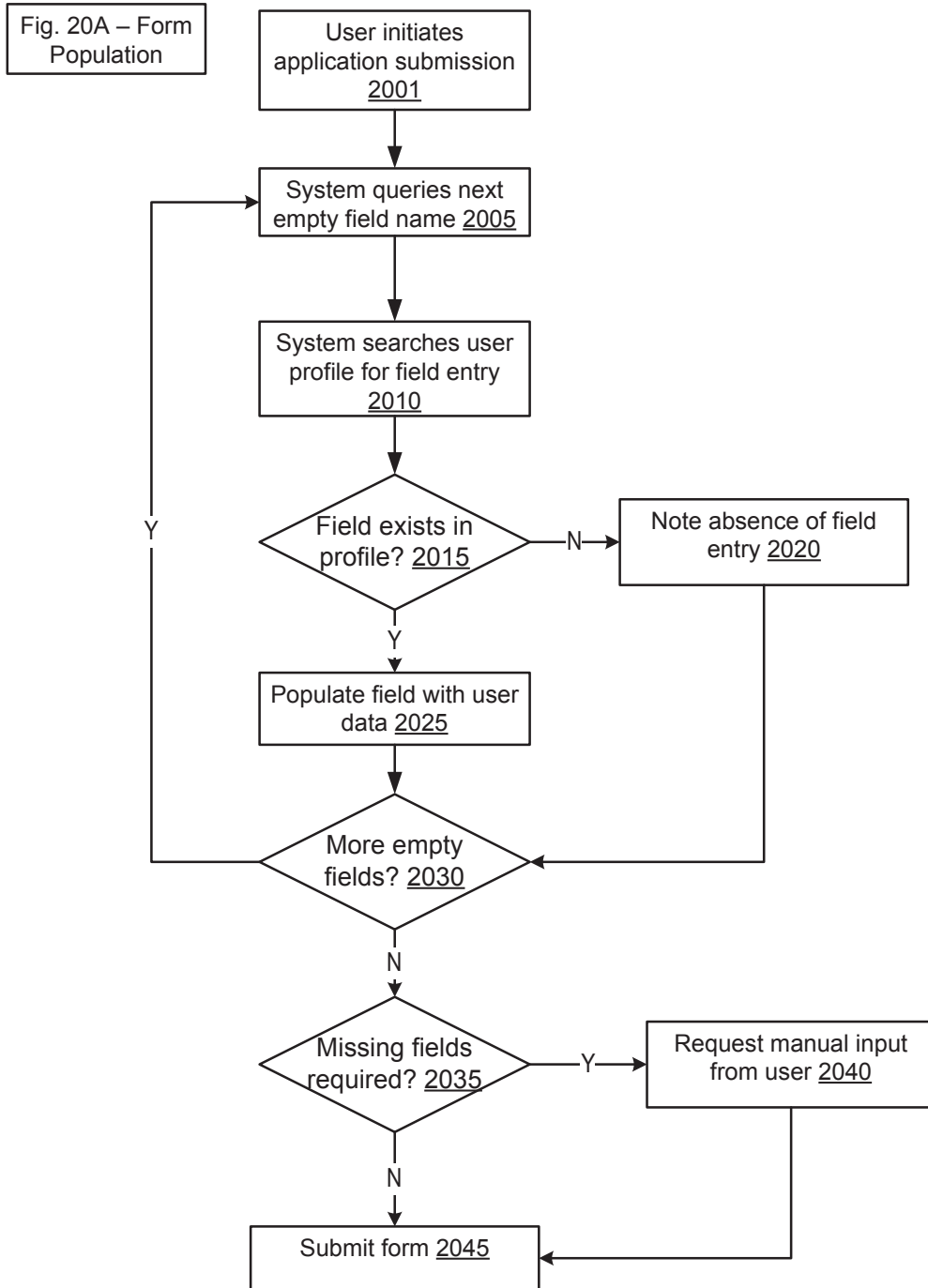


Fig. 19A – Autonomous Automated Data Submission







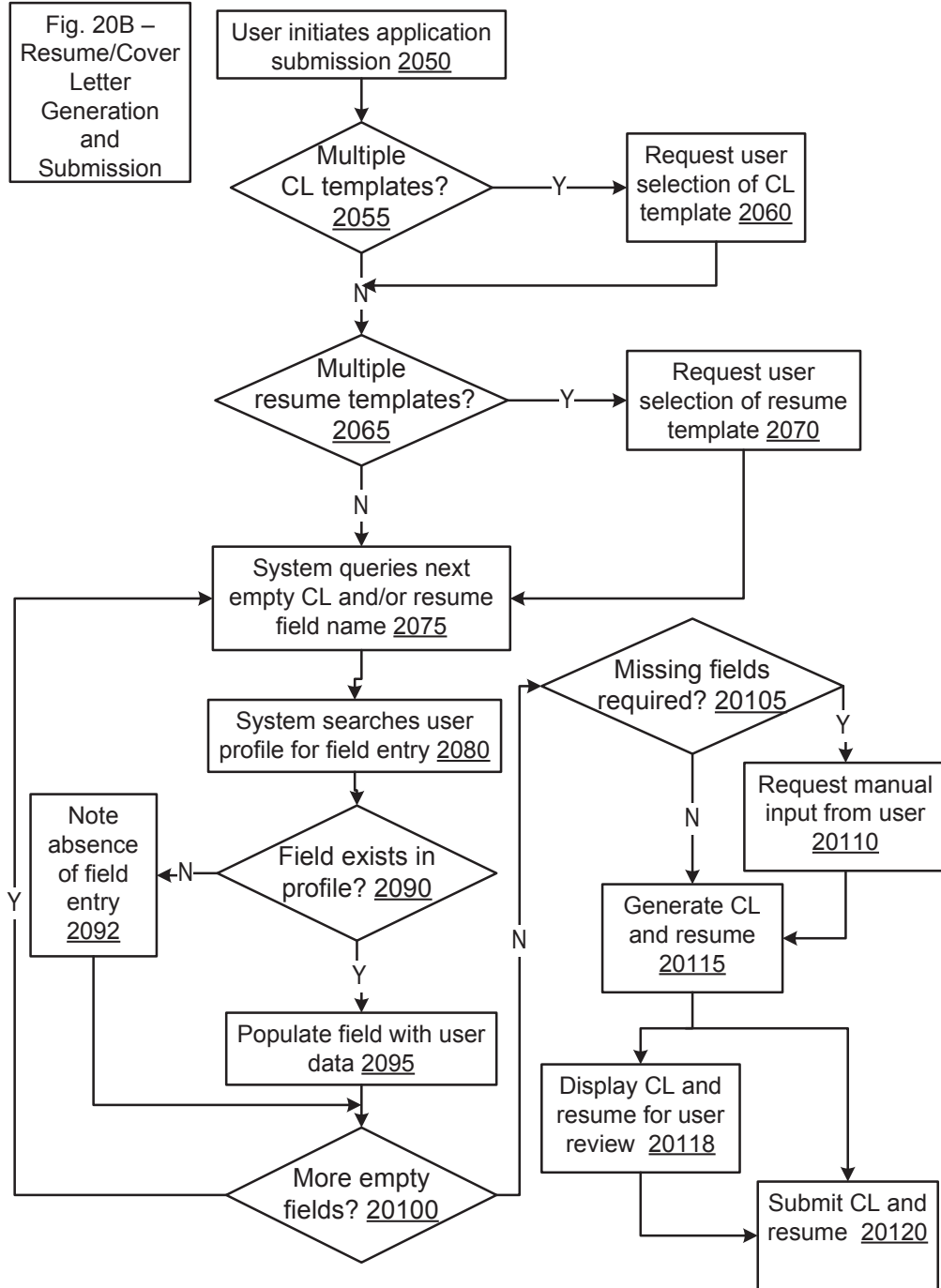


Fig. 21A

Generic Job Listing
2100

Software Engineer - 3yrs Exp.

Software Engineer - Java Programmer 2105

Embedded Software Programmer

C++ Programmer - 2yrs Exp.

Fig. 21B

Software Engineer - Java Programmer 2105

Wanted - Experienced Java programmer for e-commerce web developing projects. JavaScript language experience is preferred, as is previous web development experience.

Please forward resume to JSmith@Acme.com

OR

Click here to apply:
www.ibm.com/jobapplication.htm

<u>AODSA Tool 2110</u>	
Submit Resume	<u>2120</u>
Auto-Submit ID Data	<u>2130</u>
Auto-Email Resume	<u>2140</u>
Update Resume Data	<u>2150</u>

21100

Auto-Email Resume: 2140
 Please select:

Resume type

Software Engineering 2160

Java Programming 2165

Combination 2170

Custom 2175

Cover letter type

Specific 2180

General 2185

Professional 2190

Custom 2195

Fig. 21C

Software Engineer - Java Programmer 2105

Wanted - Experienced Java programmer for e-commerce web developing projects. JavaScript language experience is preferred, as is previous web development experience.

Please forward resume to JSmith@Acme.com

OR

Click here to apply:
www.ibm.com/jobapplication.htm

AODSA Tool 2110

Submit Resume 2120

Auto-Submit ID Data 2130

Auto-Email Resume 2140

Update Resume Data 2150

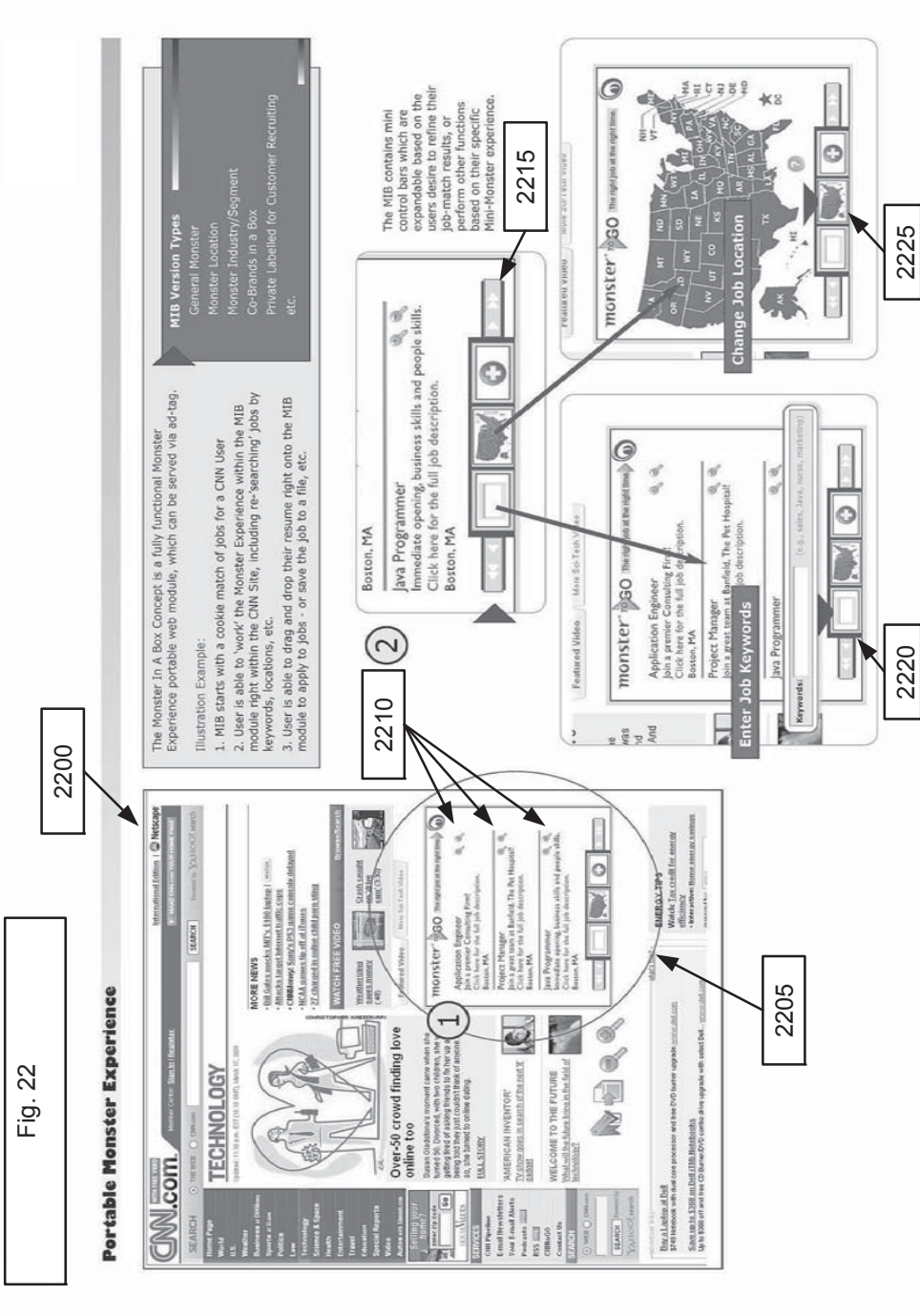


Fig. 23

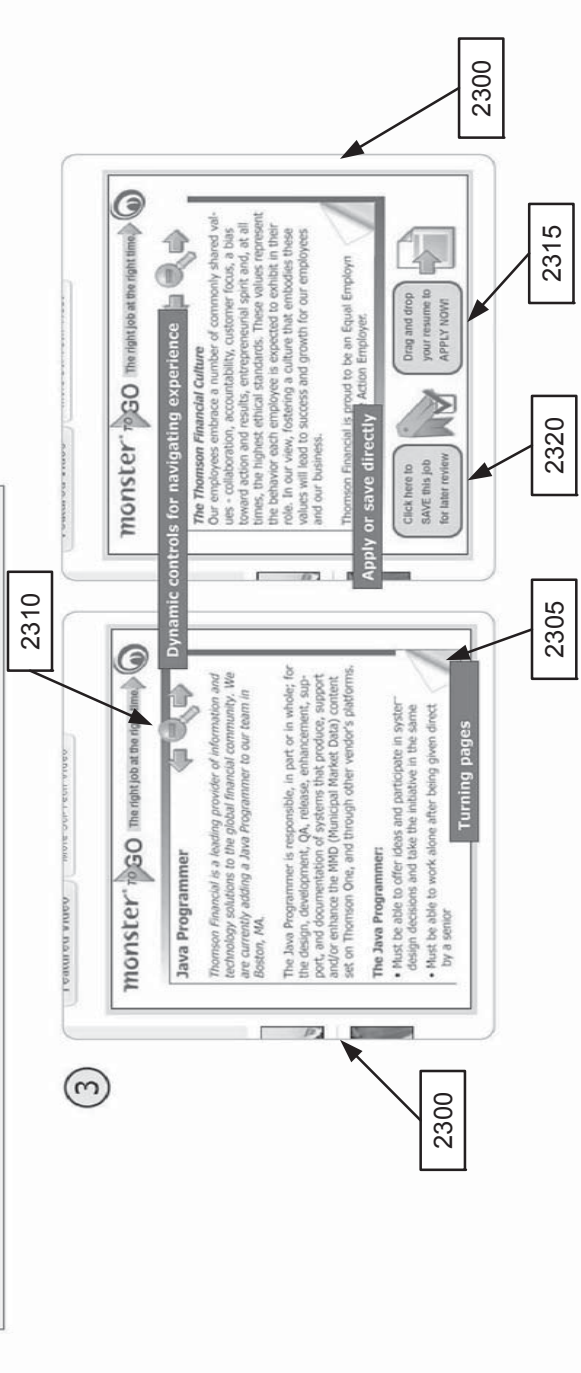
The Monster In A Box Concept is a fully functional Monster Experience portable web module.

Illustration Example Continued:

The user is able to zoom in and out of job viewing right within the module including "turning the pages" of job descriptions or company information pages.

The User will be able to do a number of functions on the spot including:

- Applying to the job by: using their Monster resume, dragging and dropping their word (et. al) resume or pdf resume, or using any other digital resume that they may have created (9pp-app).
- Saving the job(s) to view/apply to later.
- Saving the company info/content to review later.



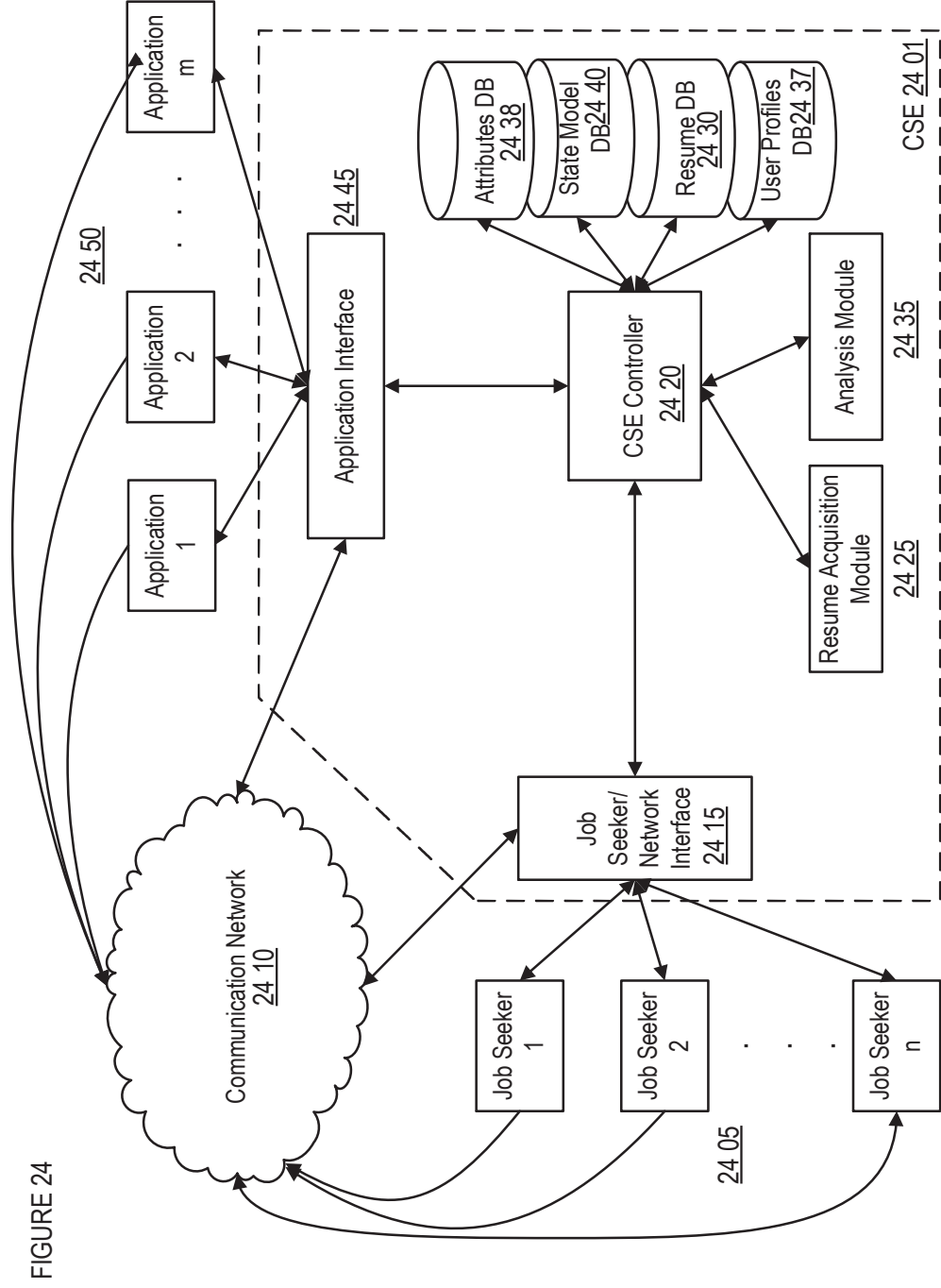


FIGURE 24

FIGURE 25

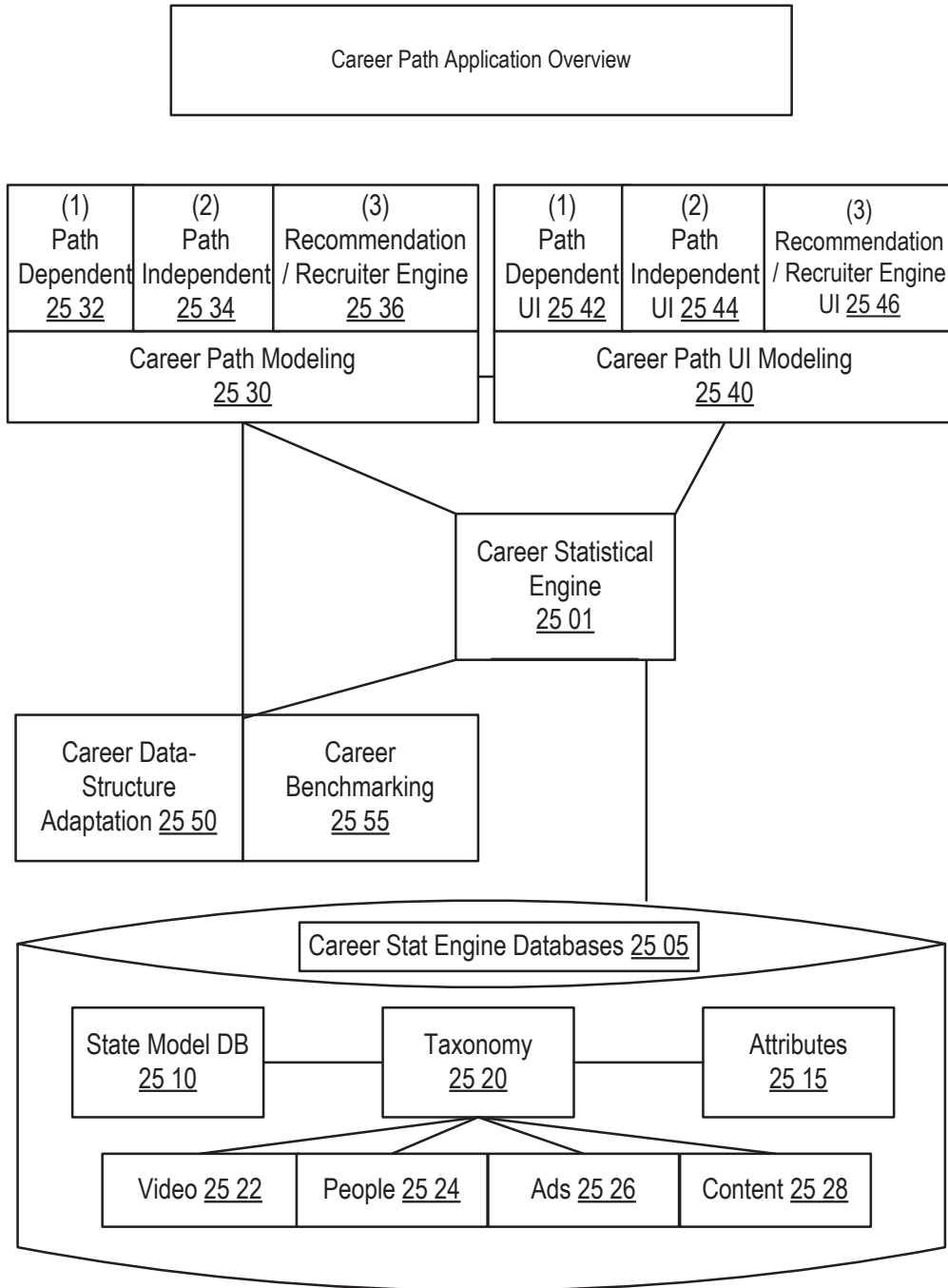


FIGURE 26A

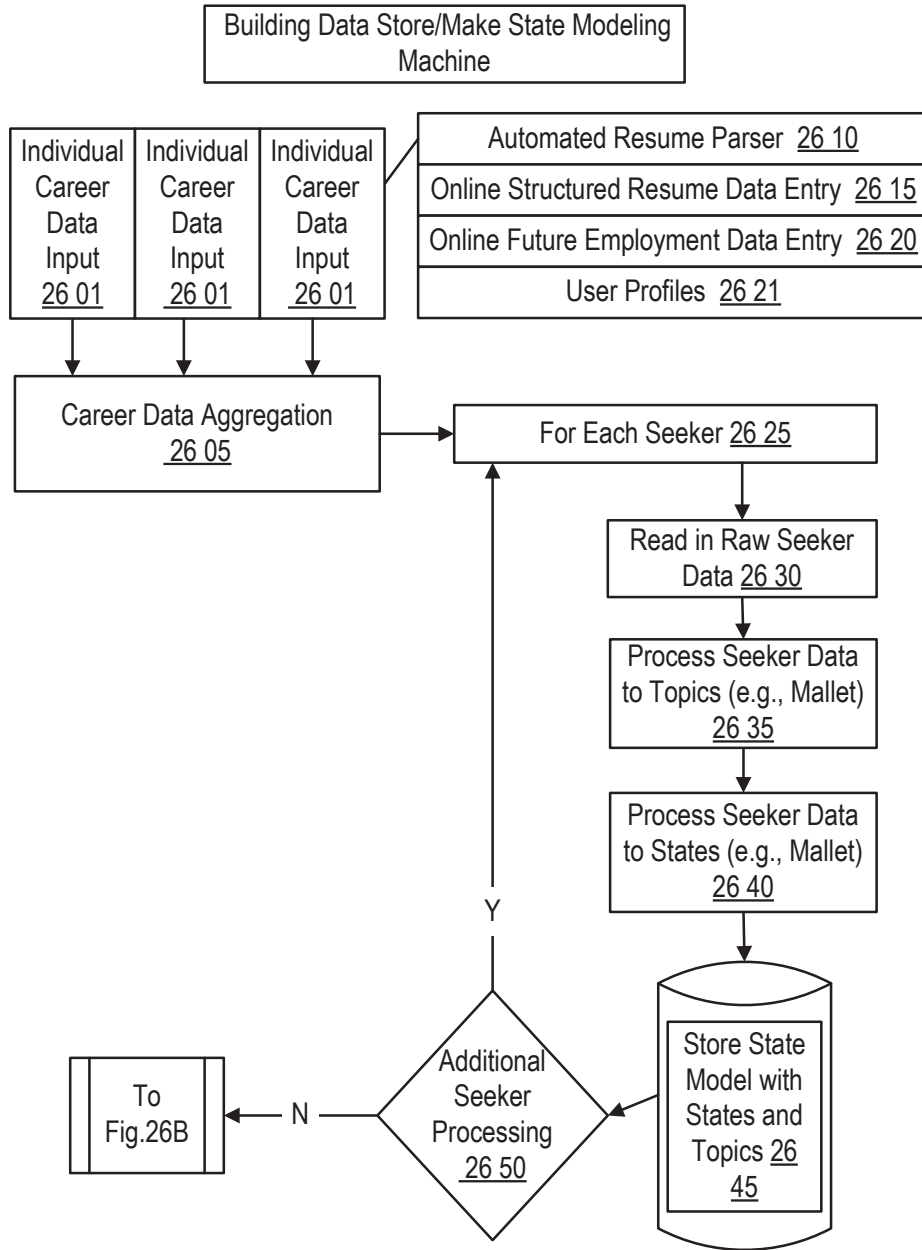
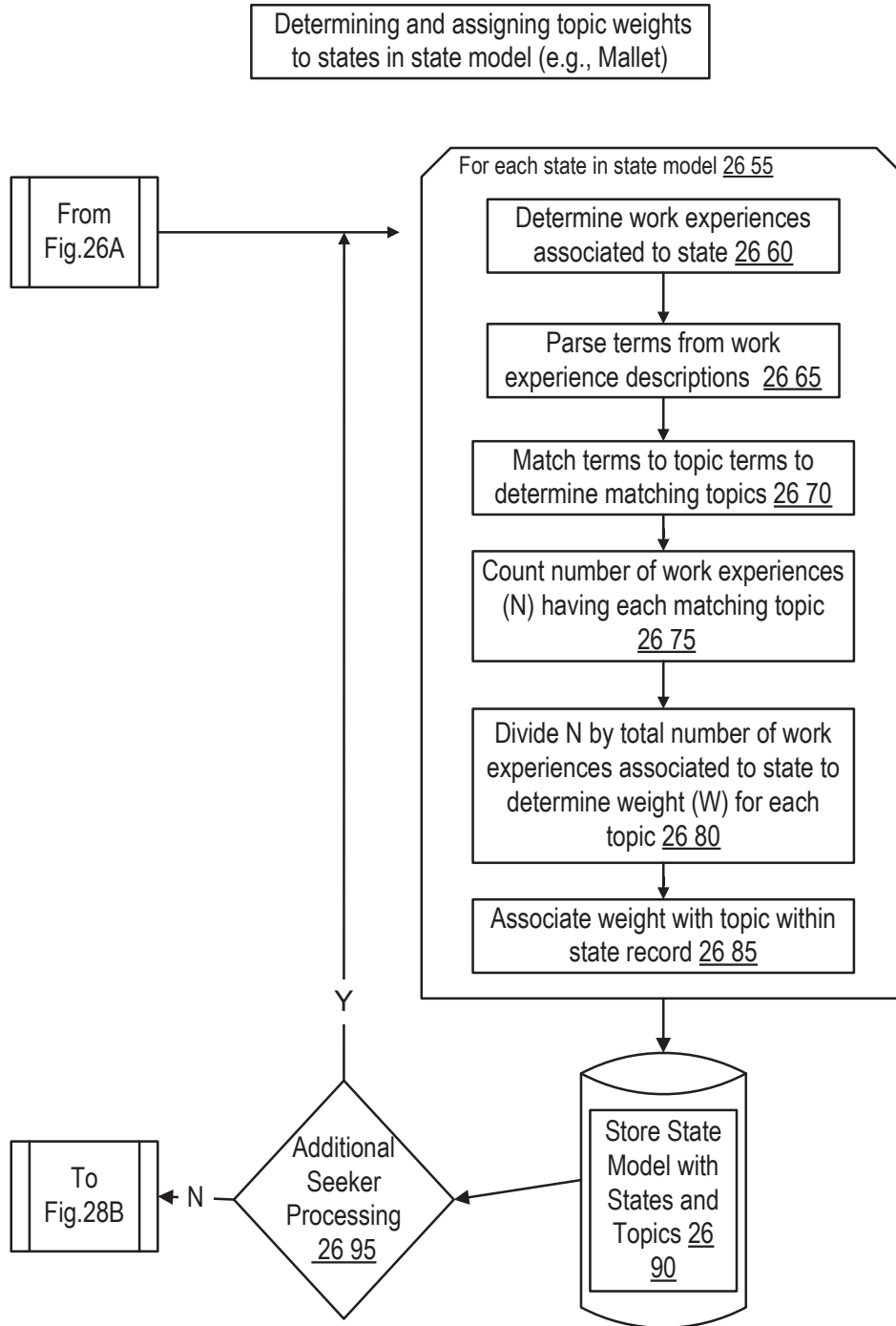


FIGURE 26B



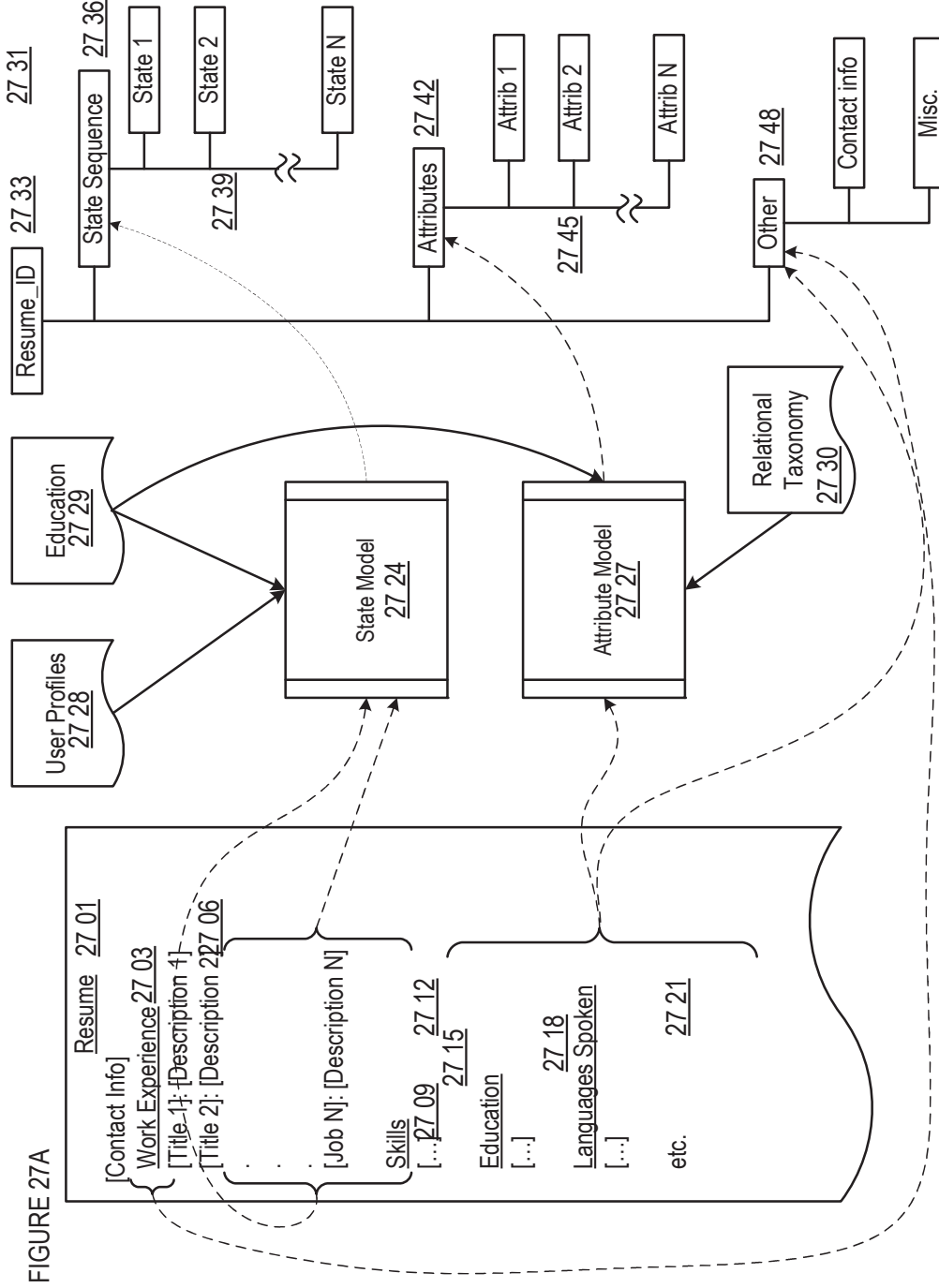


FIGURE 27A

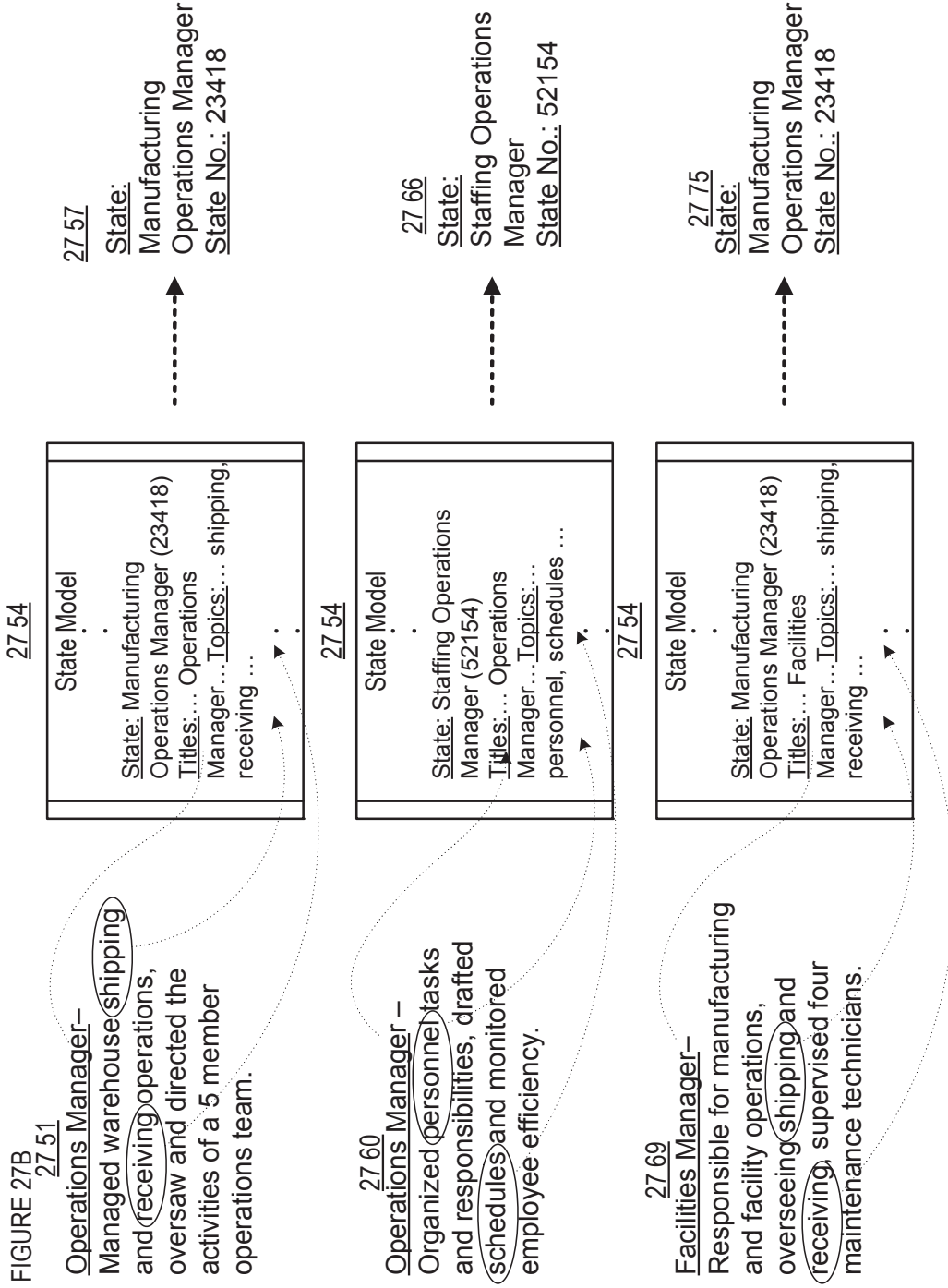


FIGURE 27C

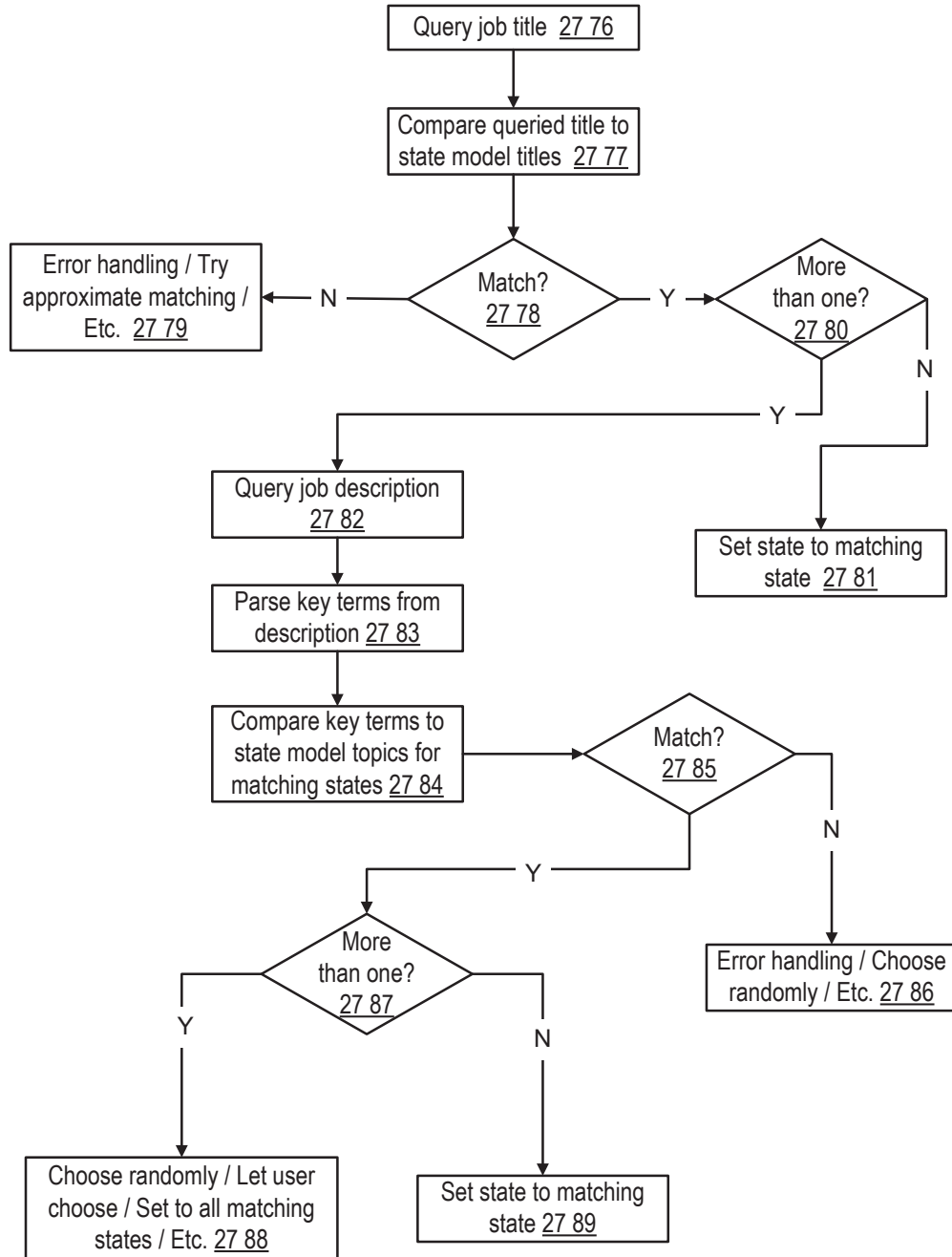


FIGURE 27D

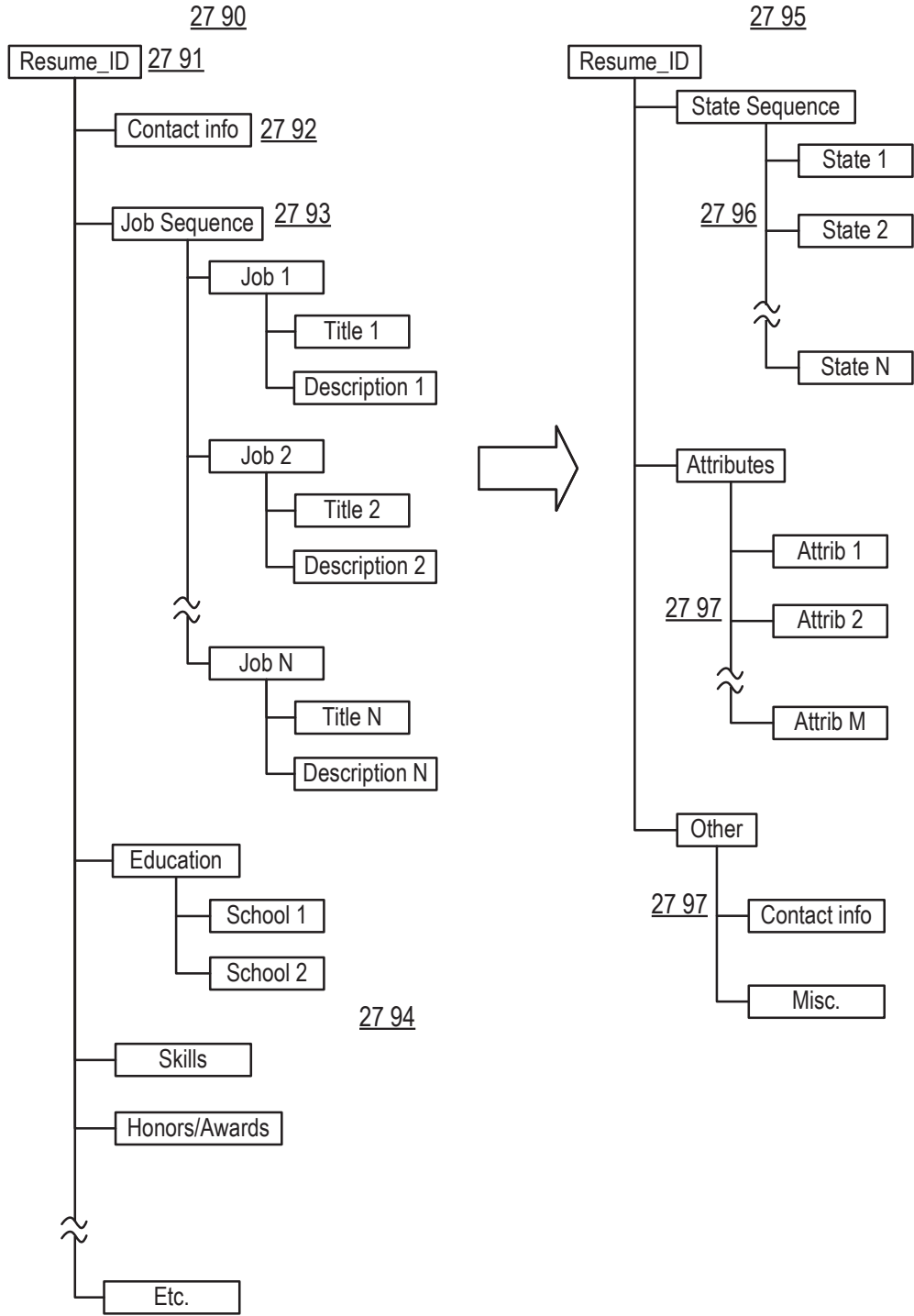


FIGURE 28A

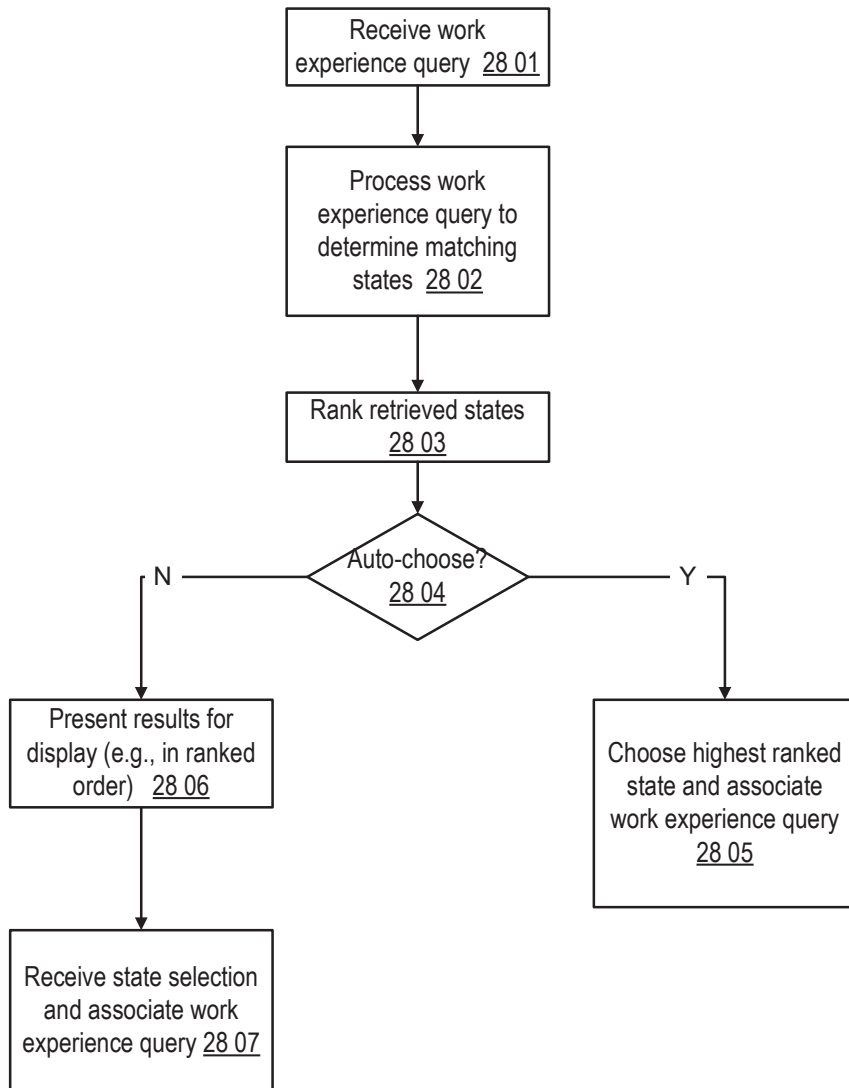


FIGURE 28B

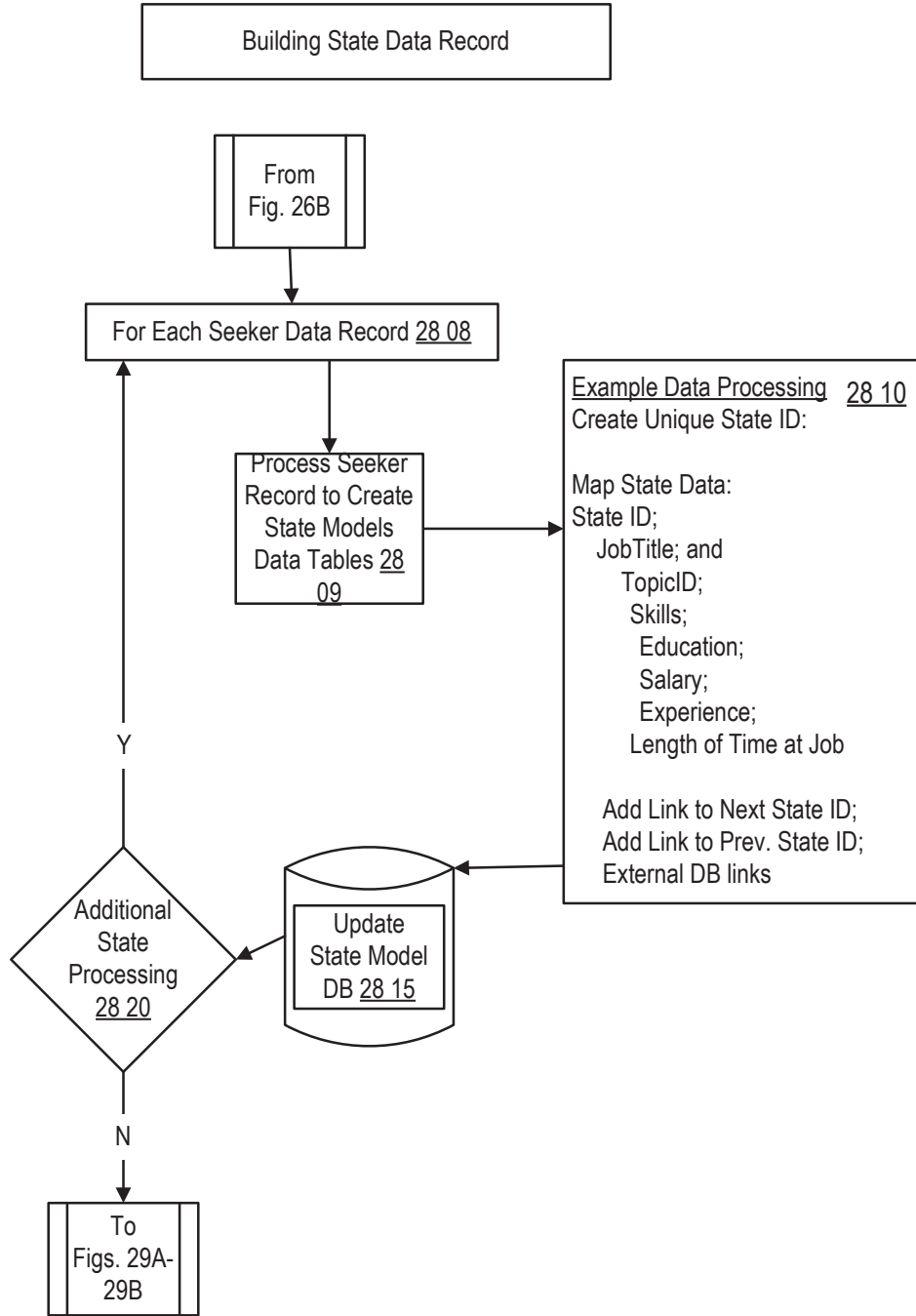


FIGURE 29A

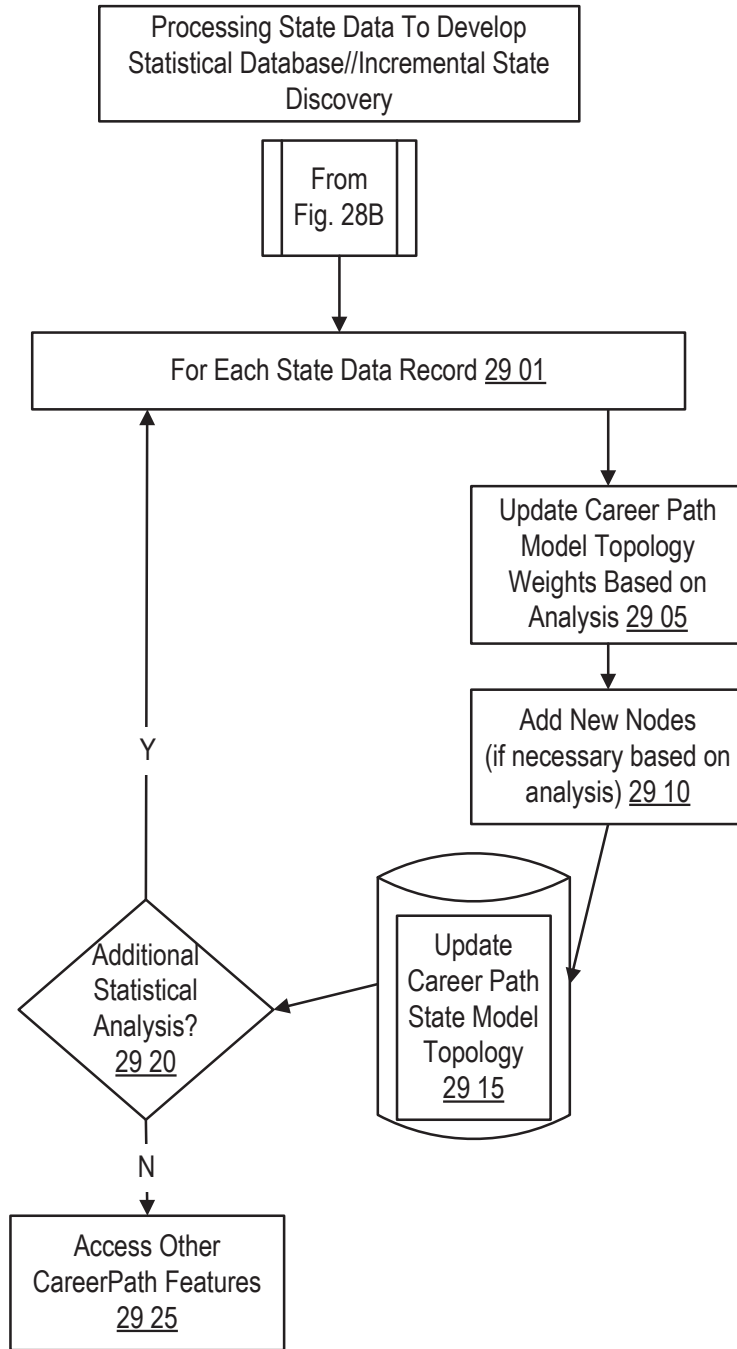


FIGURE 29B

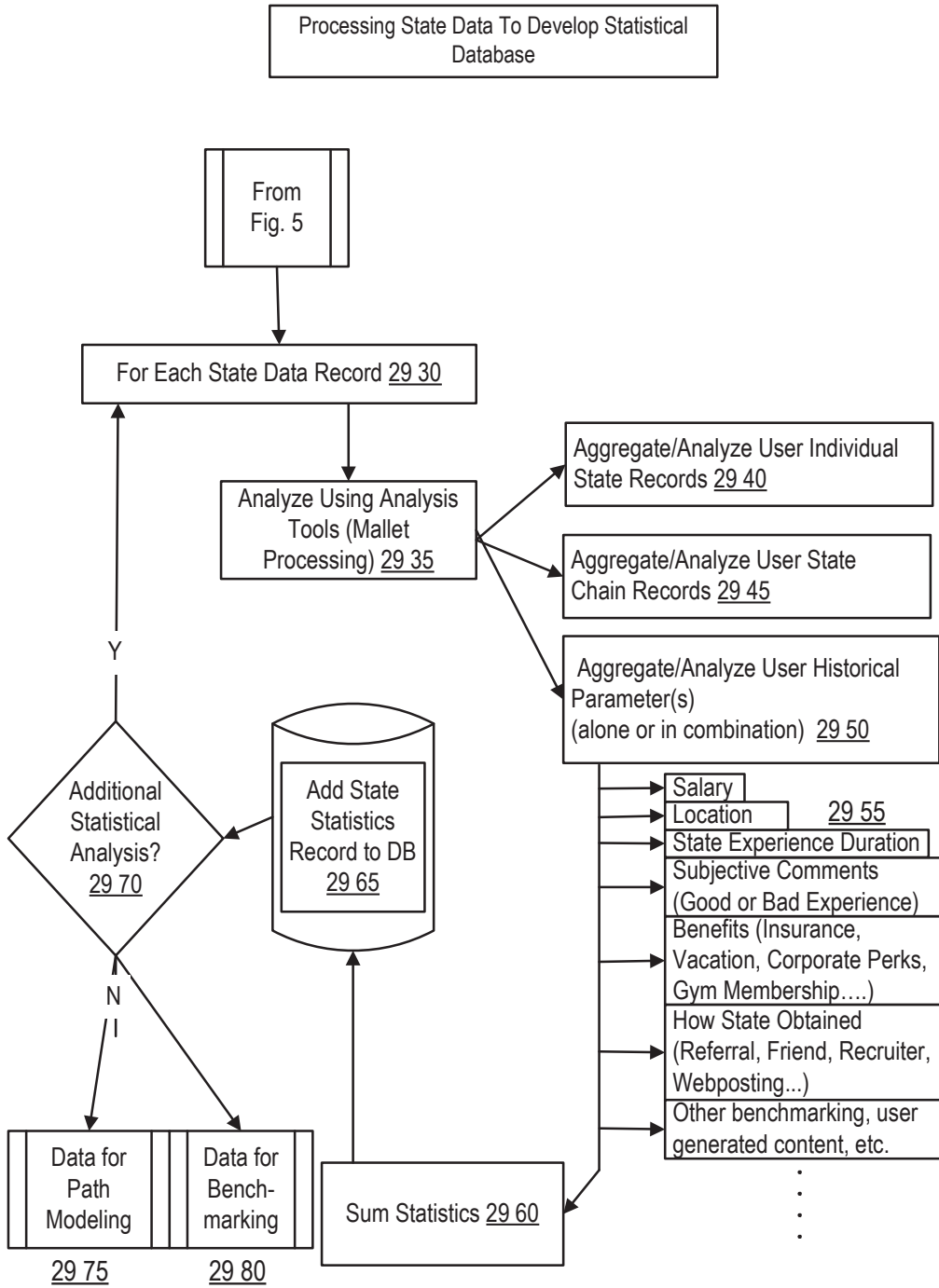


FIGURE 30

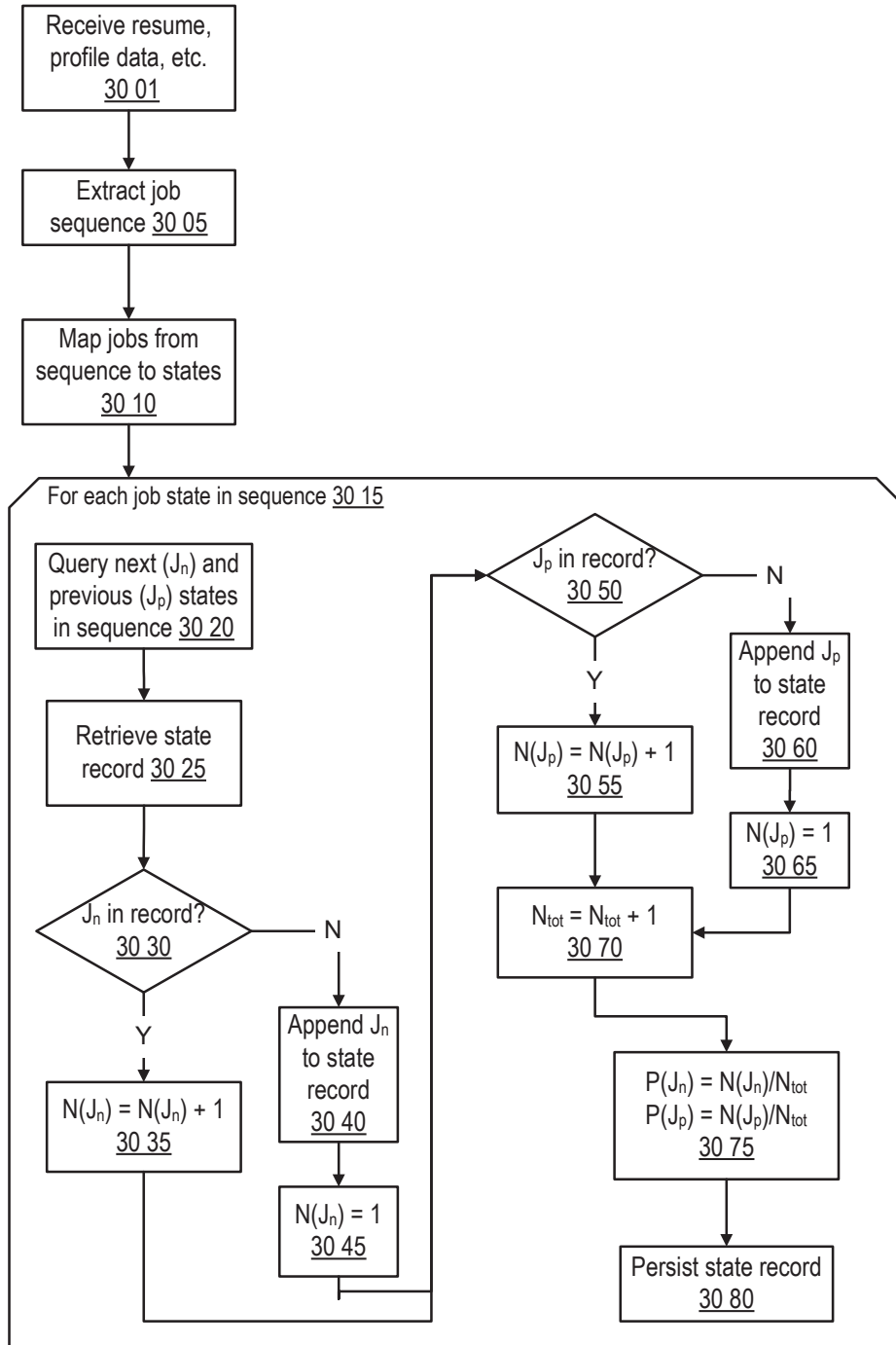


FIGURE 31

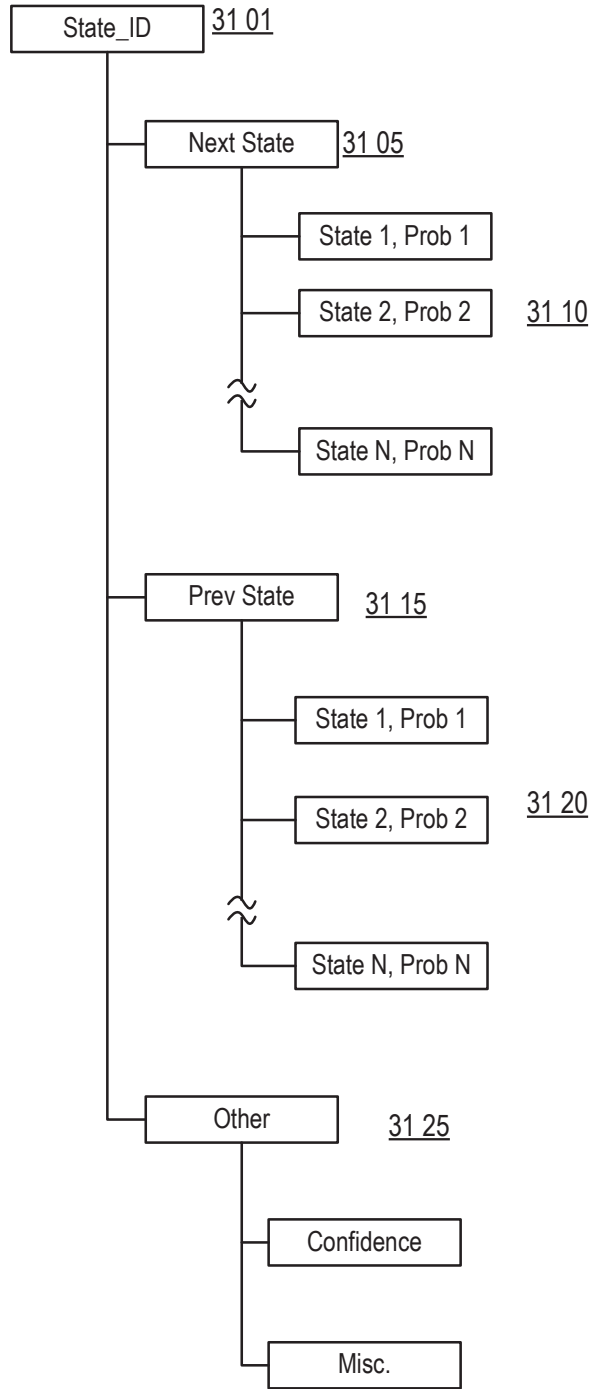


FIGURE 32

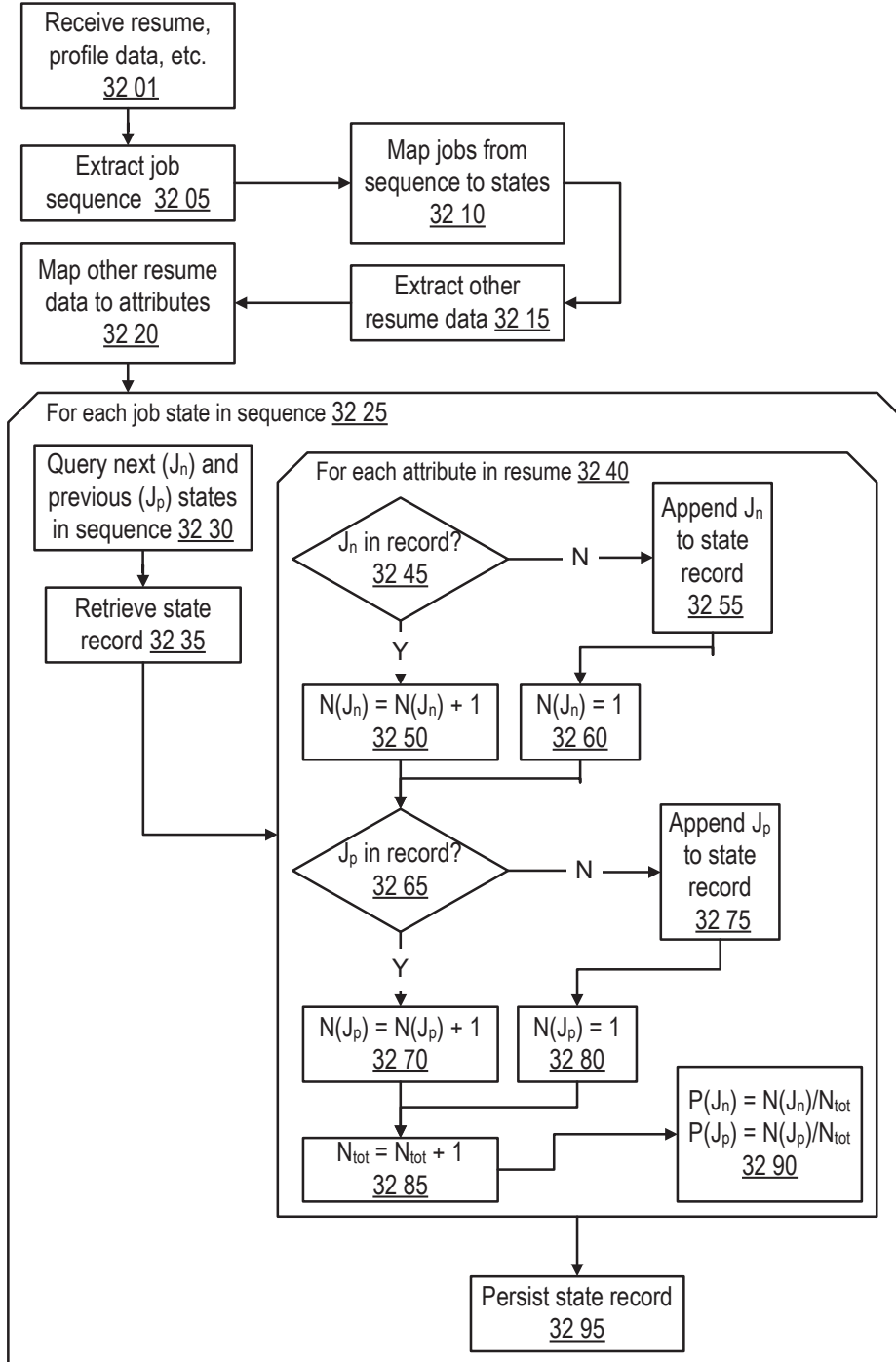


FIGURE 33

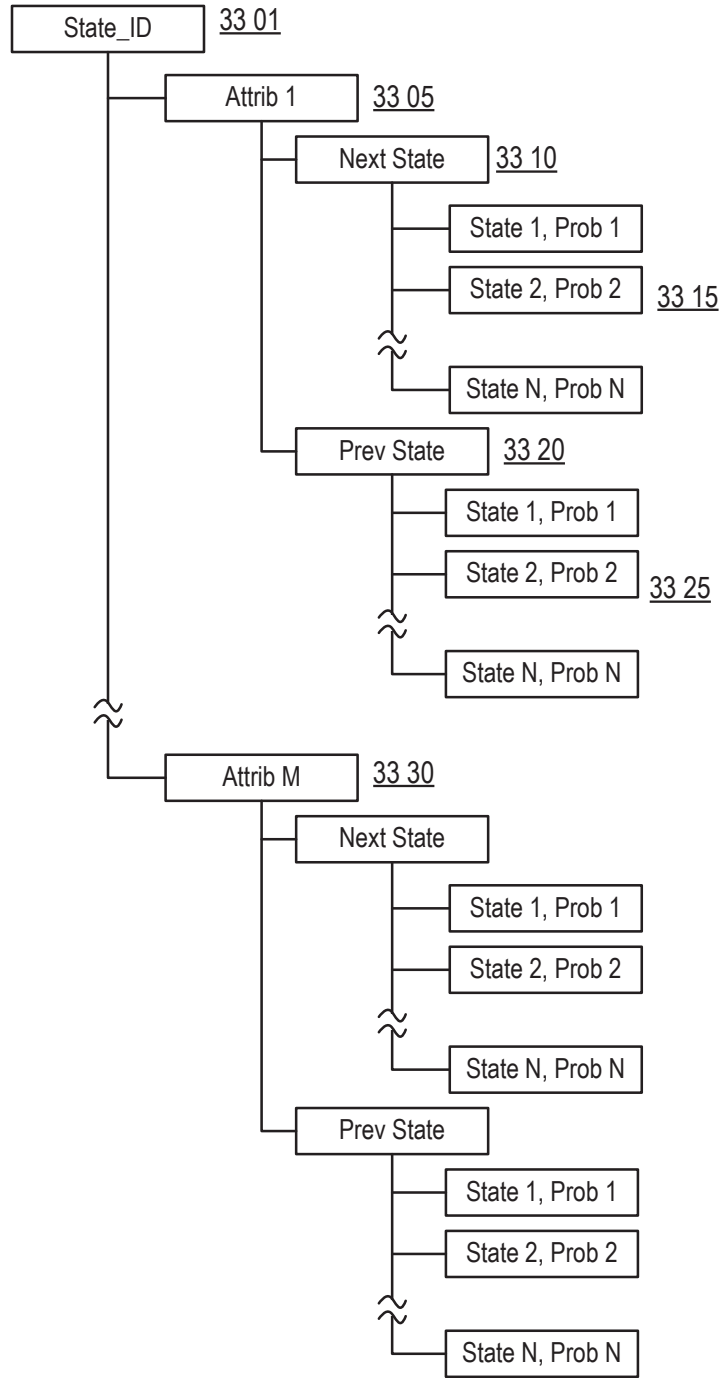


FIGURE 34

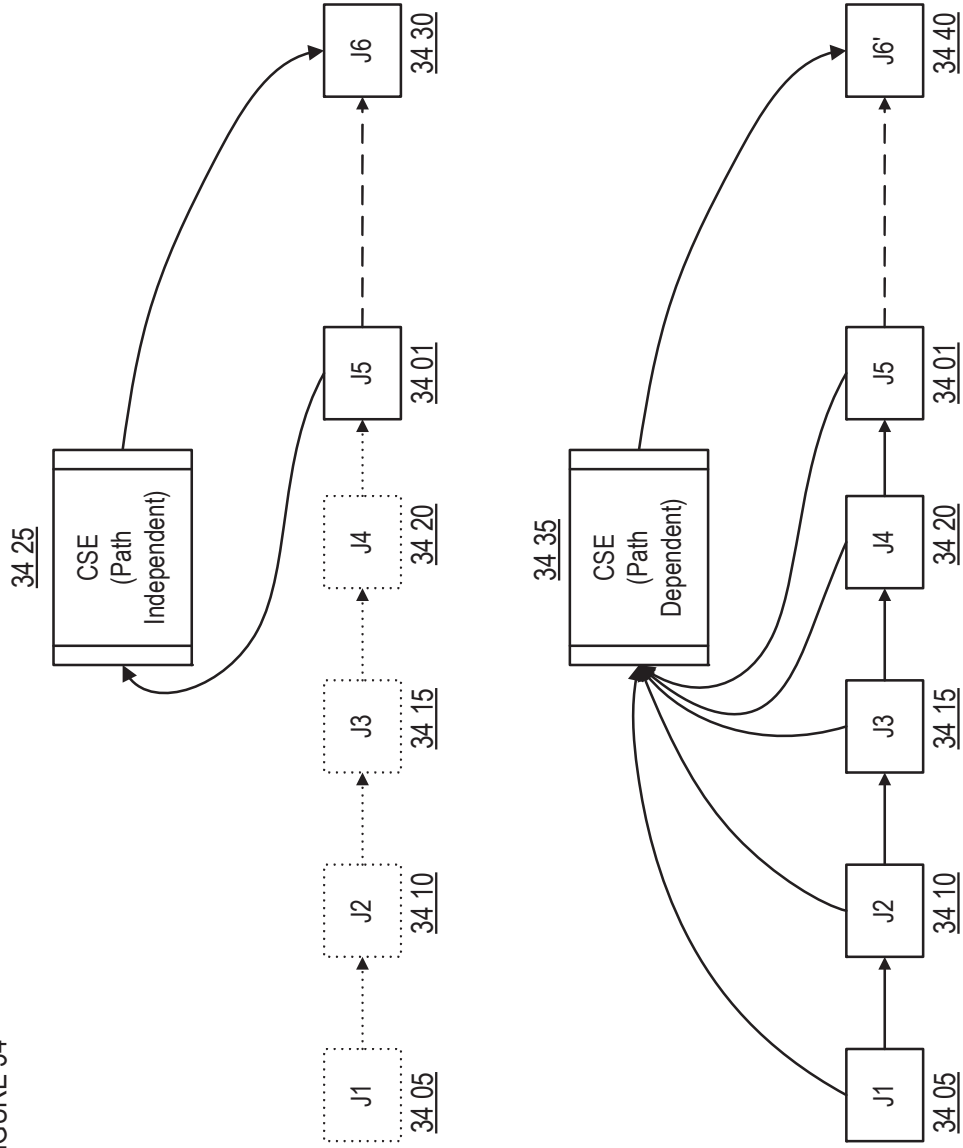


FIGURE 35

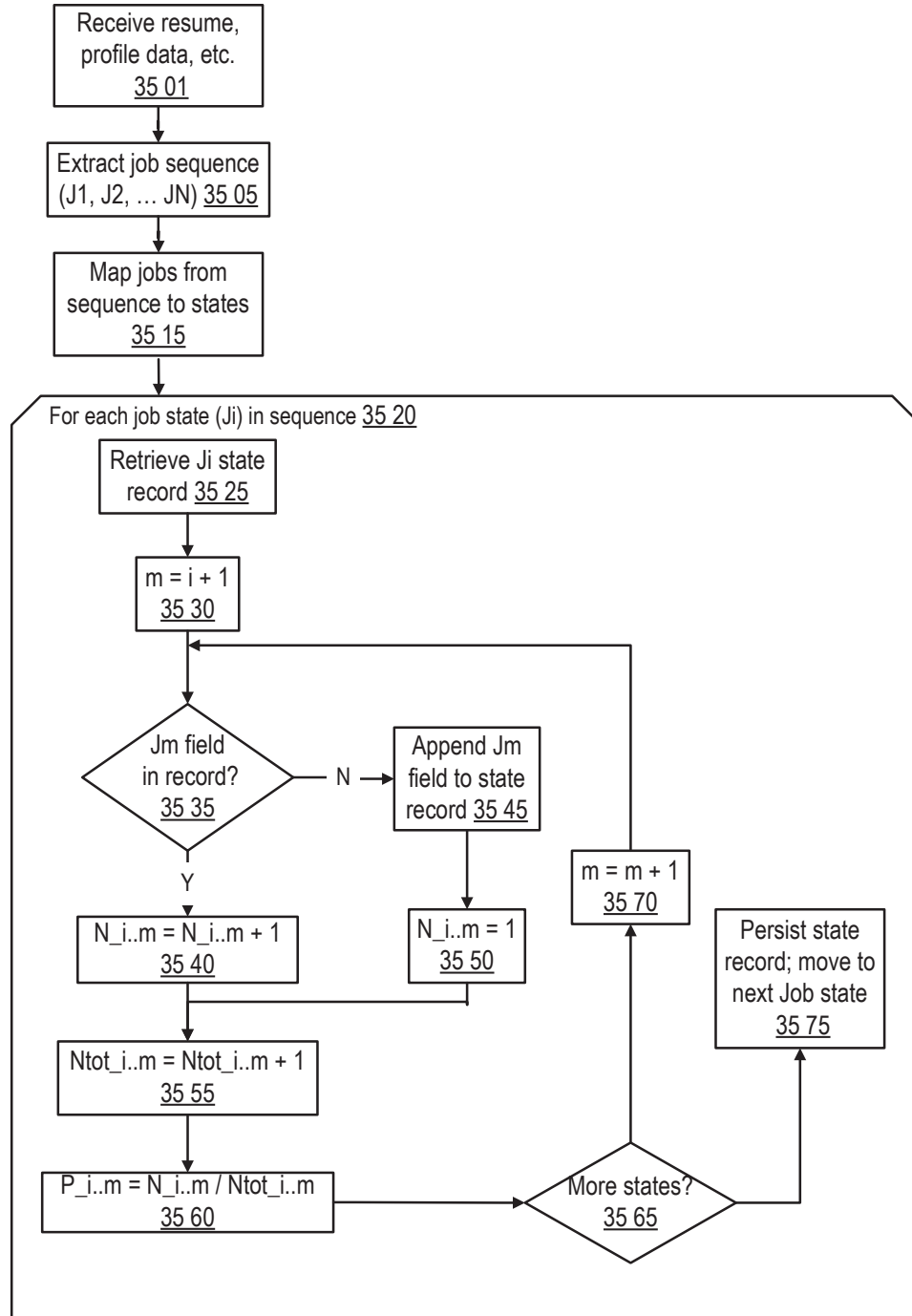


FIGURE 36

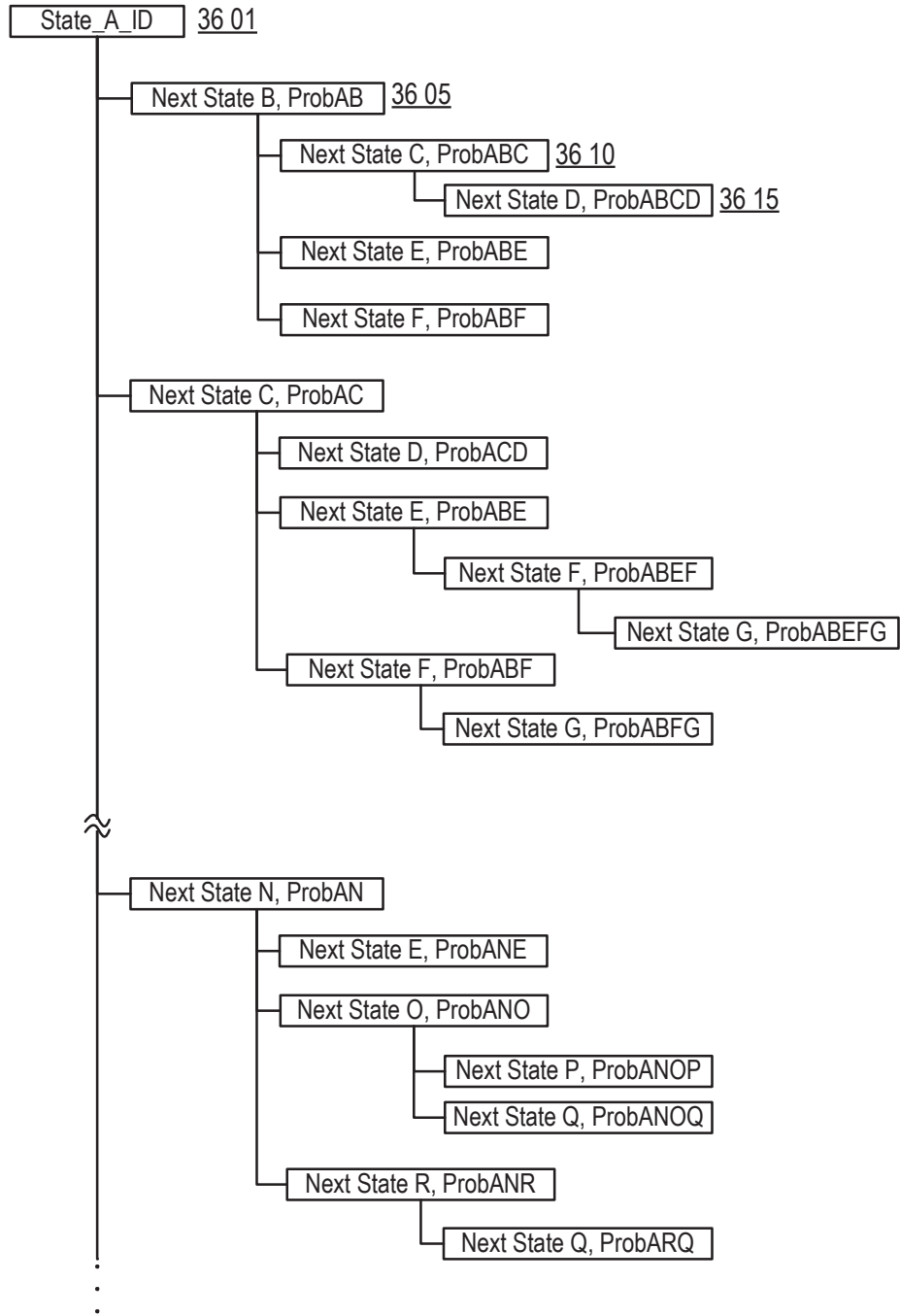


FIGURE 37A

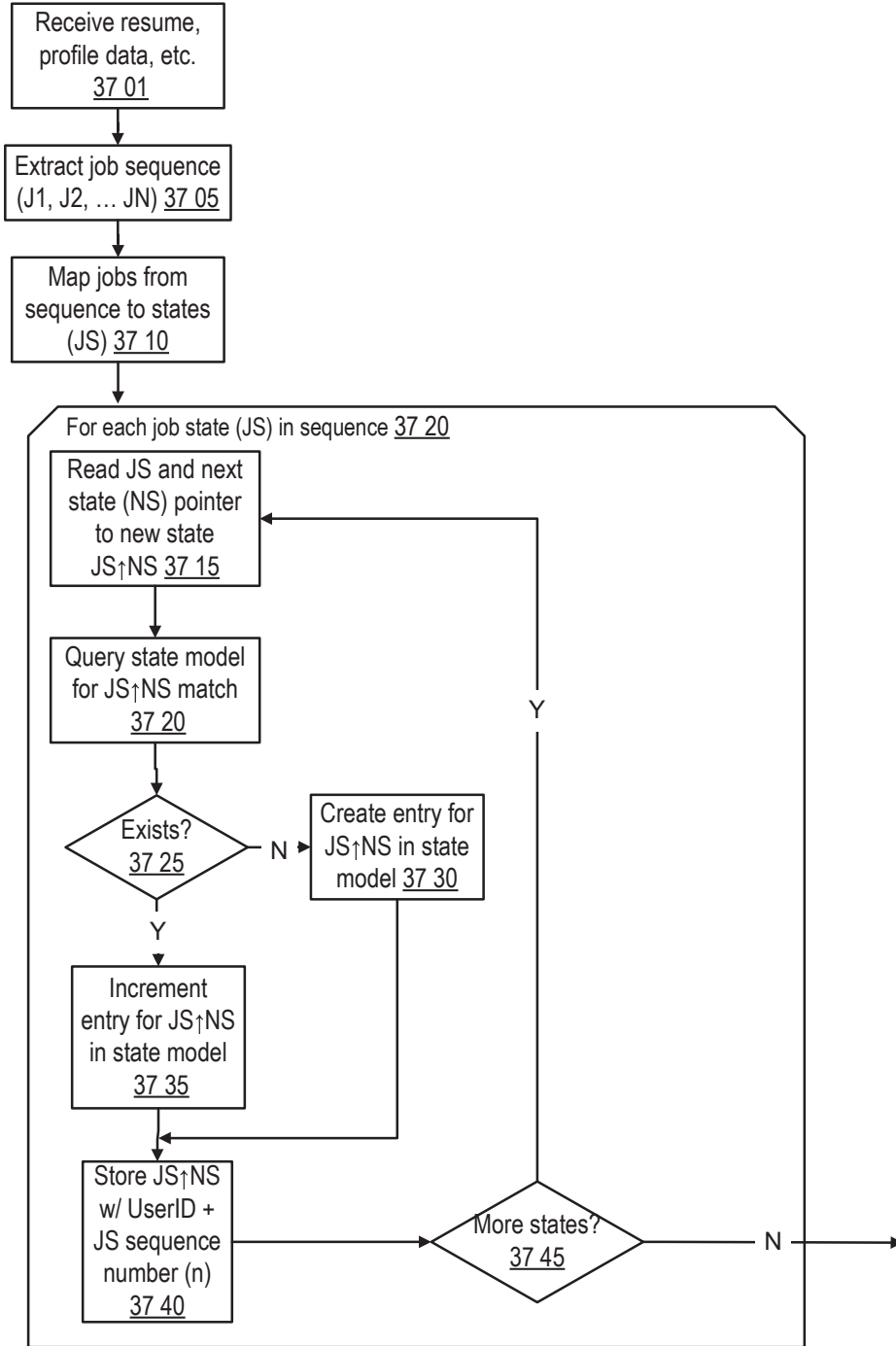
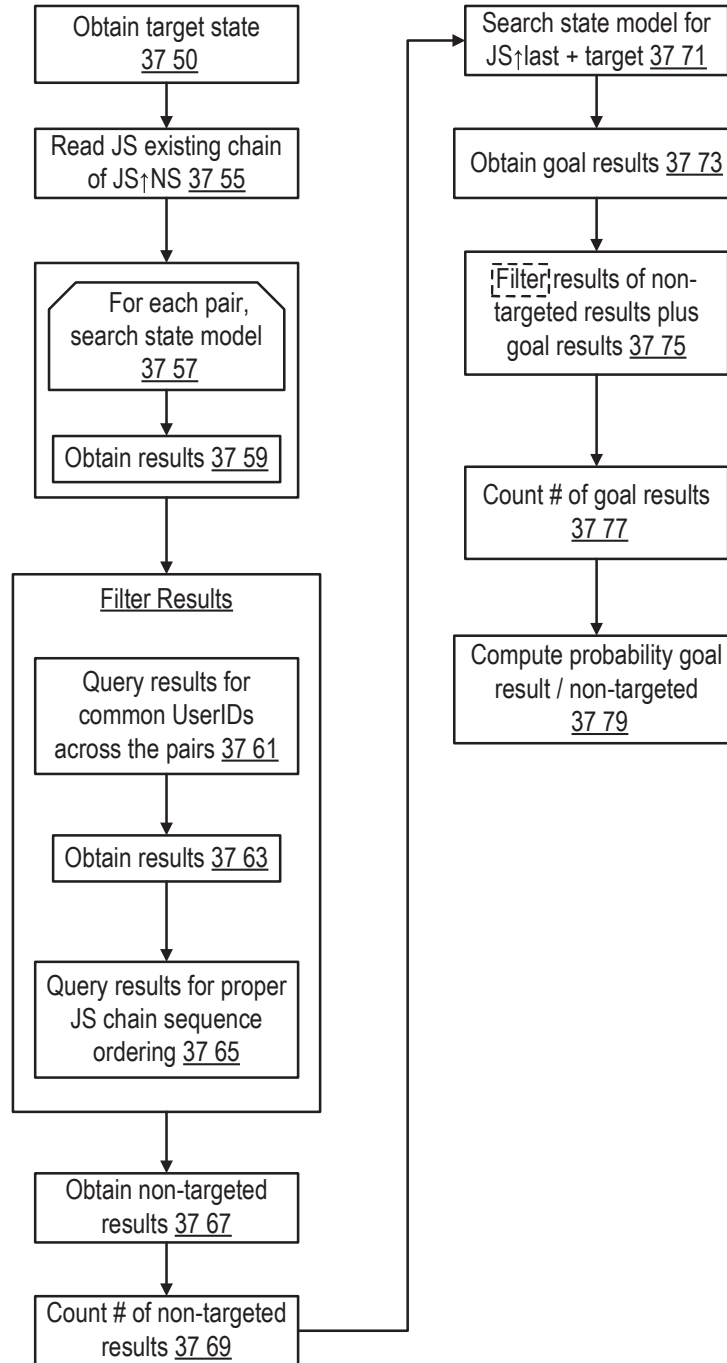
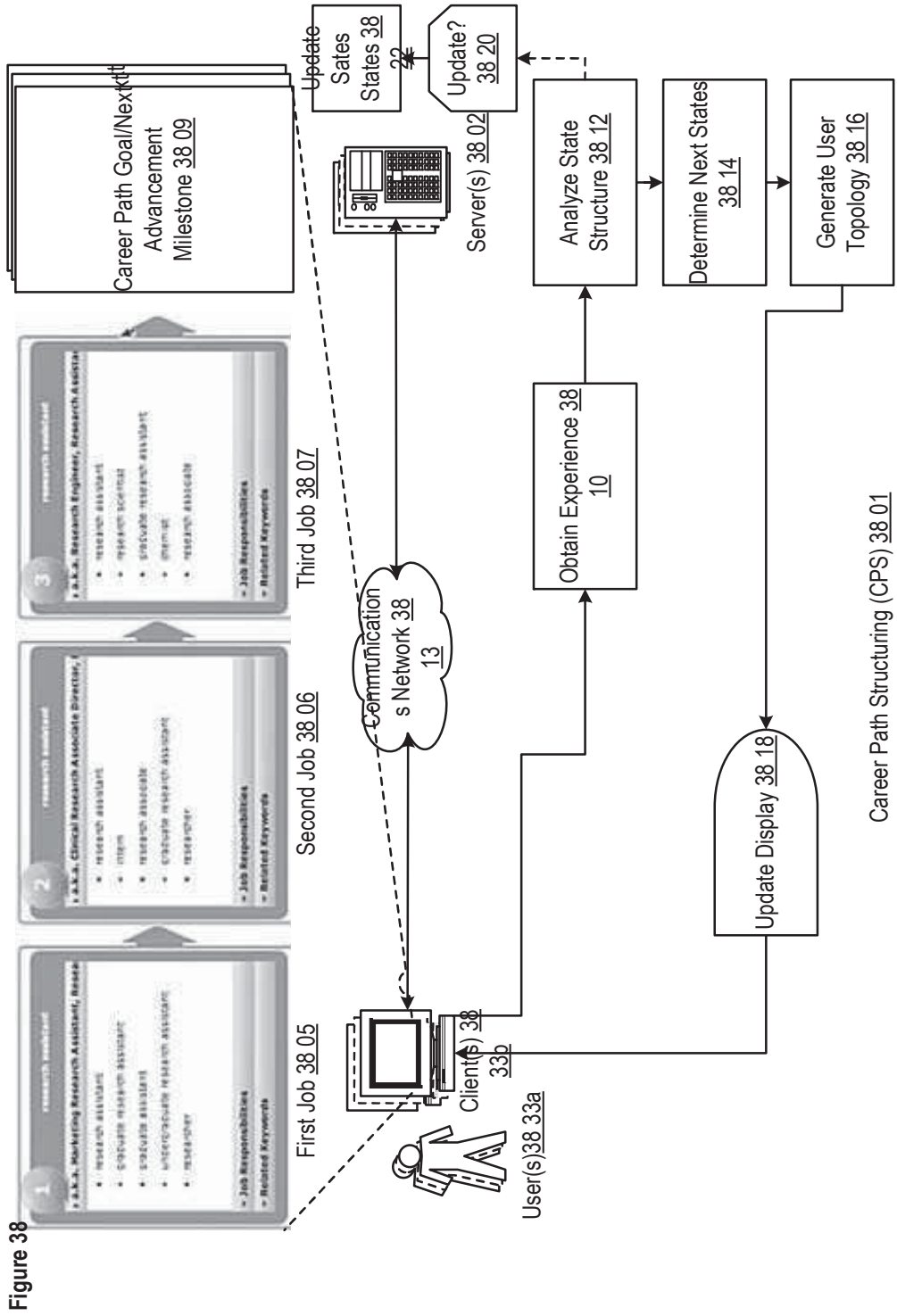


FIGURE 37B





Career Path structuring (CPS) Overall Flow

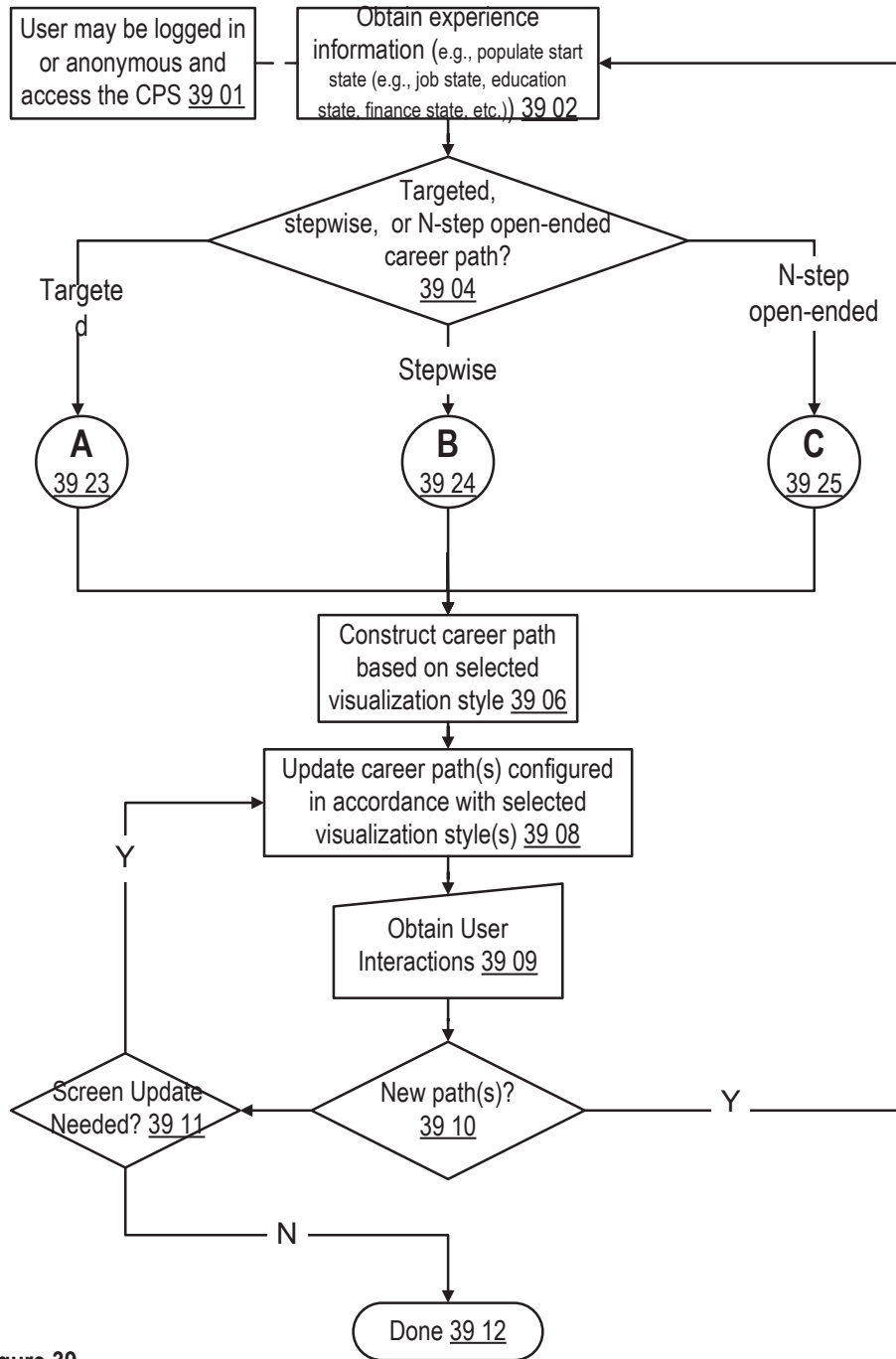
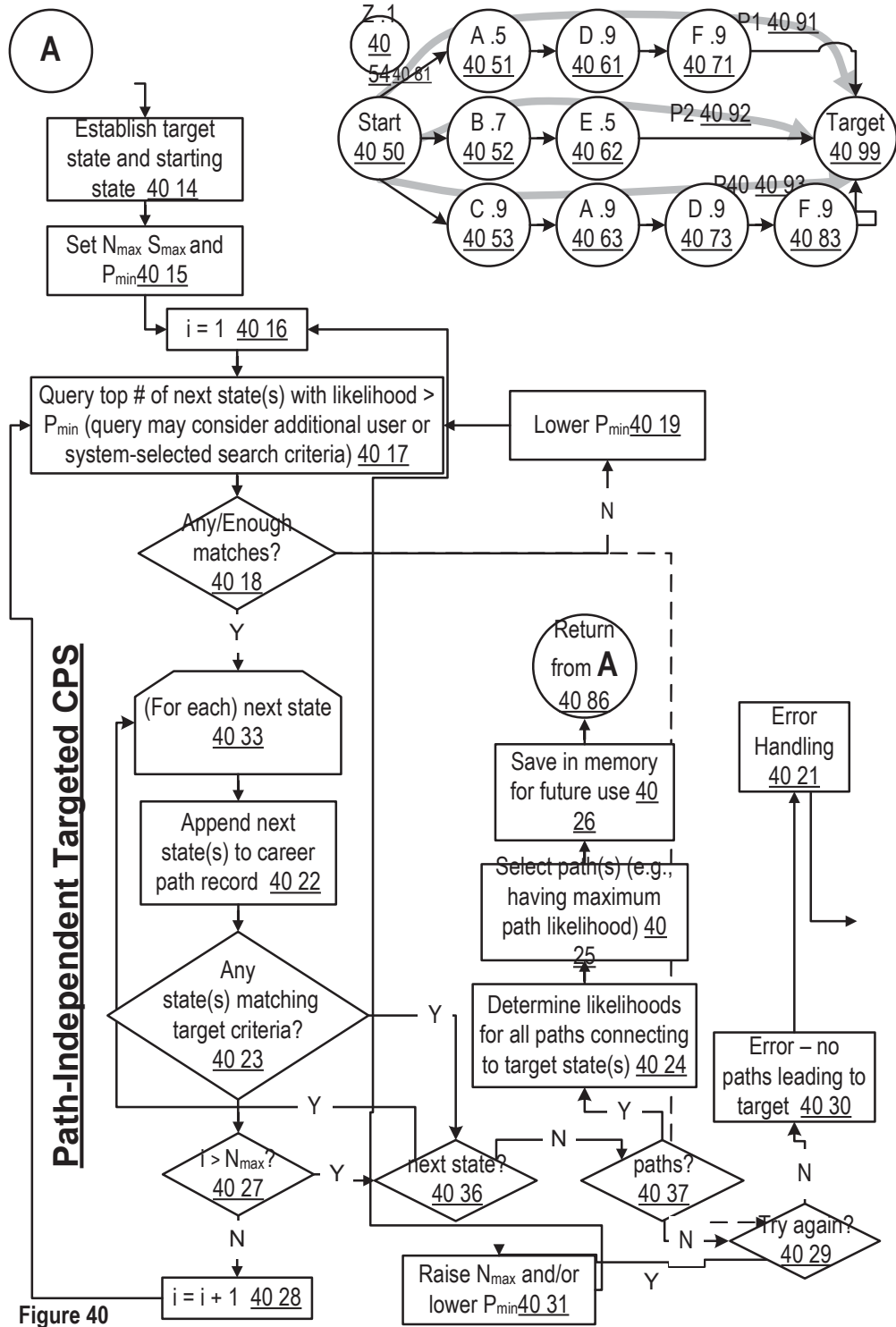


Figure 39

62/167



B

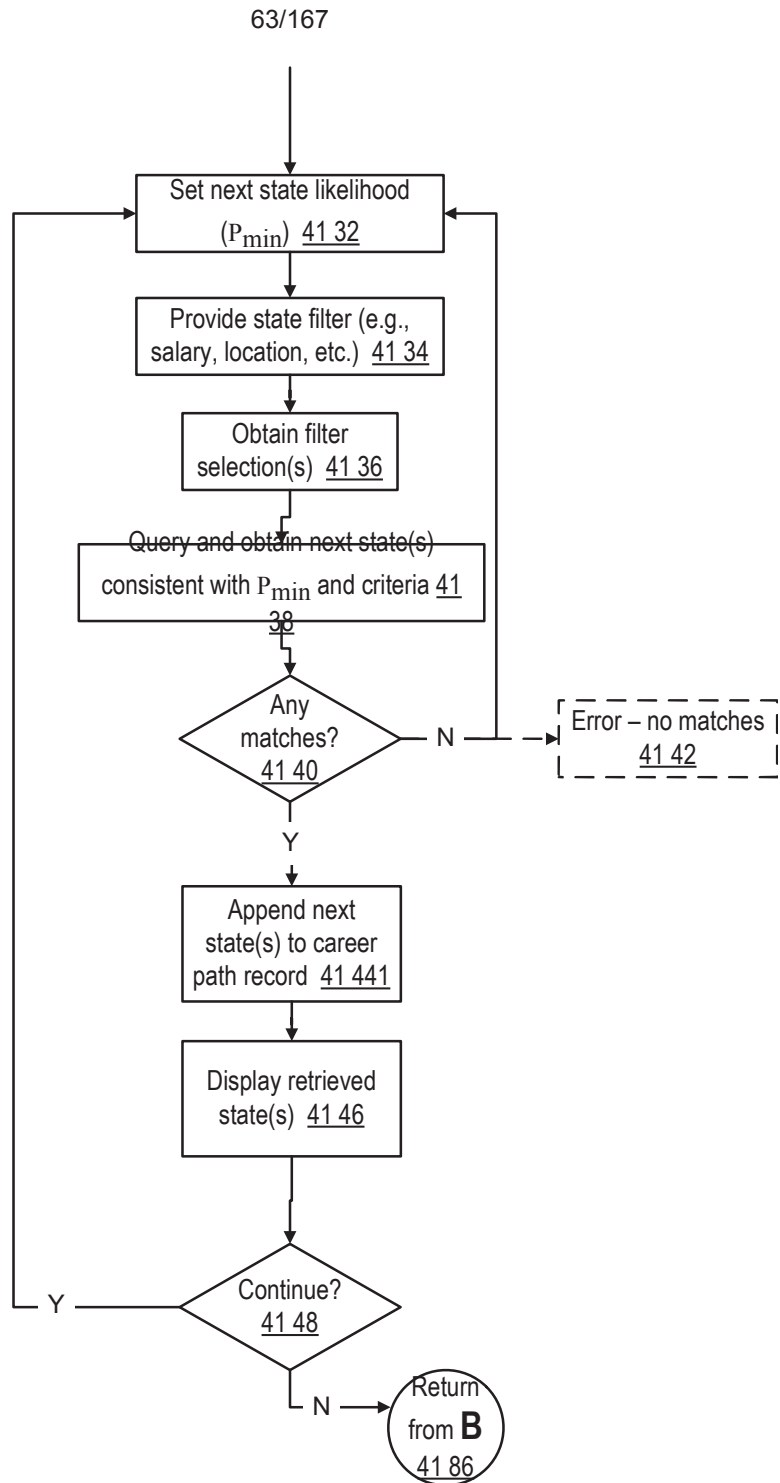


Figure 41

Path-Independent Stepwise CPS

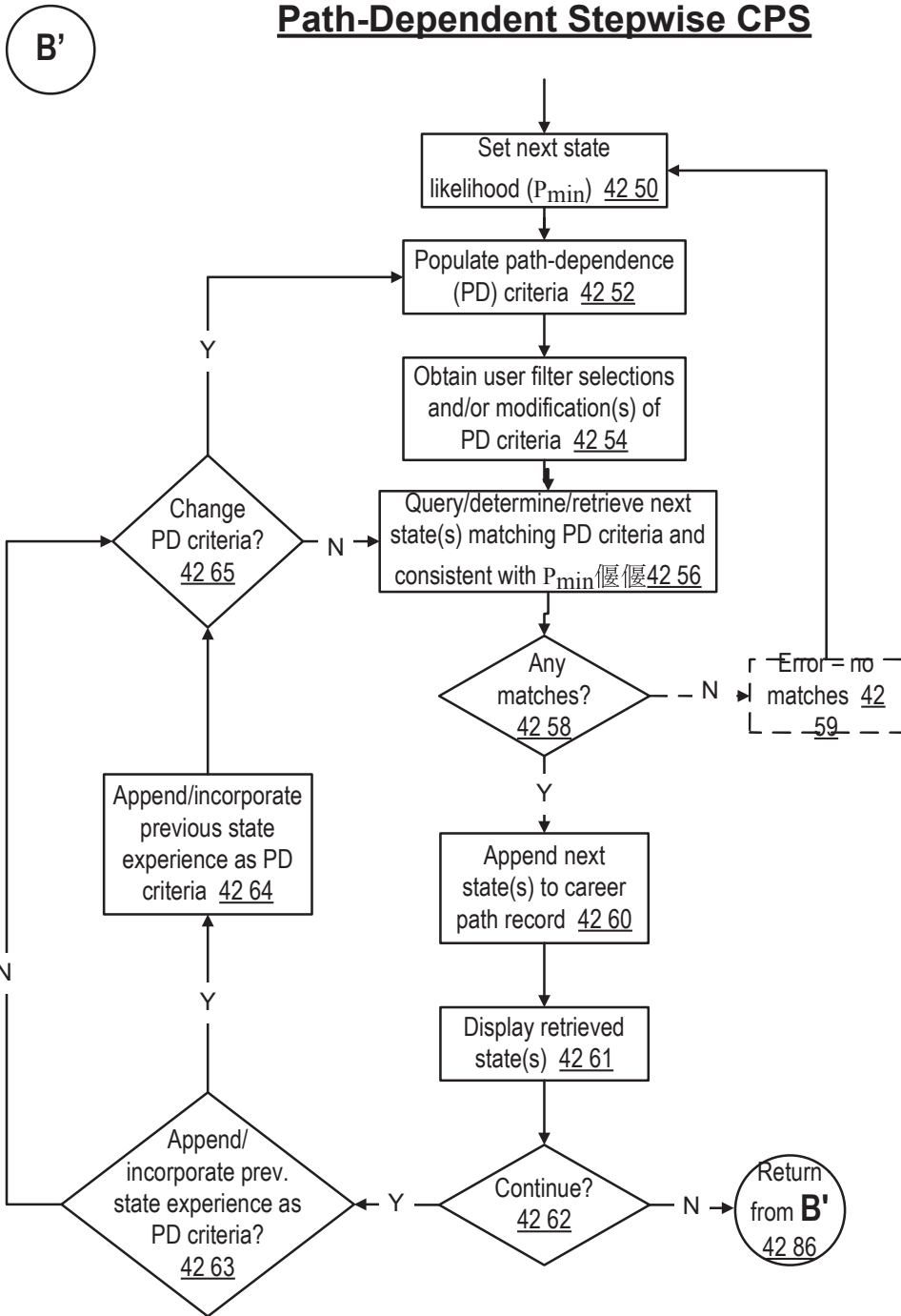


Figure 42

Path-Independent N-Step Open-Ended CPS

C

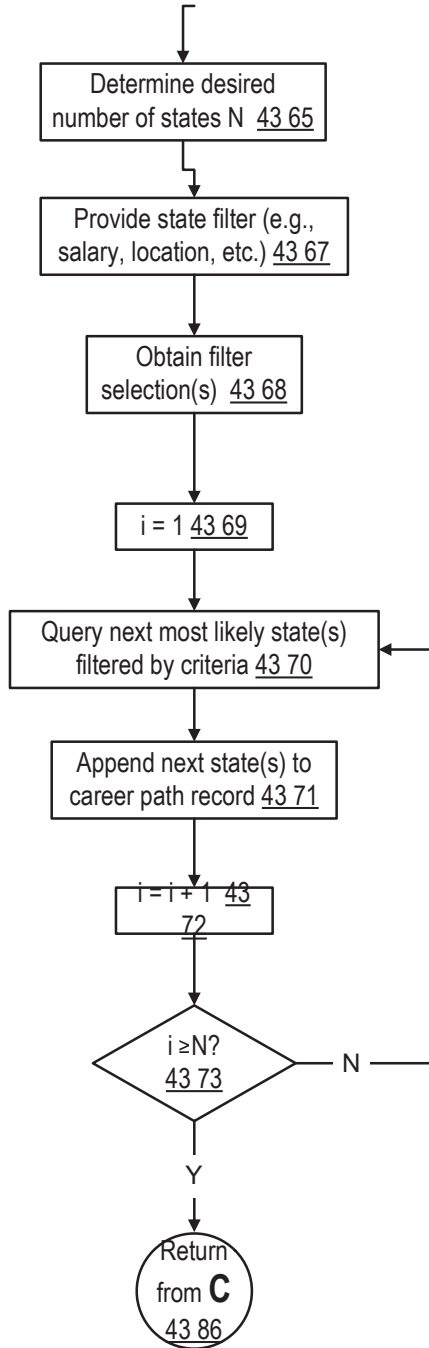


Figure 43

Path-Dependent N-Step Open-Ended CPS

C'

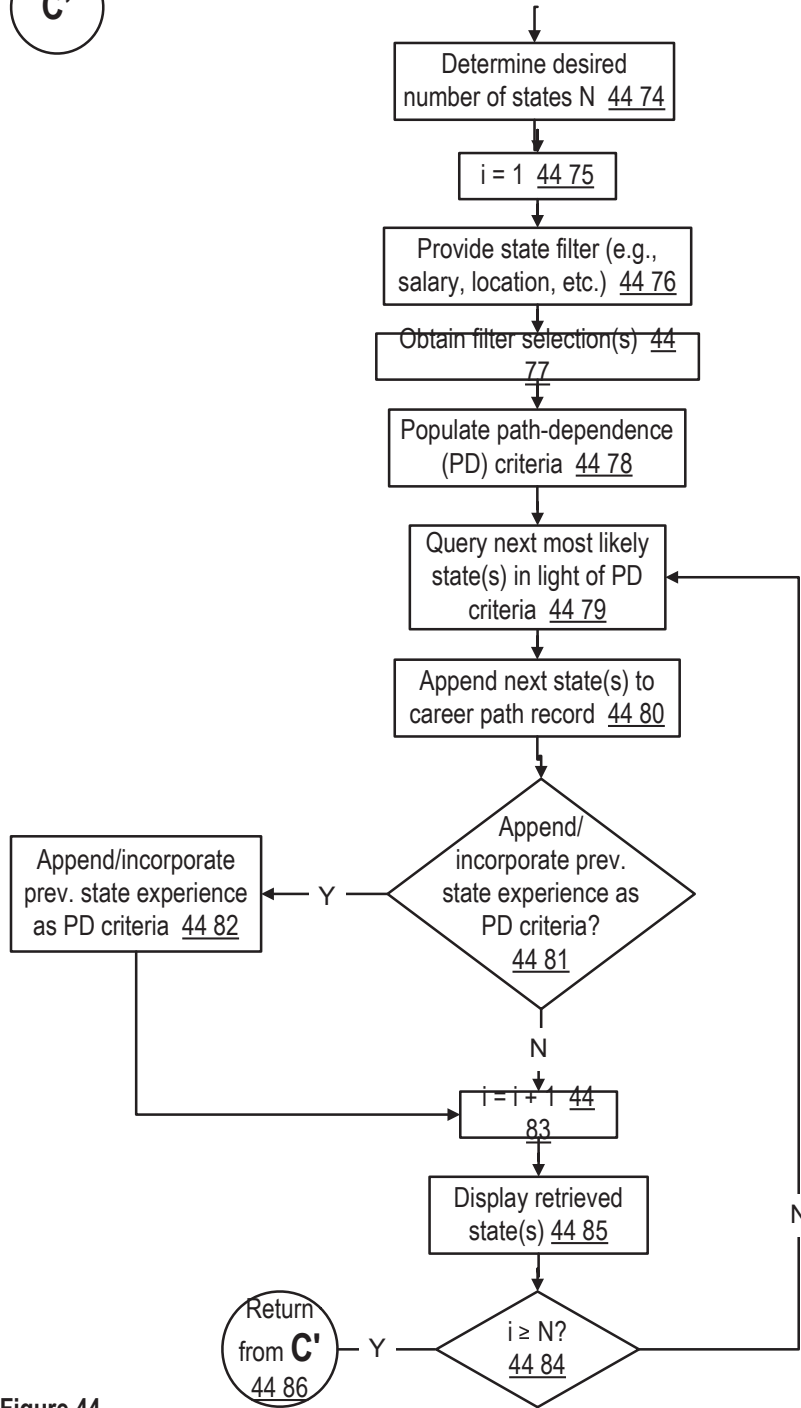


Figure 44

Gap Analysis

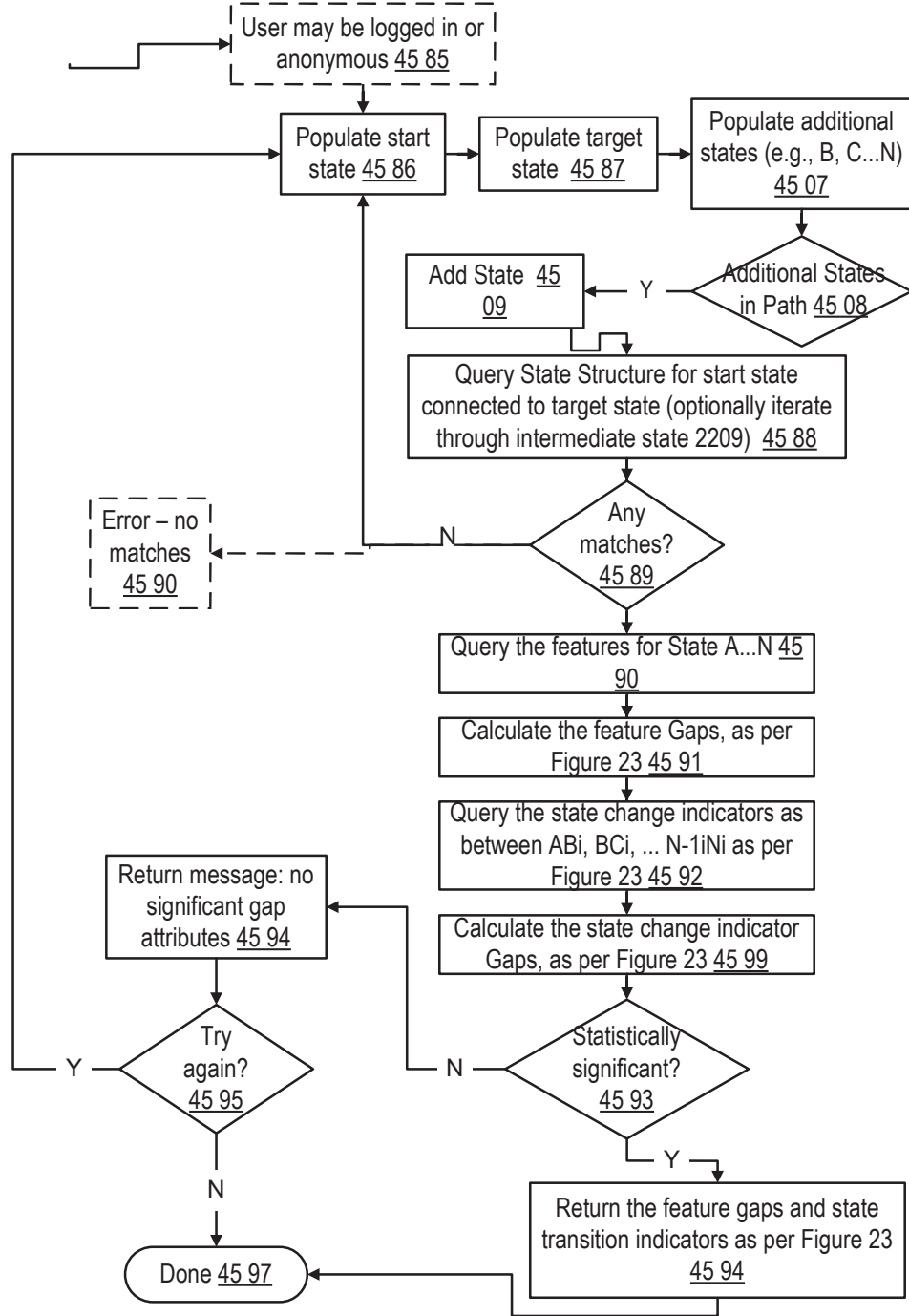
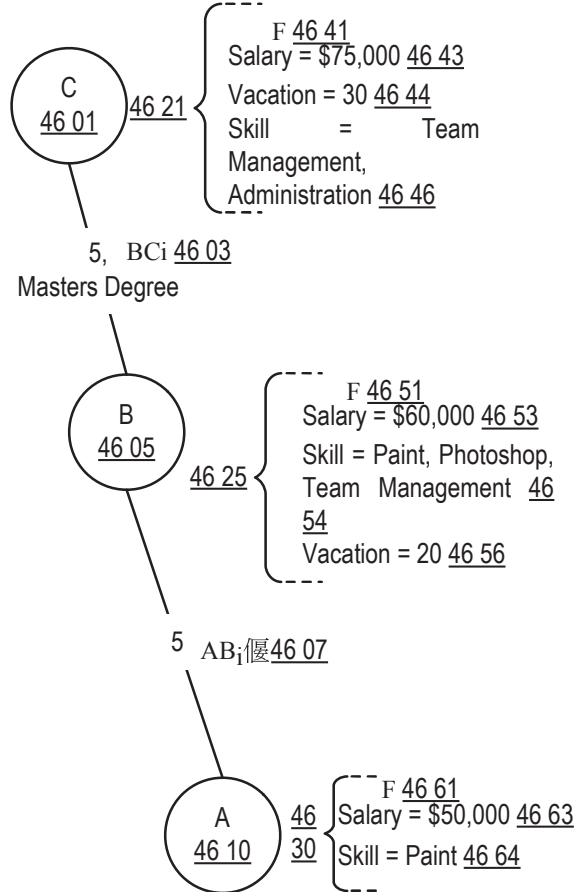


Figure 45



$$G_i(A \rightarrow B) = B_F - A_F + AB_i \quad 46 \ 99$$

$$G_i(B \rightarrow C) = C_F - B_F + BC_i \quad 46 \ 98$$

$$G_i(A \rightarrow C) = C_F - B_F - A_F + AB_i + BC_i \quad 46 \ 97$$

Figure 46

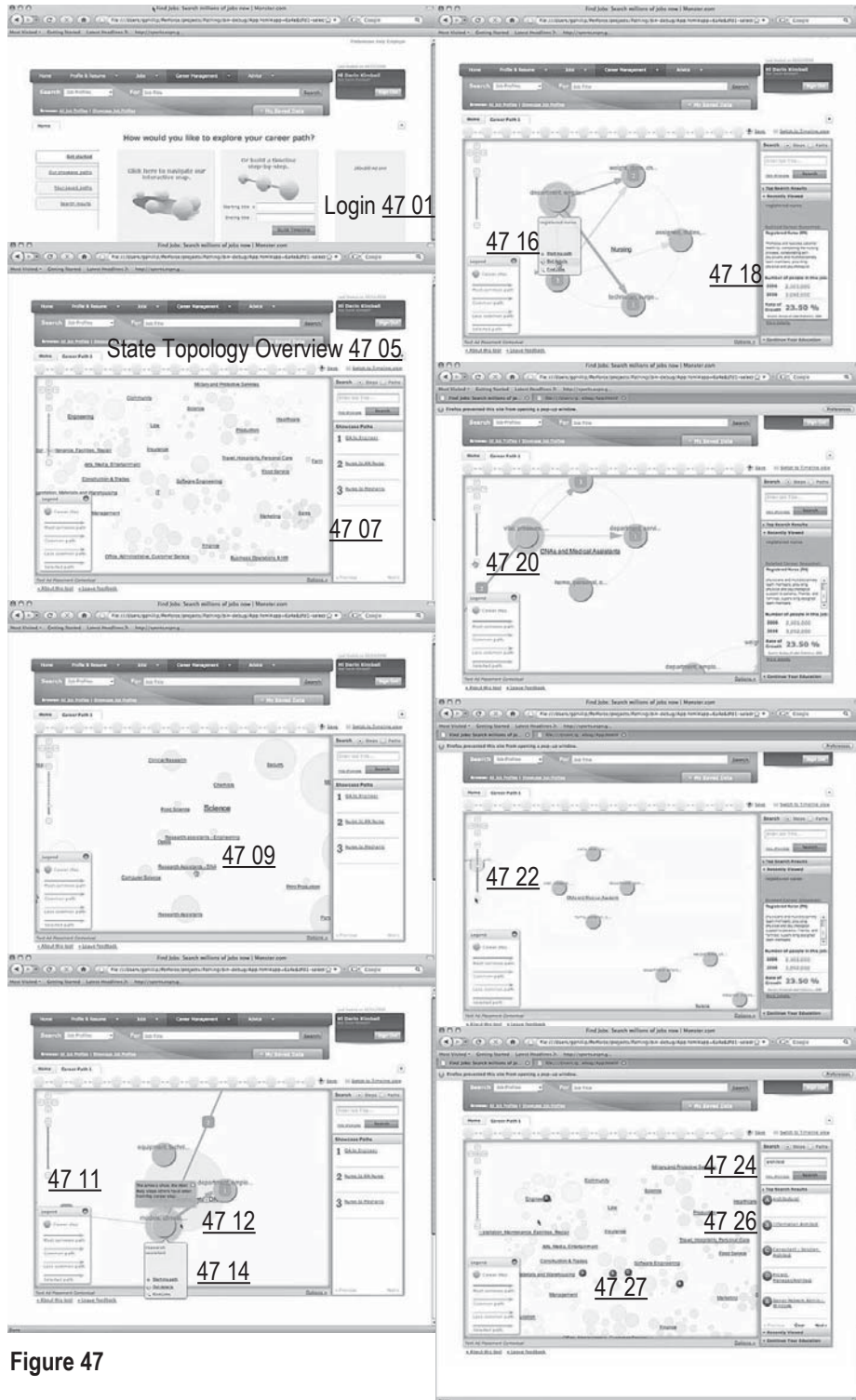
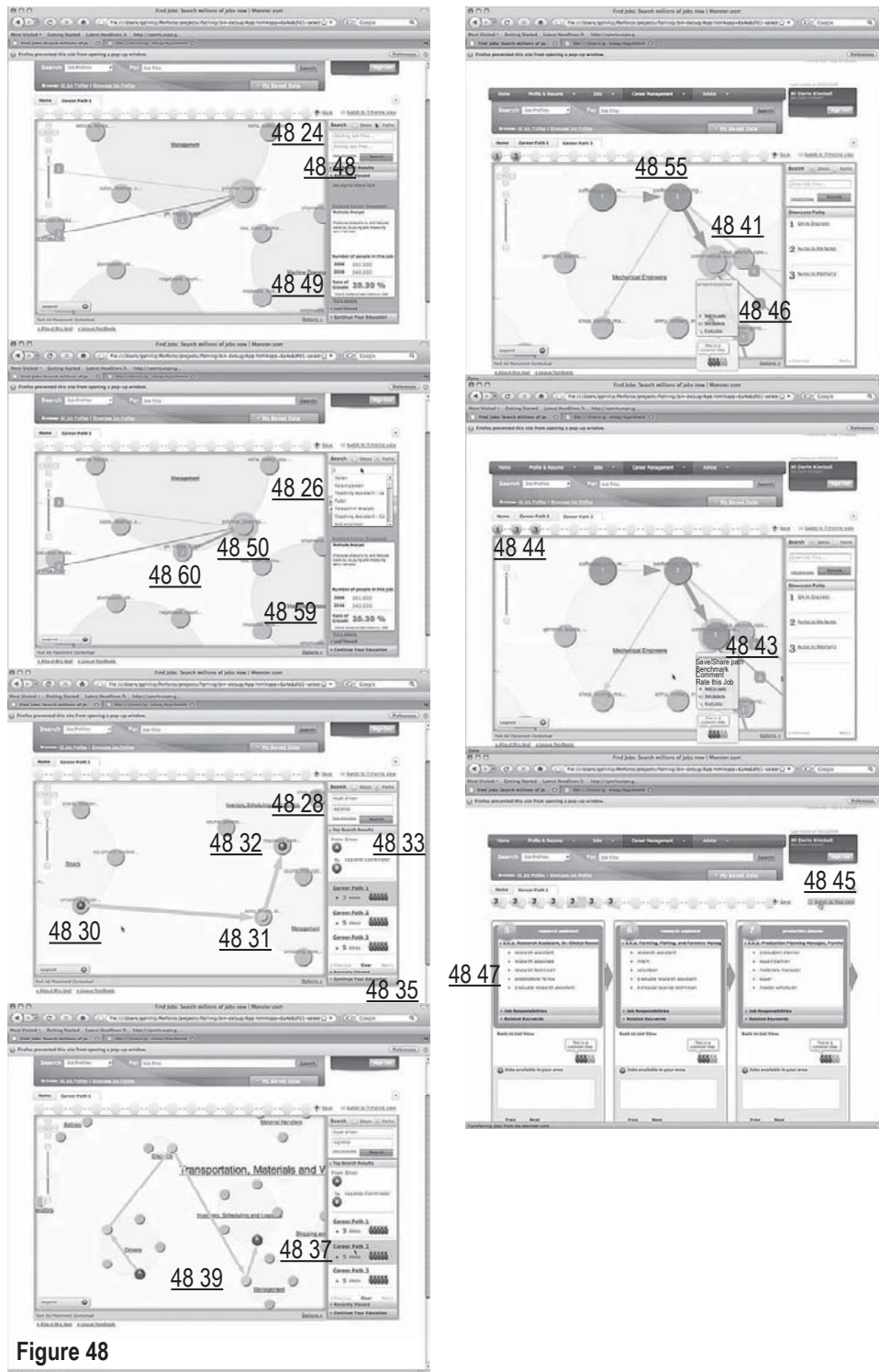
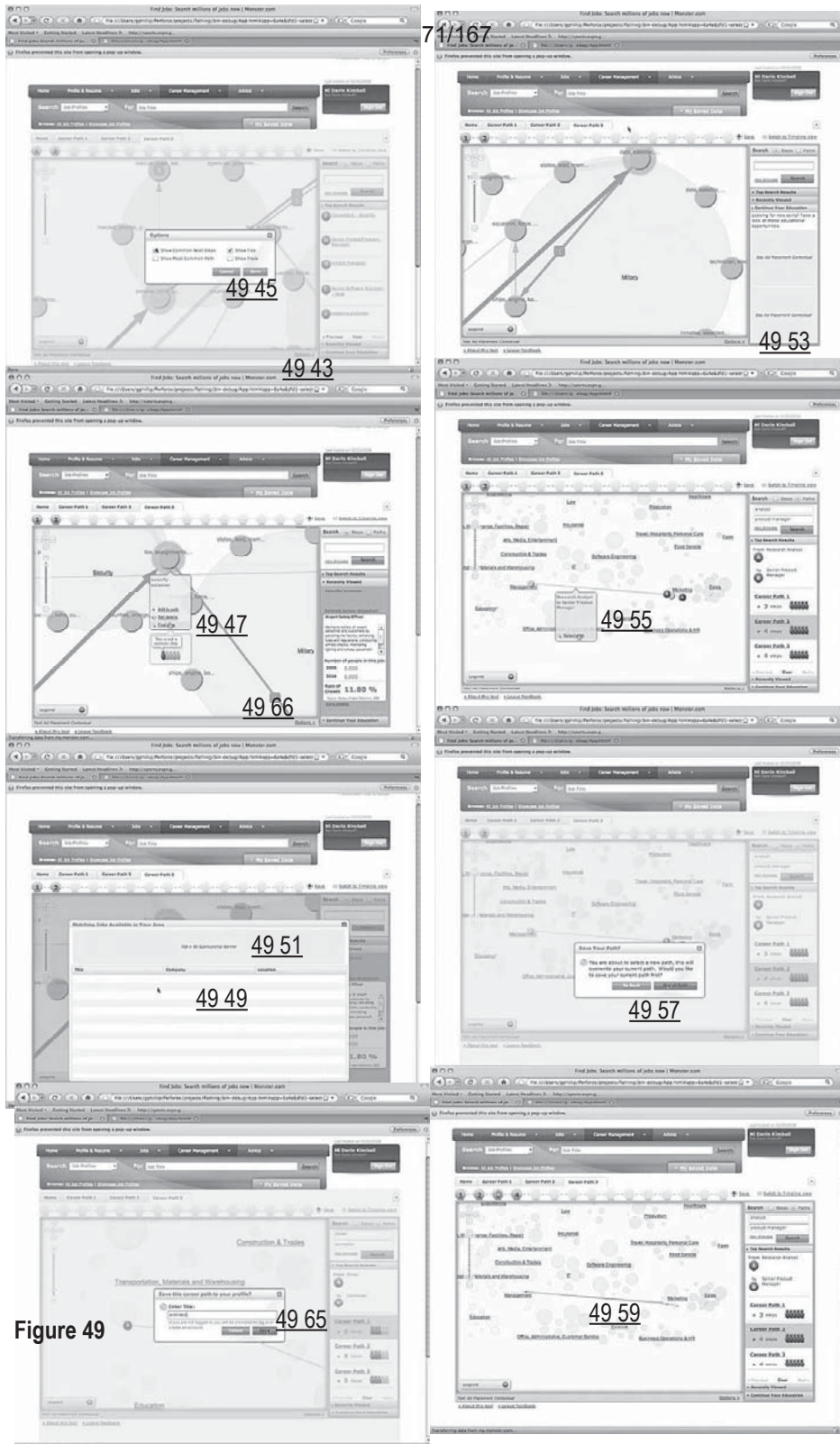


Figure 47





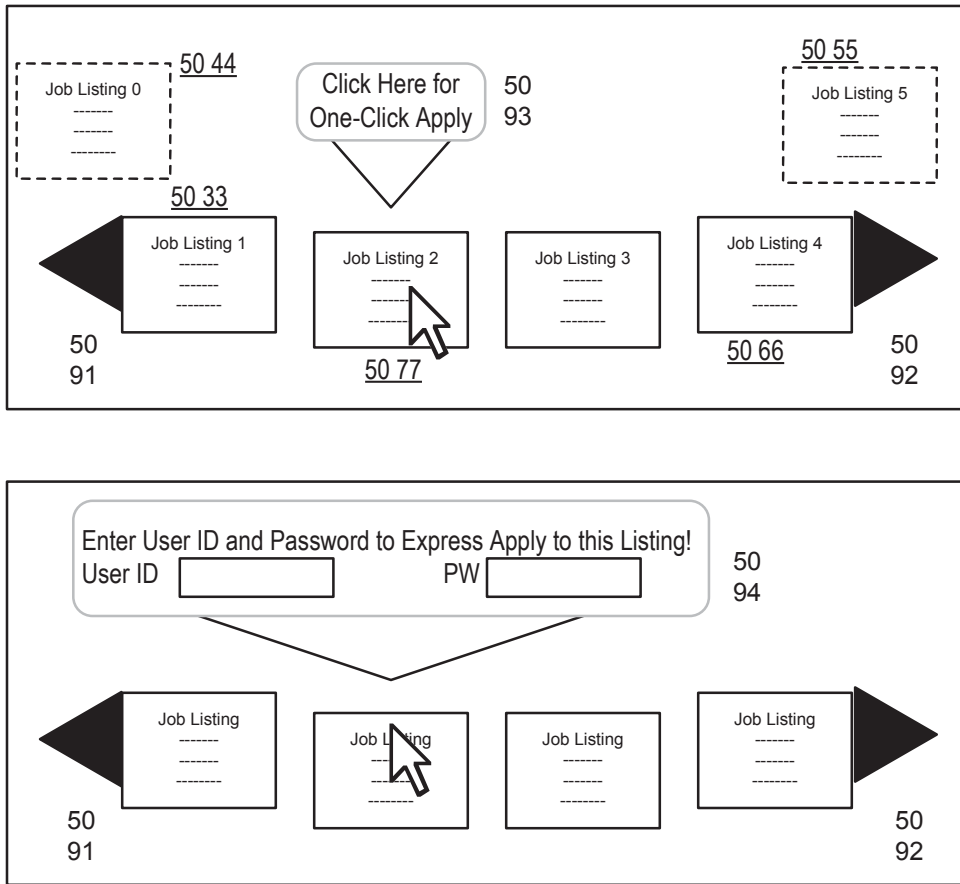


Figure 50

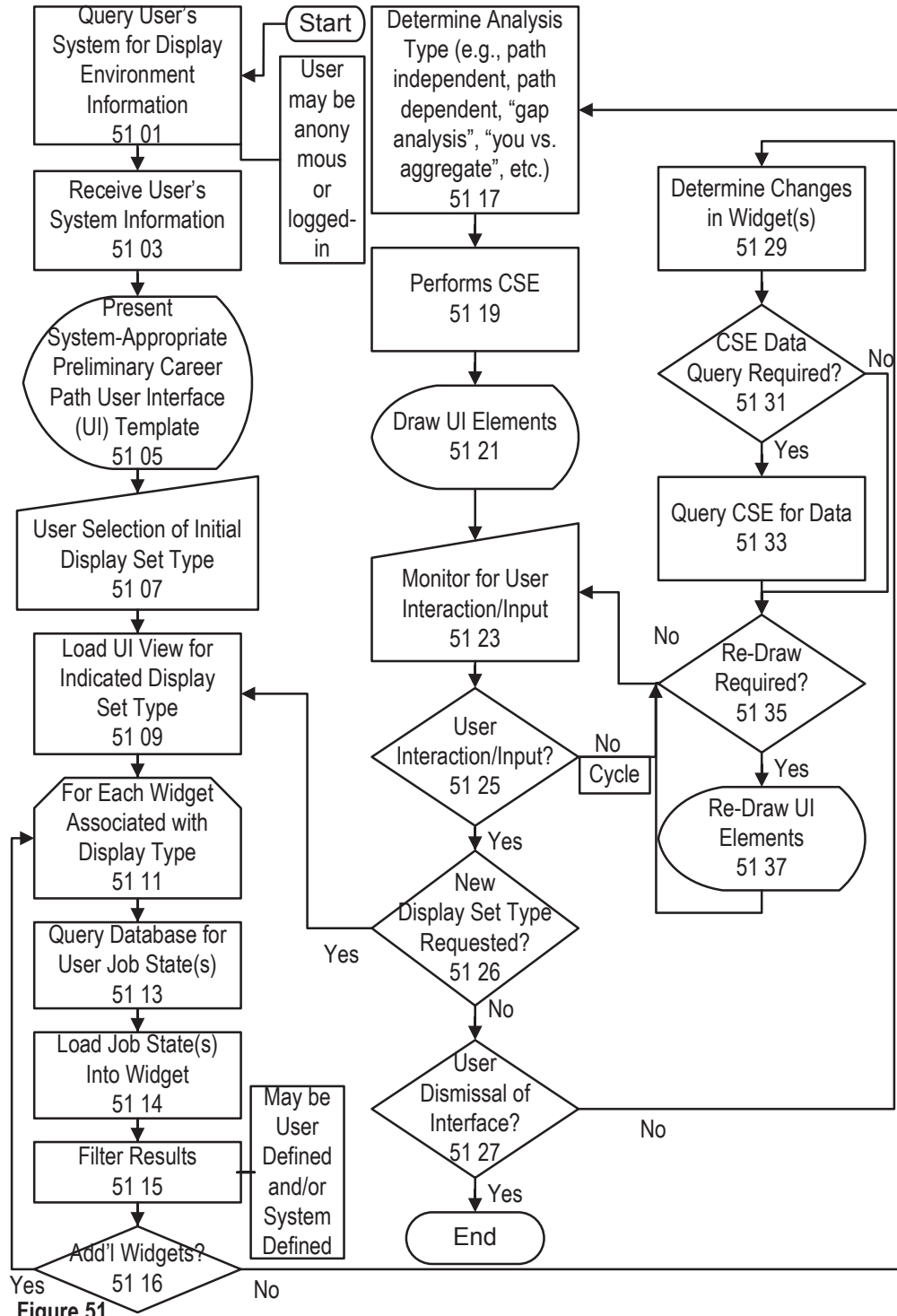


Figure 51

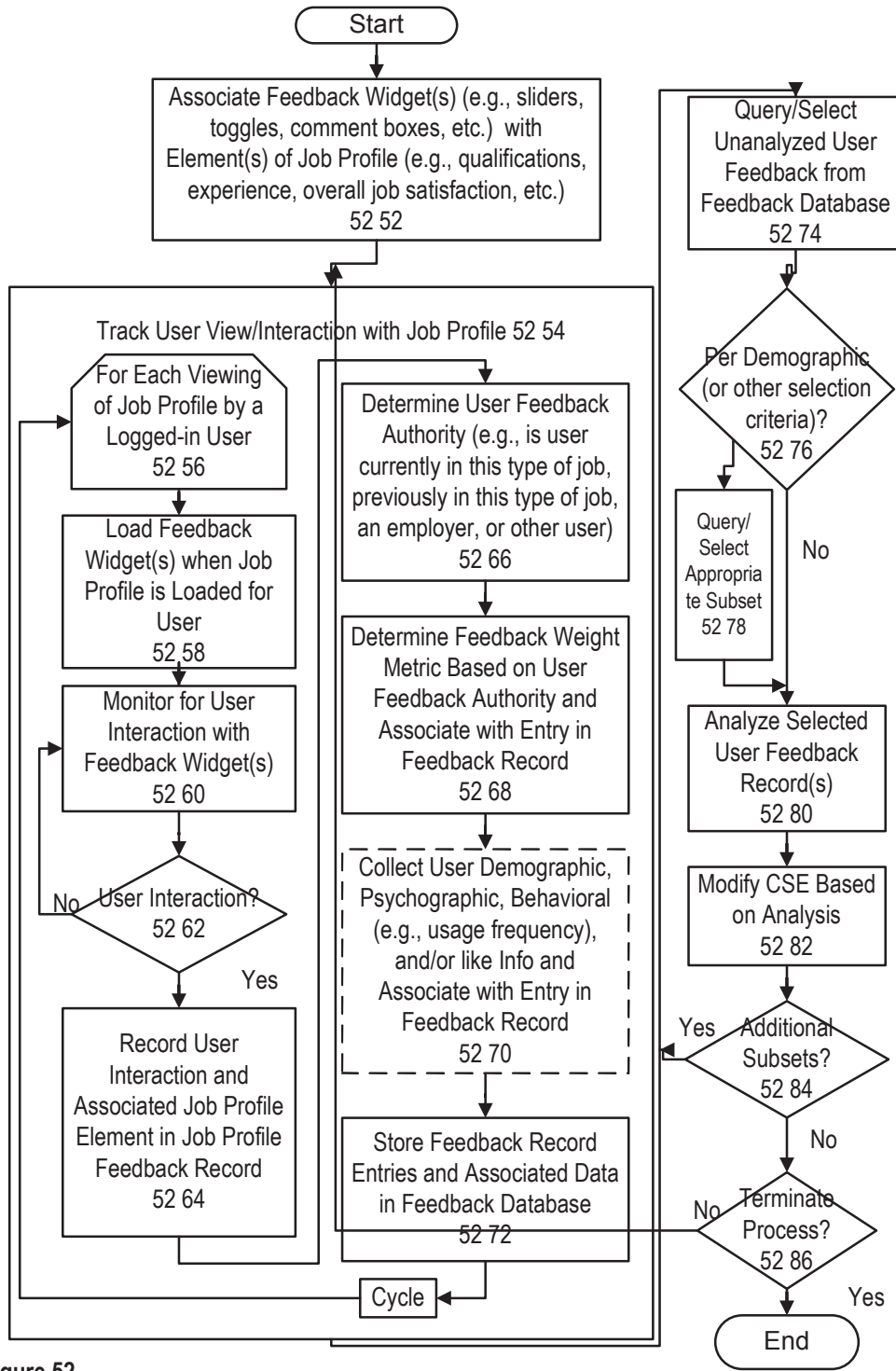


Figure 52

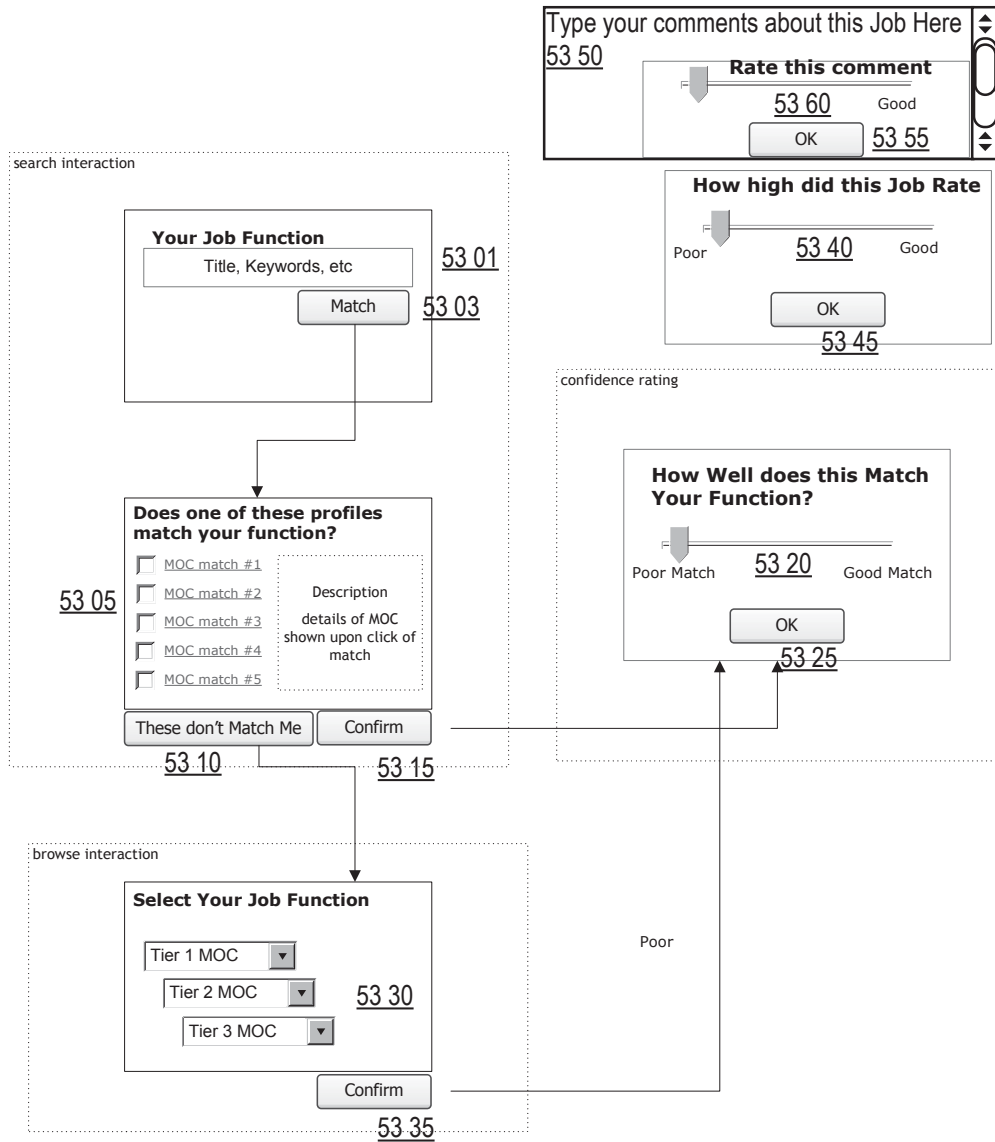


Figure 53

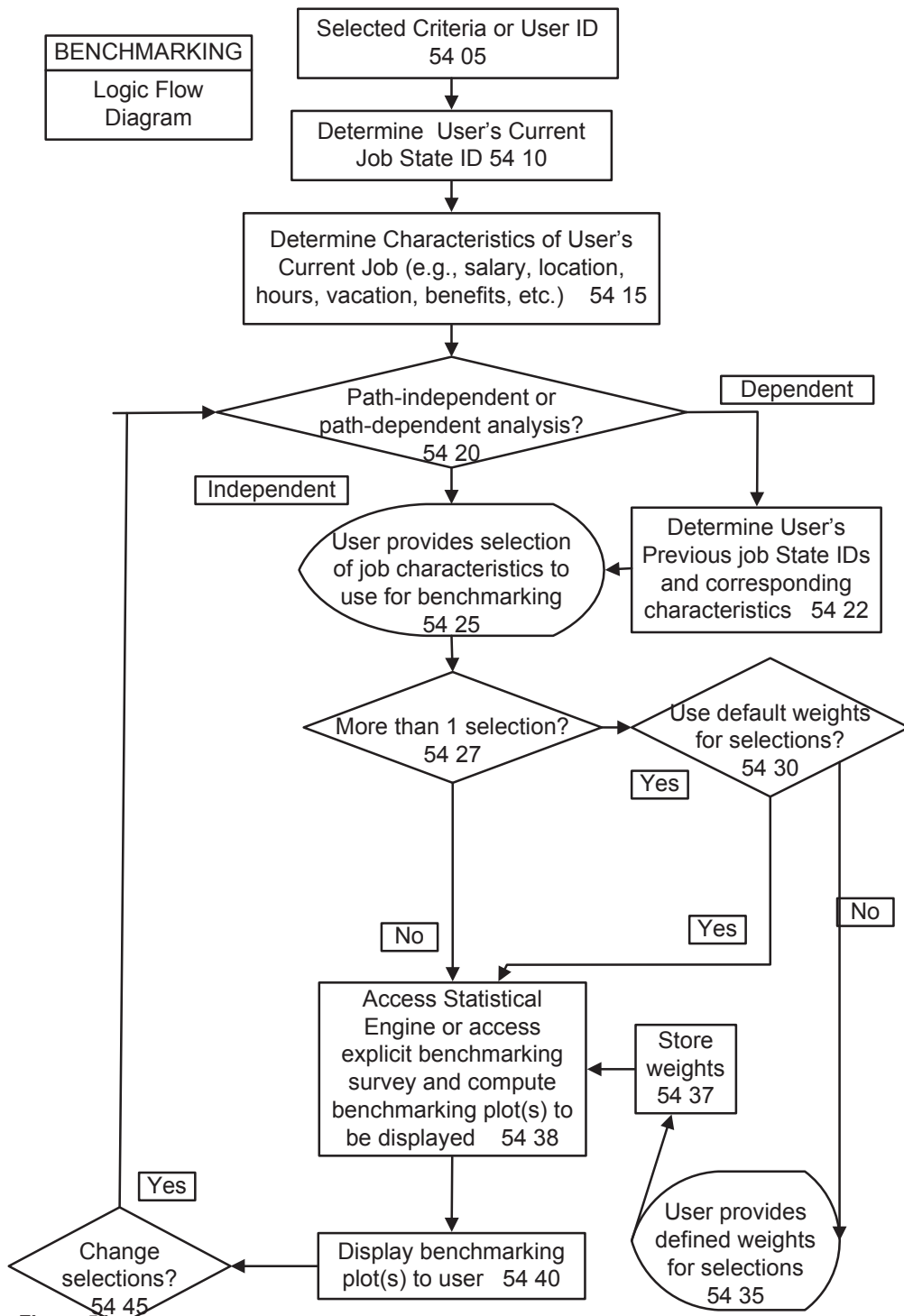


Figure 54

Figure 55

Please select job characteristics to use for benchmarking

<input checked="" type="checkbox"/>	Region	location 55 52
<input checked="" type="checkbox"/>	Title	55 54
<input checked="" type="checkbox"/>	Salary	
<input checked="" type="checkbox"/>	Vacation	
<input type="checkbox"/>	Benefits	
<input type="checkbox"/>	Education	55 56

55 50

What's your salary?
See where you rank compared to others taking this survey

55 60

What's your salary and vacation time?
See where you rank compared to others taking this survey

Your salary is \$50,000 **55 66**

This is correct

I also earn commission:

monthly **55 65**

annually **55 67**

On selecting link, you may be asked to log in to your account.

Your Commission
Commission \$ USD **Confirm**

What's your salary and vacation time?
See where you rank compared to others taking this survey

55 90

What's your weighted salary and vacation time?
See where you rank compared to others taking this survey

Weight for Salary (0-100%) **Confirm**

55 80

What's your salary and vacation time?
See where you rank compared to others taking this survey

55 70

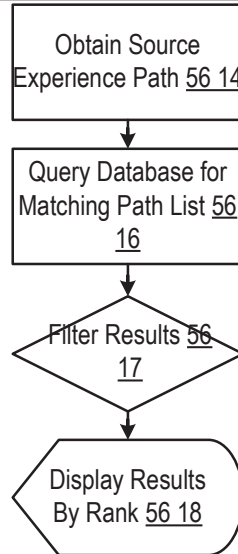
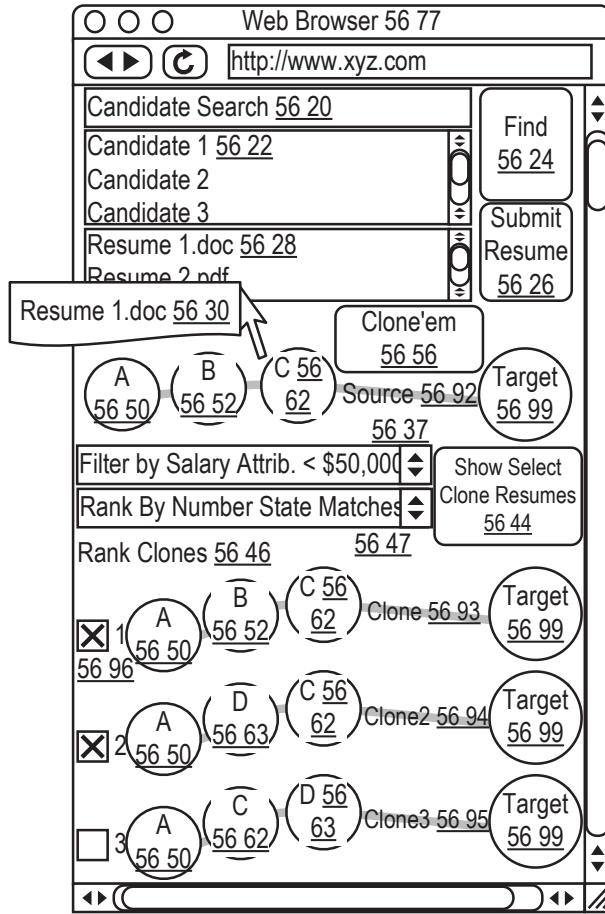
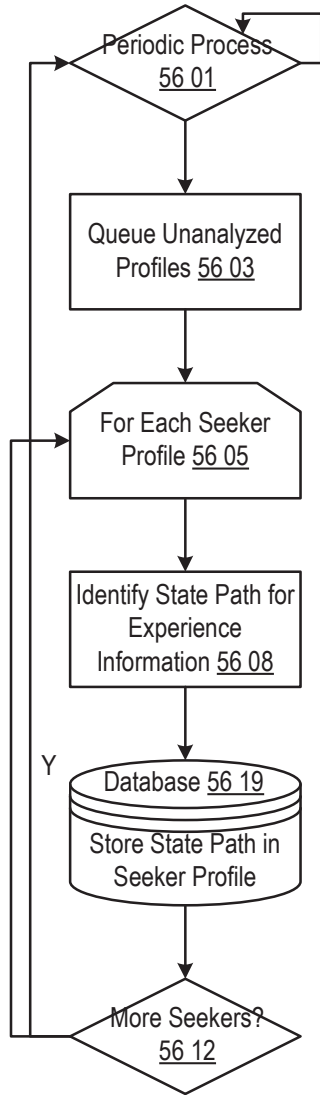


Figure 56

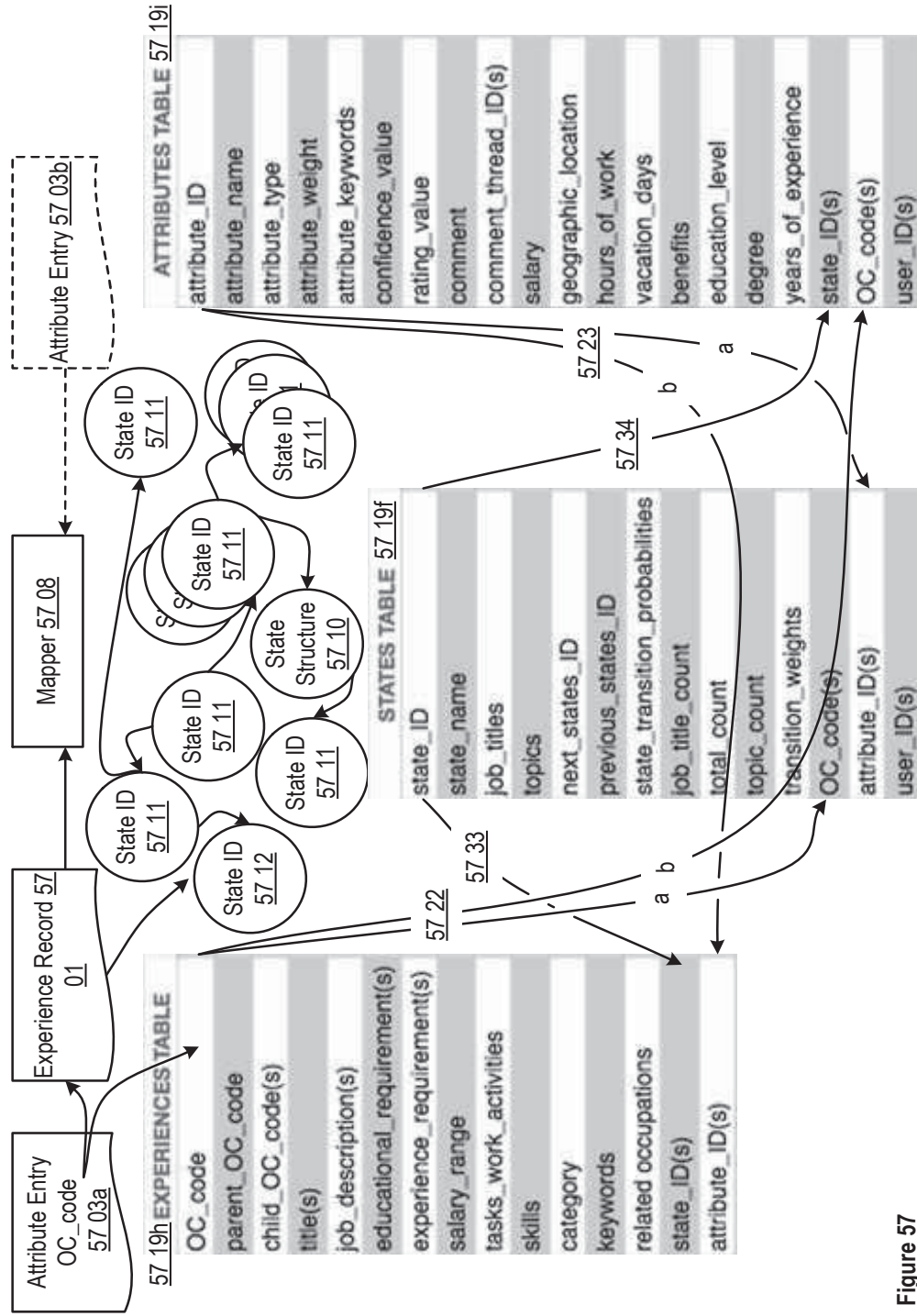
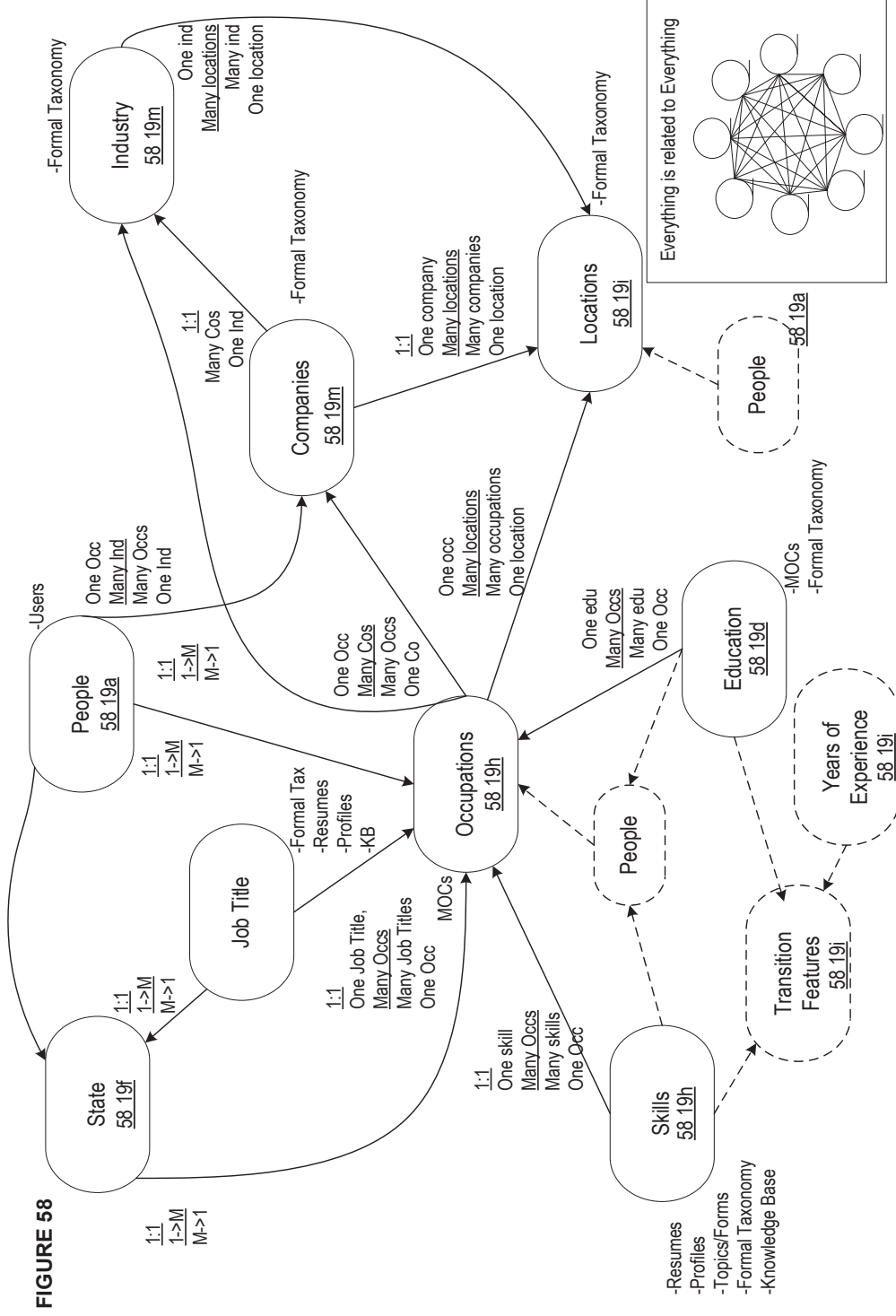


Figure 57



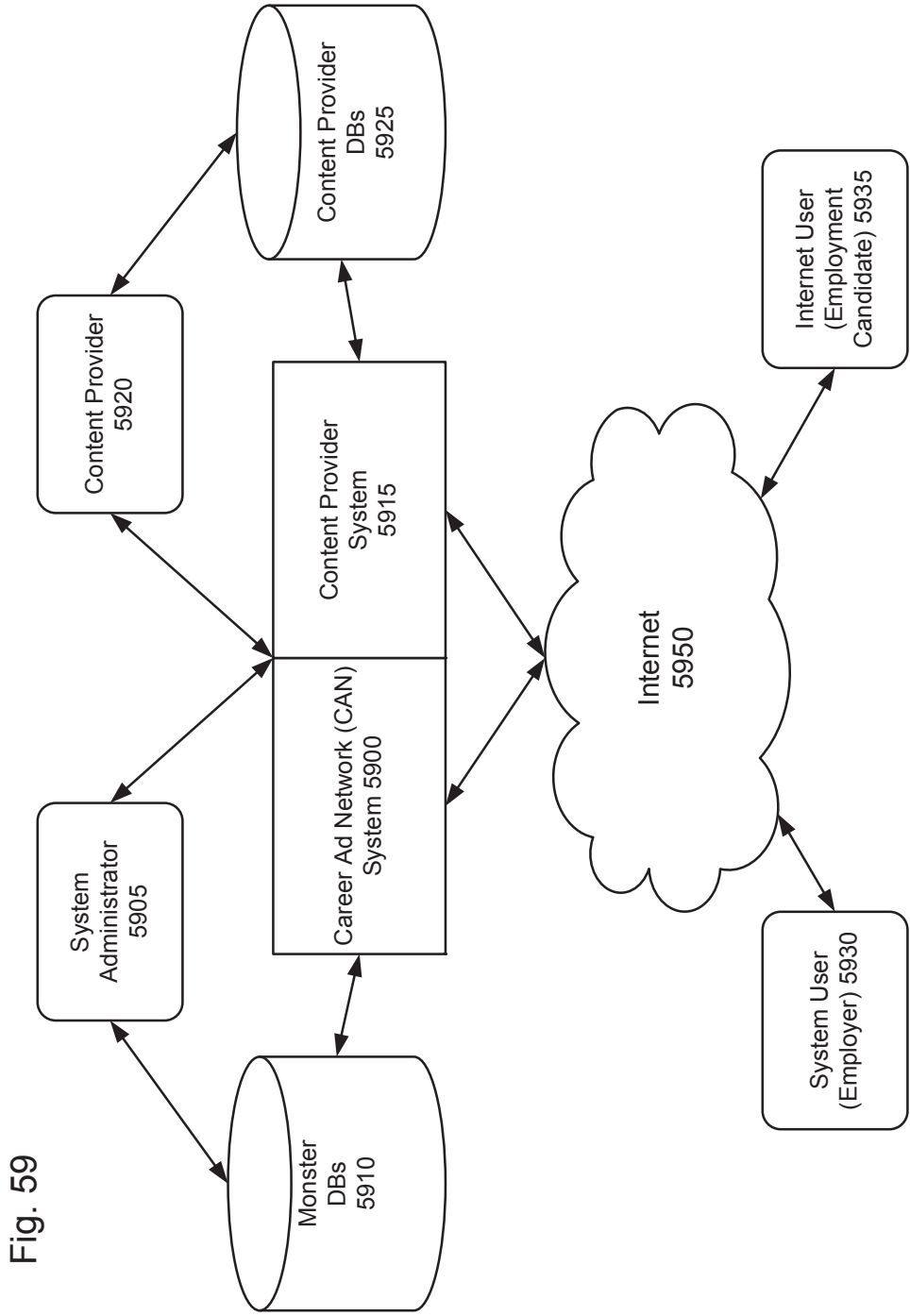


Fig. 59

Fig. 60A

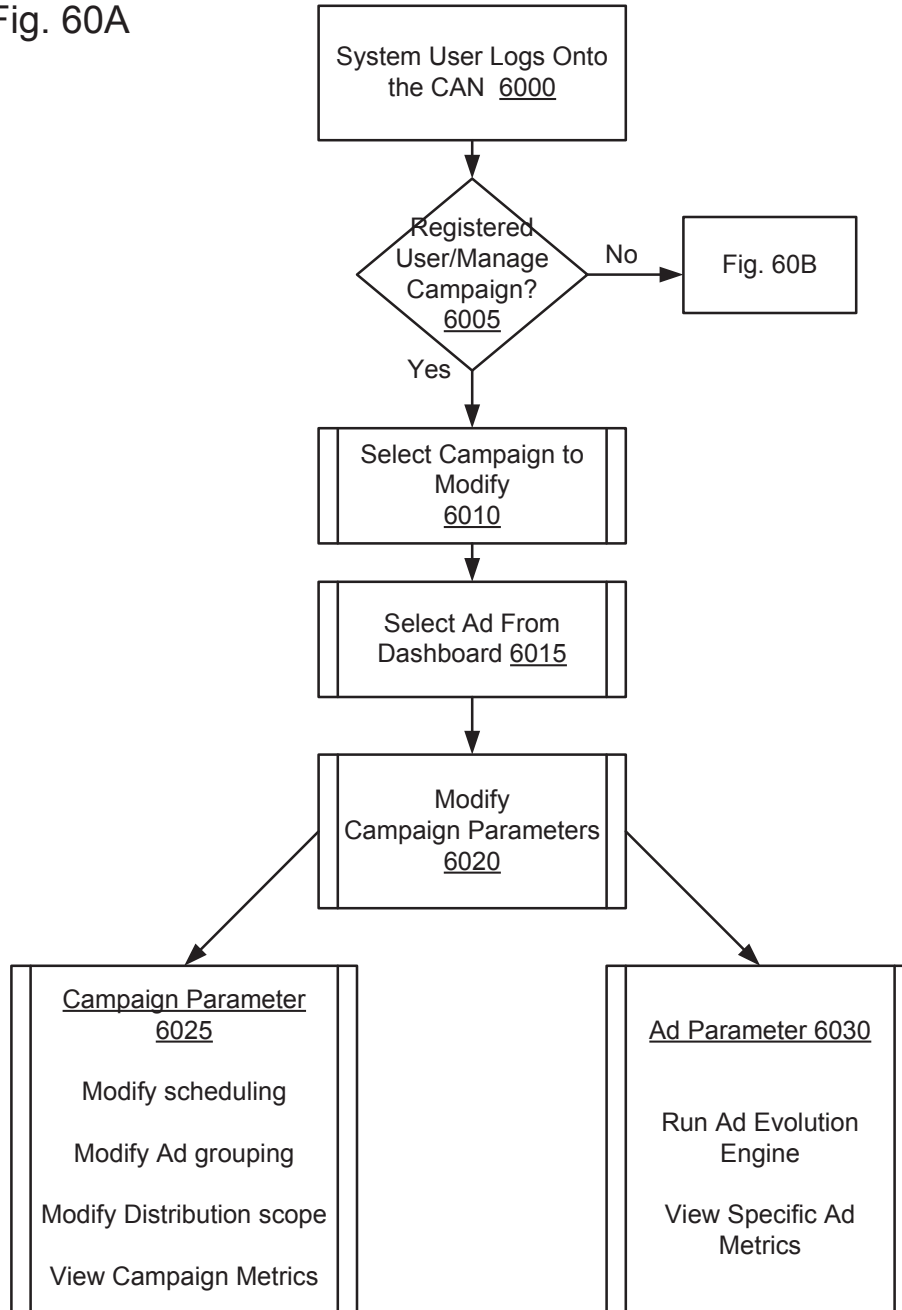
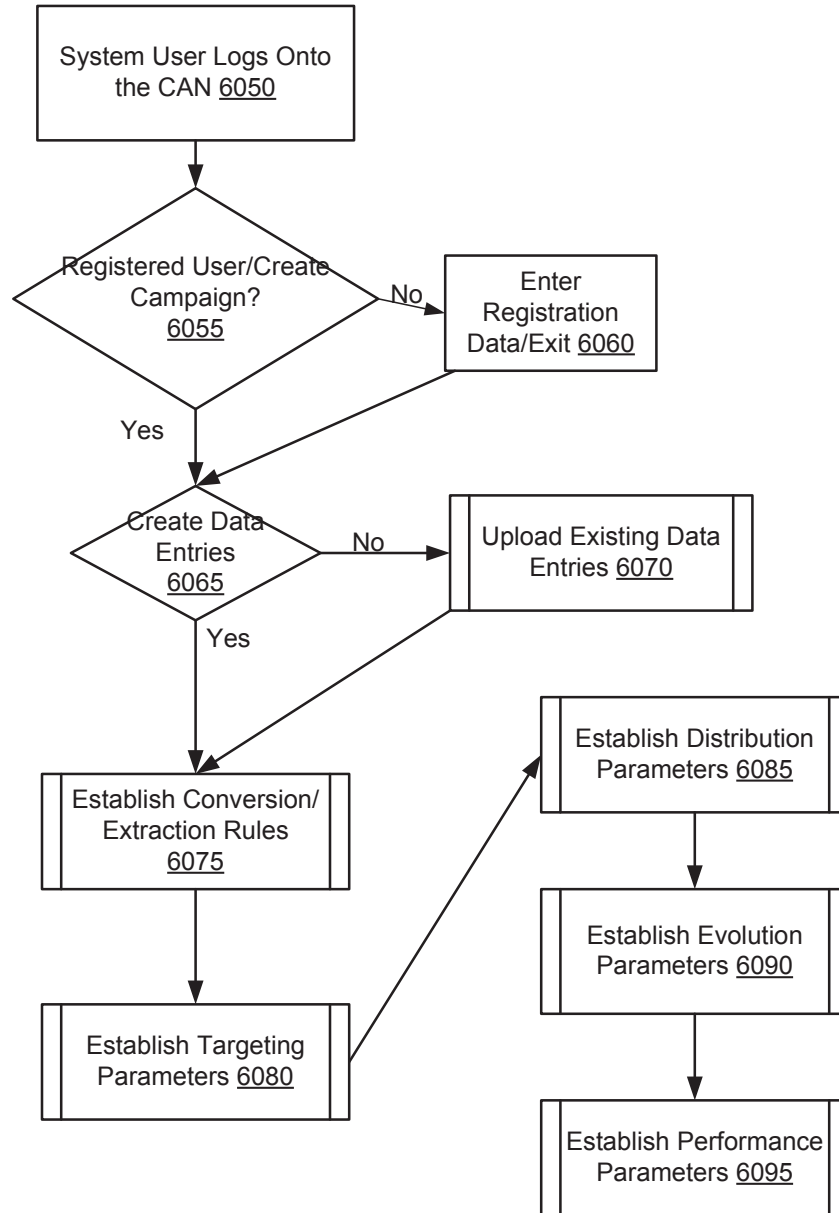


Fig. 60B



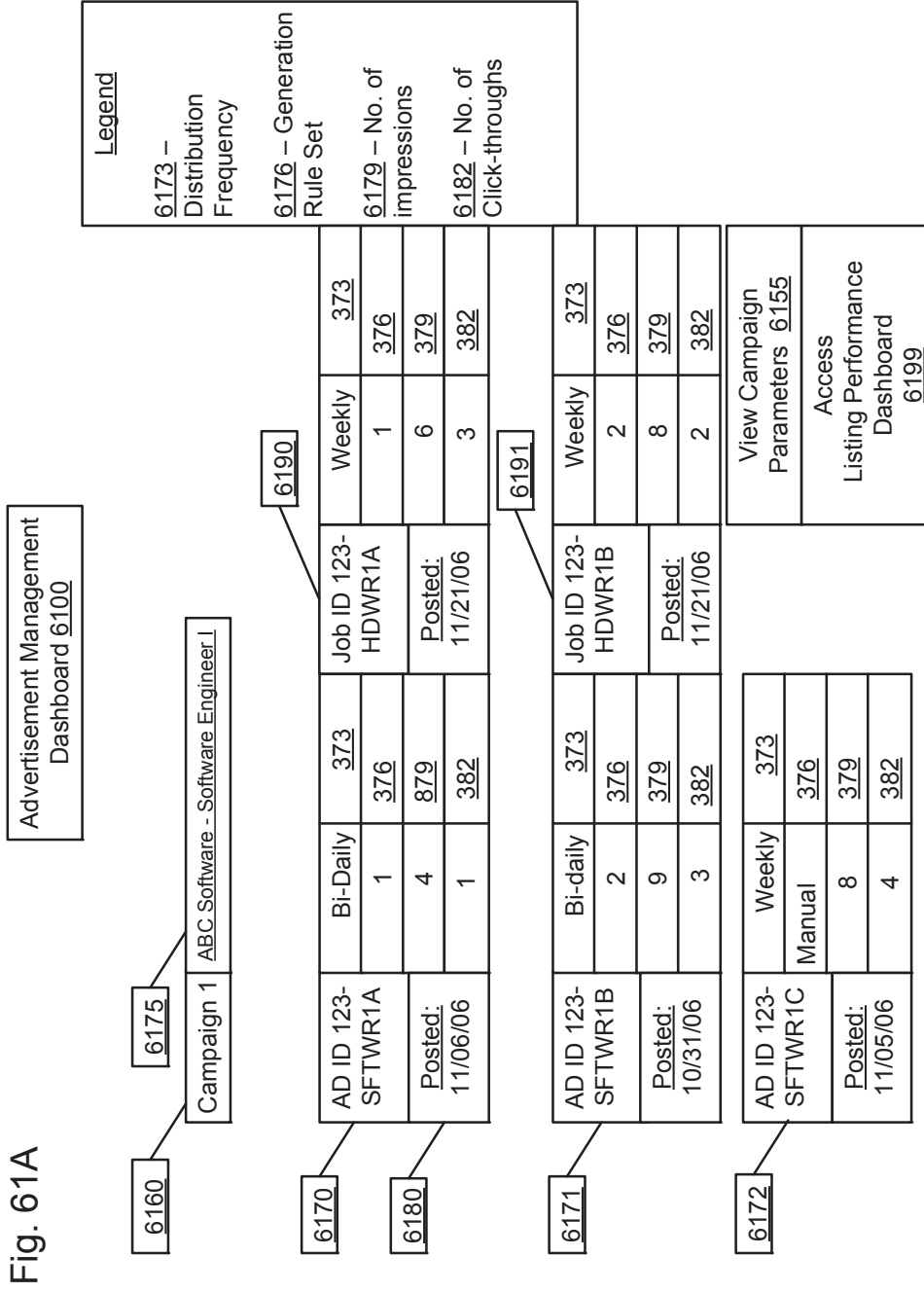


Fig. 61B

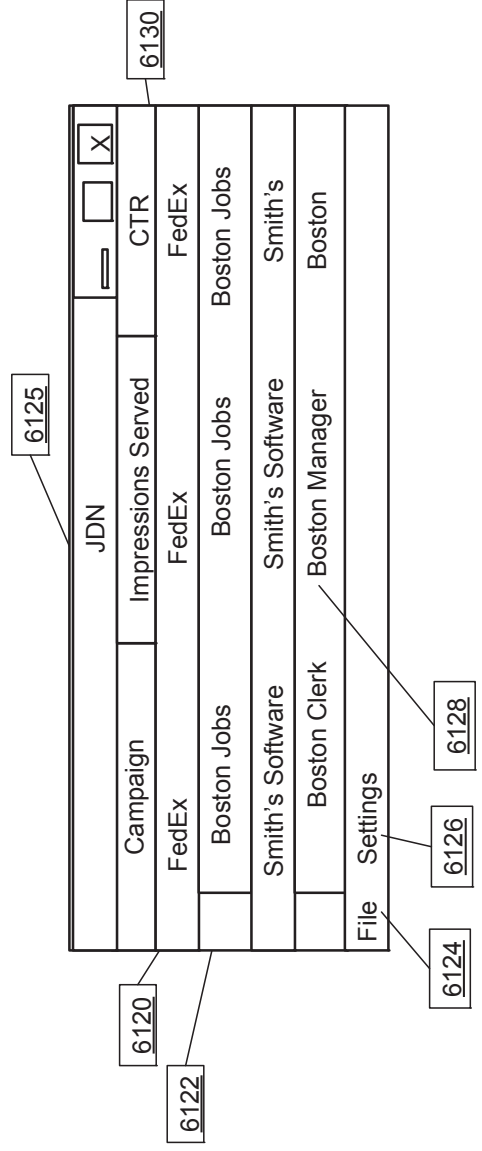


Fig. 62A

6250 **Job Search Site** Search Jobs My Saved Jobs For Employers

6260 **Software Engineer** 6255 **ABC Software Co.**

6265 **Starting Pay:** \$100,000

6270 **Job Location:**
New York, New York

6275 **Job Description:**
Will design, develop and maintain software on Microsoft .Net platform. Serve as technical lead on development efforts, including participating in requirements analysis and design. Will have responsibility for developing solutions for all tiers of the application; code-behinds, controls, middle-tier business components, database layer, etc. Assist in formulating standards and development practices across multiple related teams.

6280 **Qualifications:**

- * Two or more years of extensive experience with C# and ASP.NET is ideal.
- * Knowledge in OO development principles, multi-tier web application design and relational database principles required.
- * History of delivering XML-based, Webservice, Component-Based and Multi-Threaded applications to production is desired.
- * Experience working with large project team development environments.
- * Good fundamental and hands on knowledge of with MS SQL Server.
- * Strong Analytical Skills and good problem solving skills.
- * Experience working in a regionally distributed development group.
- * Prior use of NAnt, NUnit, NDoc and CruiseControl.Net, or similar products would be a plus.

6285 **The Benefits:**

Health benefits: Blue Cross and Blue Shield PPO Program Guardian DMO and Dental PDO Plans Vision care Flexible Spending Accounts Supplement life insurance for employees and their families Discounted auto and homeowners insurance	Employer provided benefits: Life Insurance Short Term and Long Term Disability Plans Employee Assistance Plan Competitive Paid Time Off program Education Tuition Reimbursement Plan for Continuing Education
---	---

Job Search Site **APPLY NOW**
6290

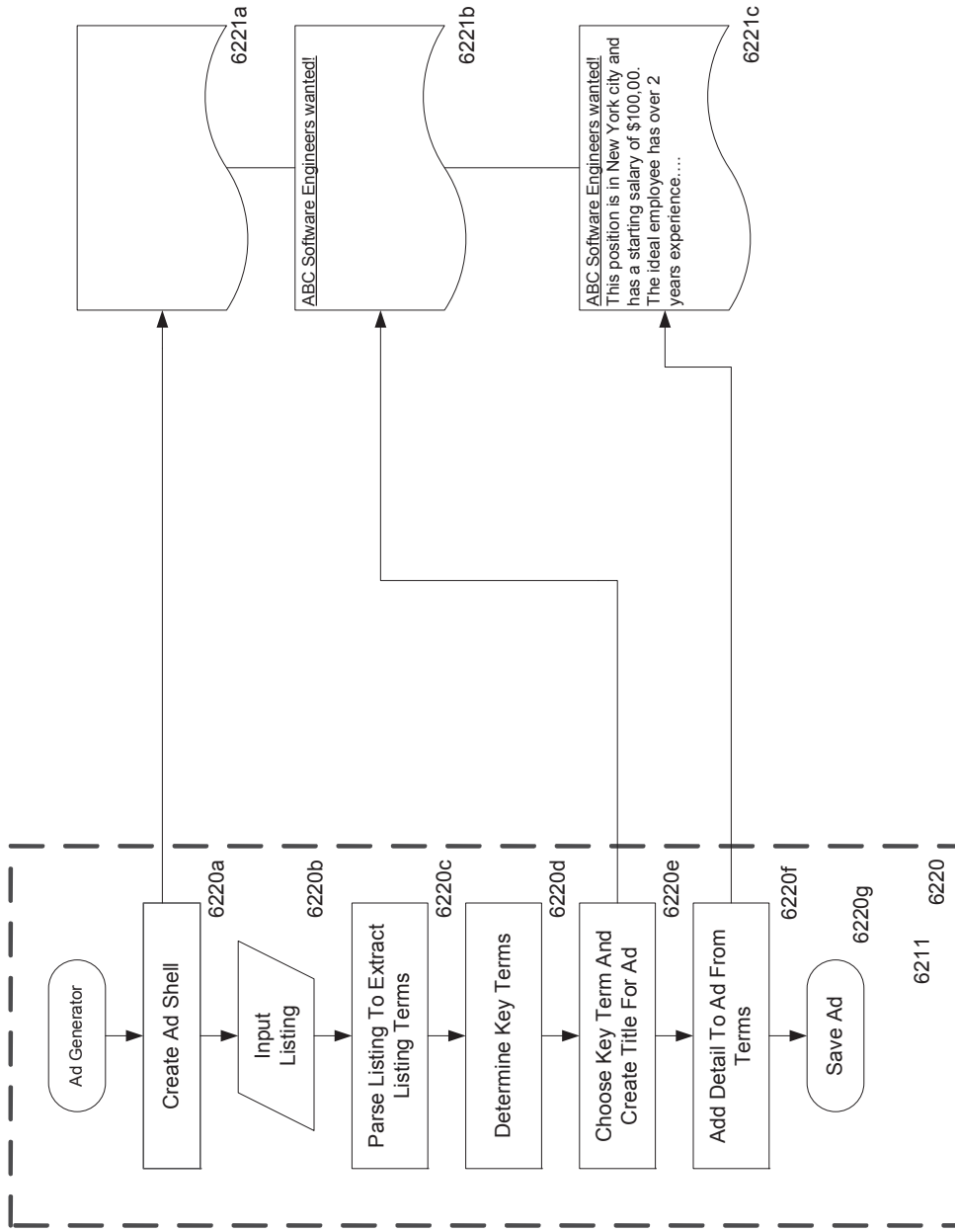


Fig. 62B

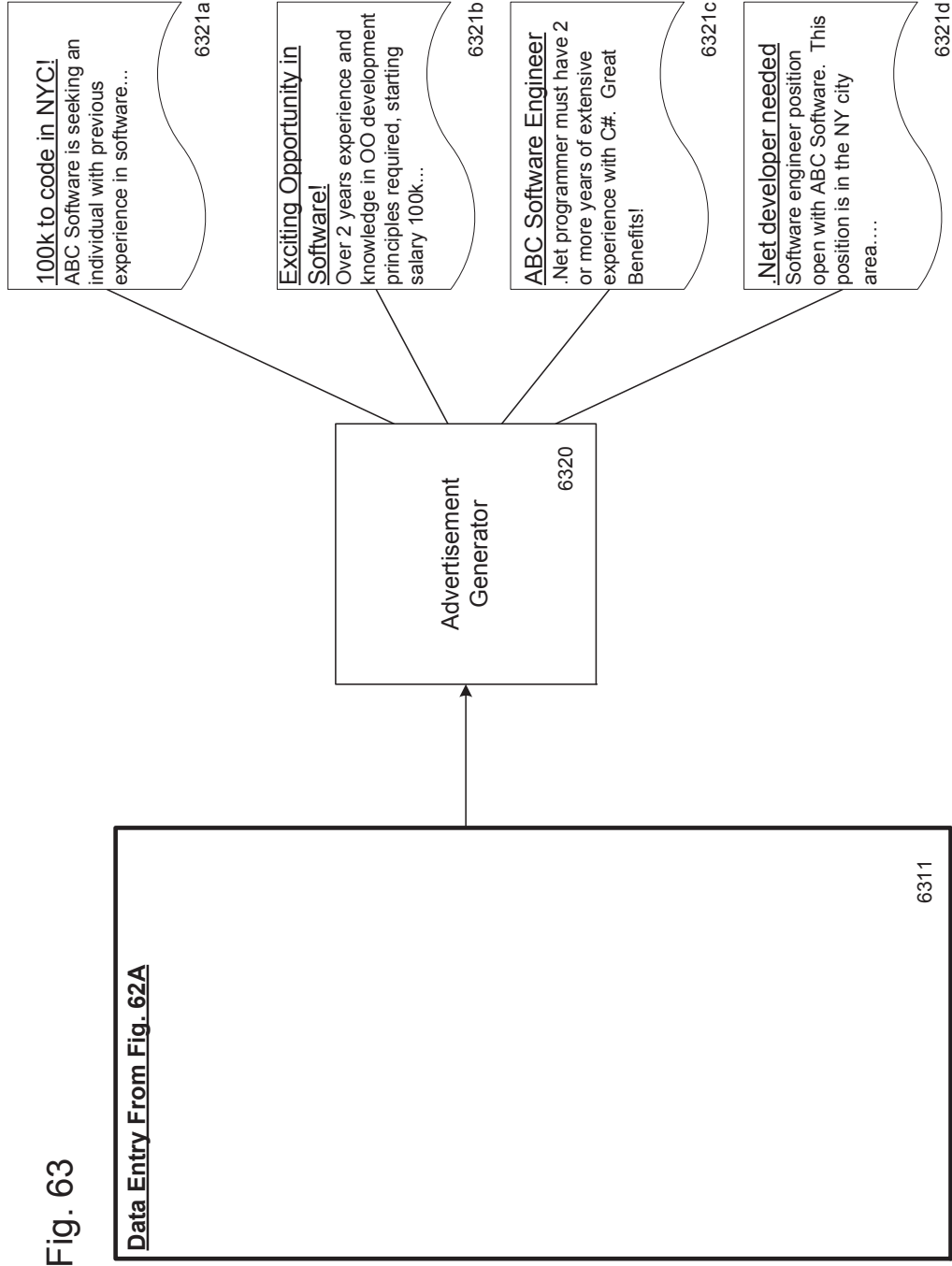


Fig. 63

Data Entry From Fig. 62A

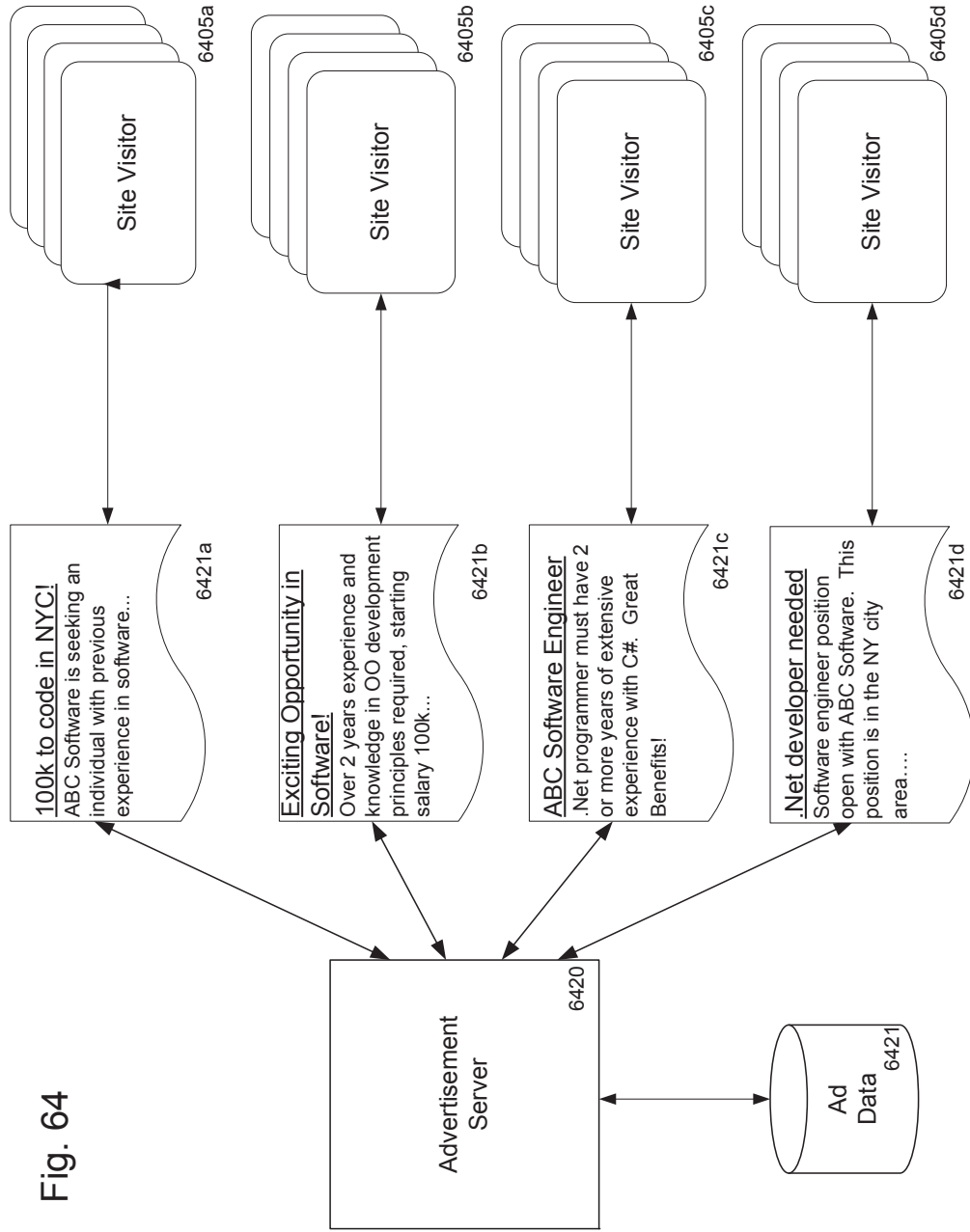


Fig. 64

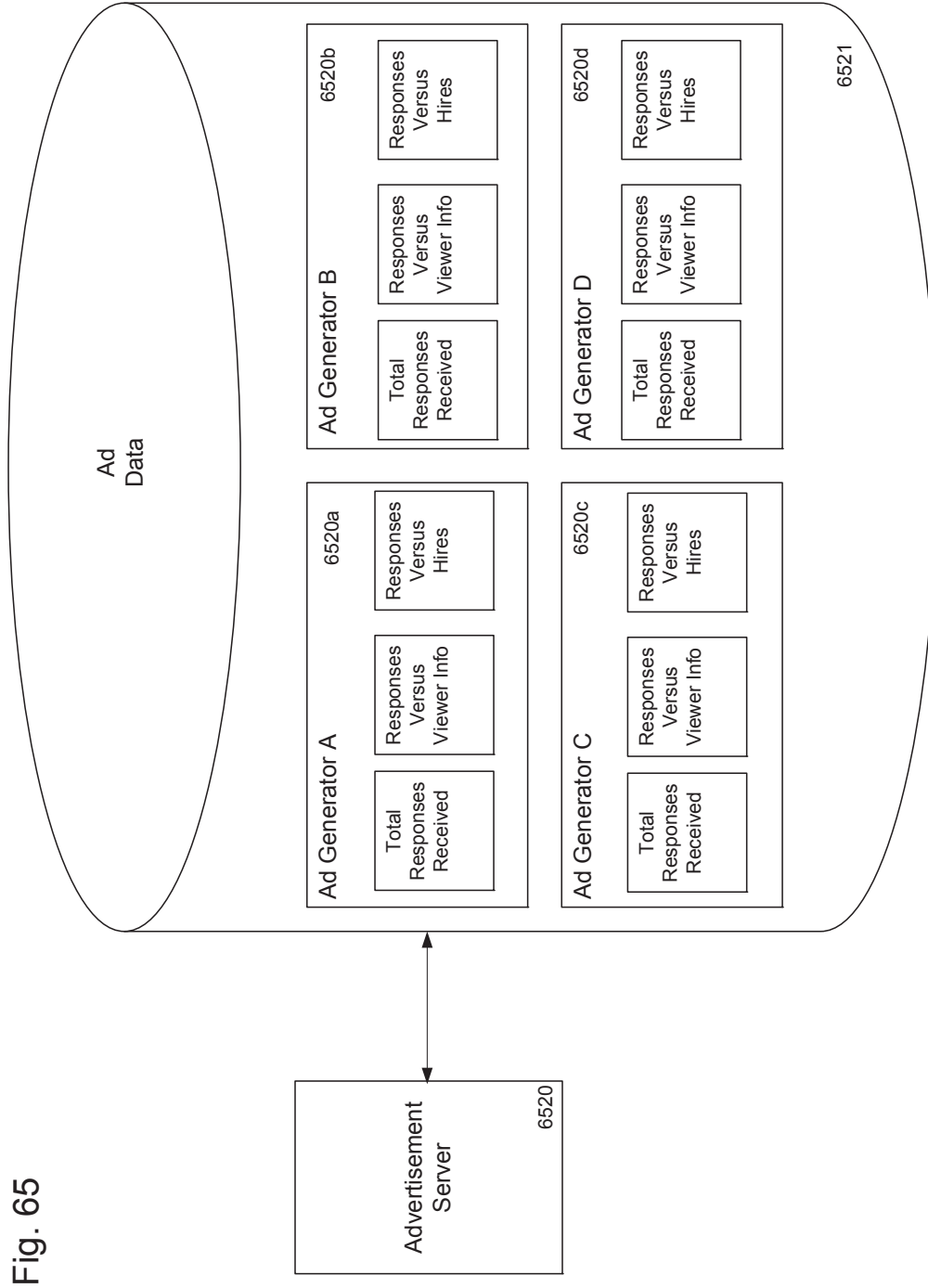
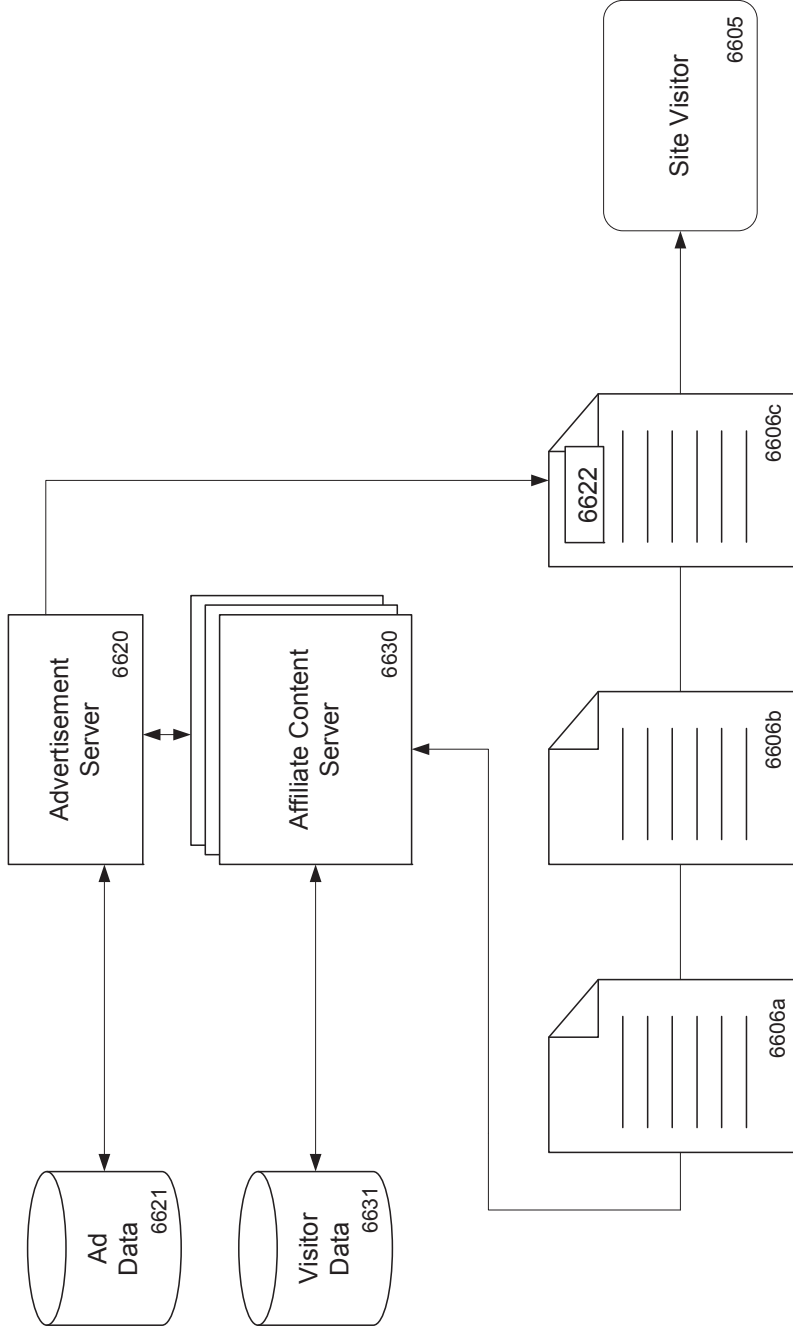


Fig. 65

Fig. 66



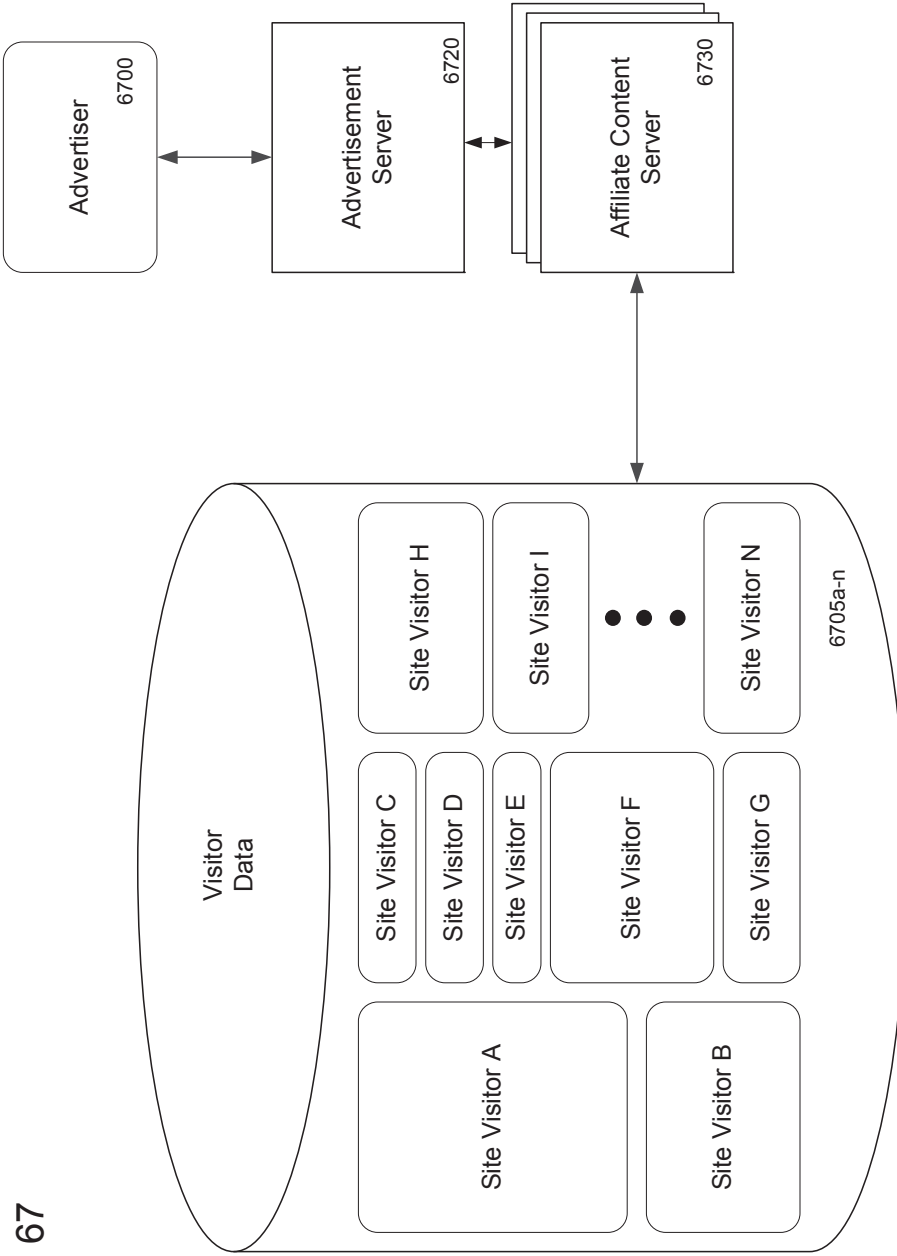


Fig. 67

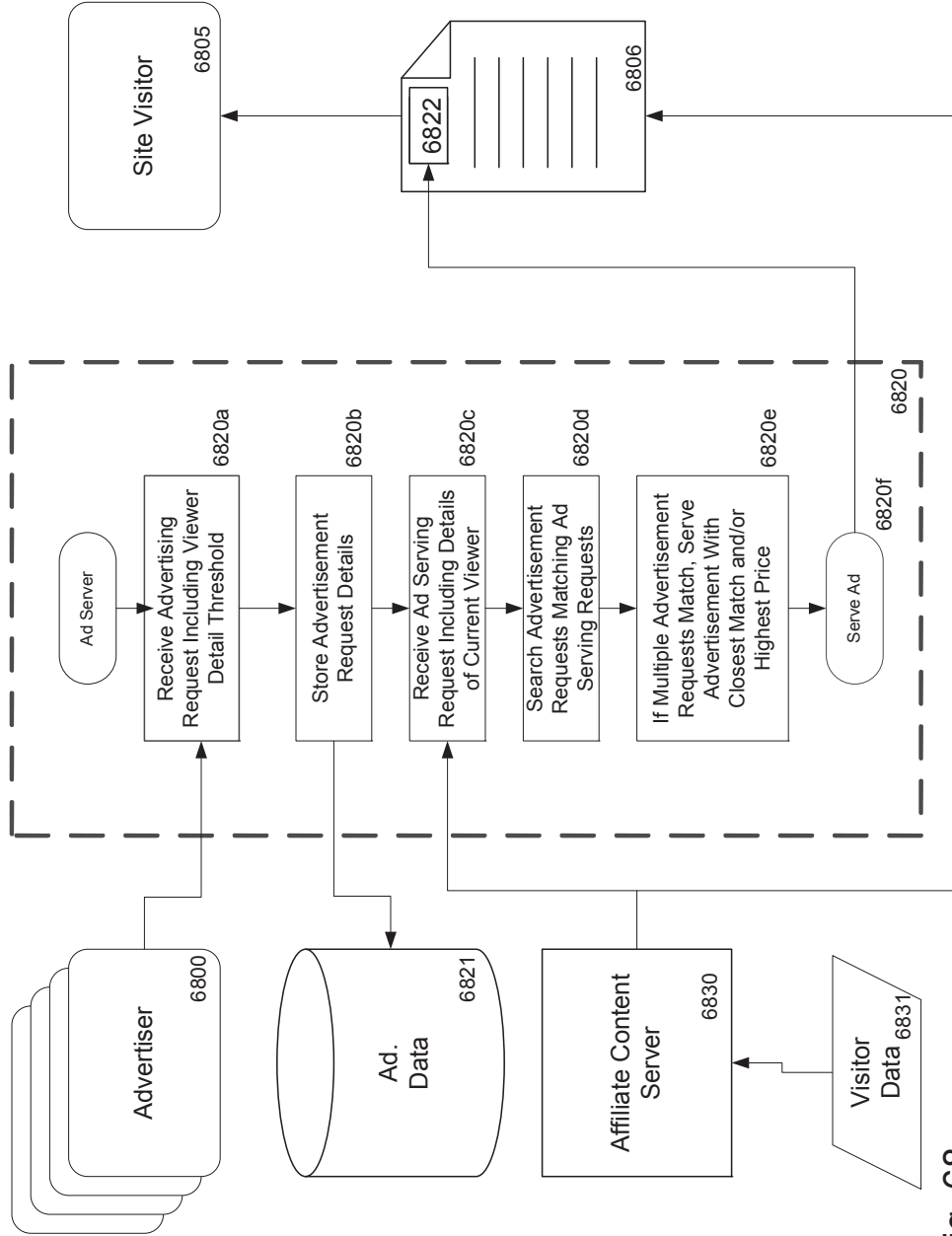
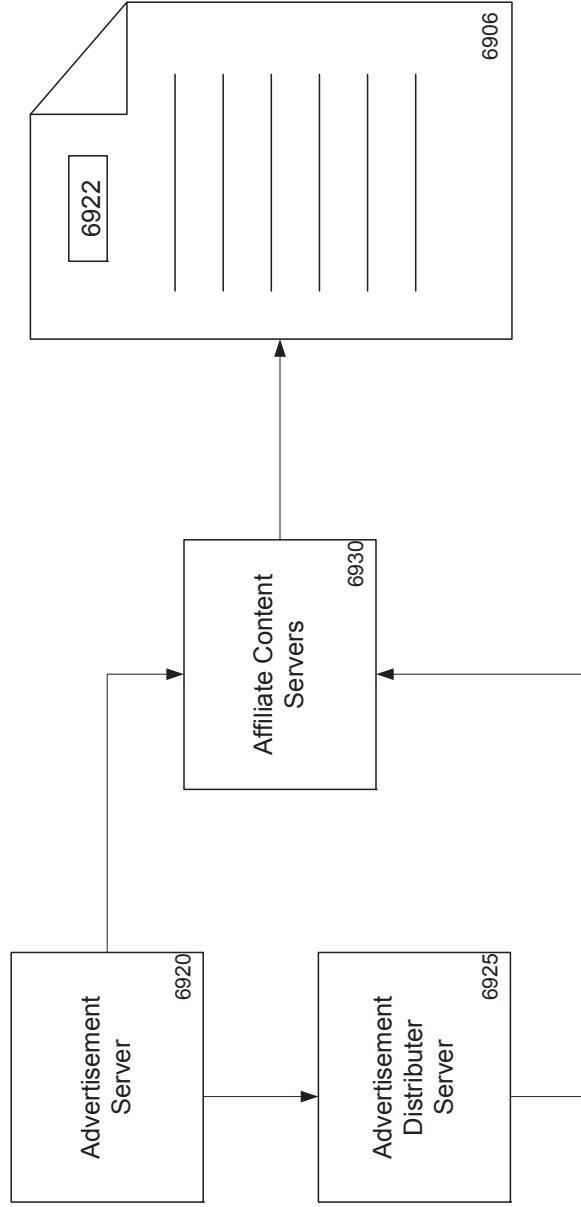


Fig. 68

Fig. 69



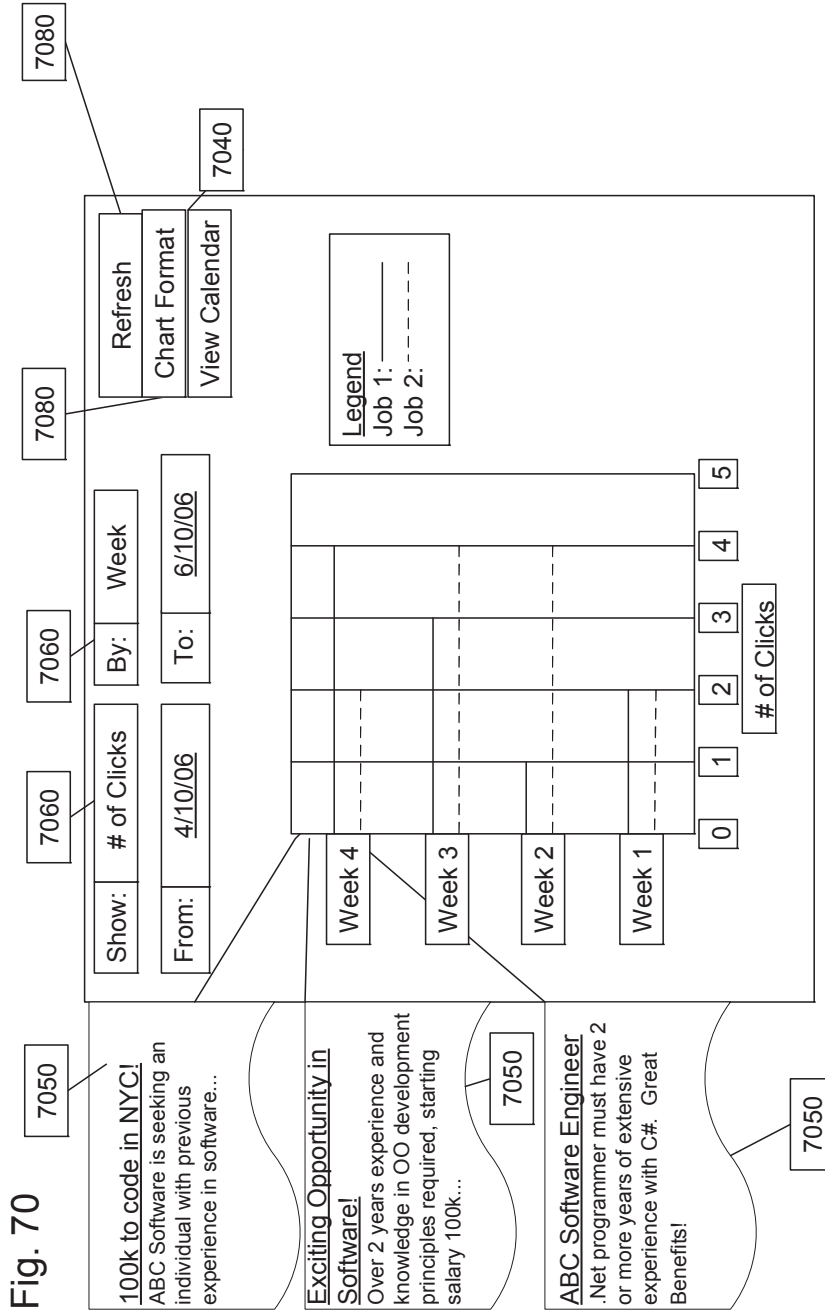
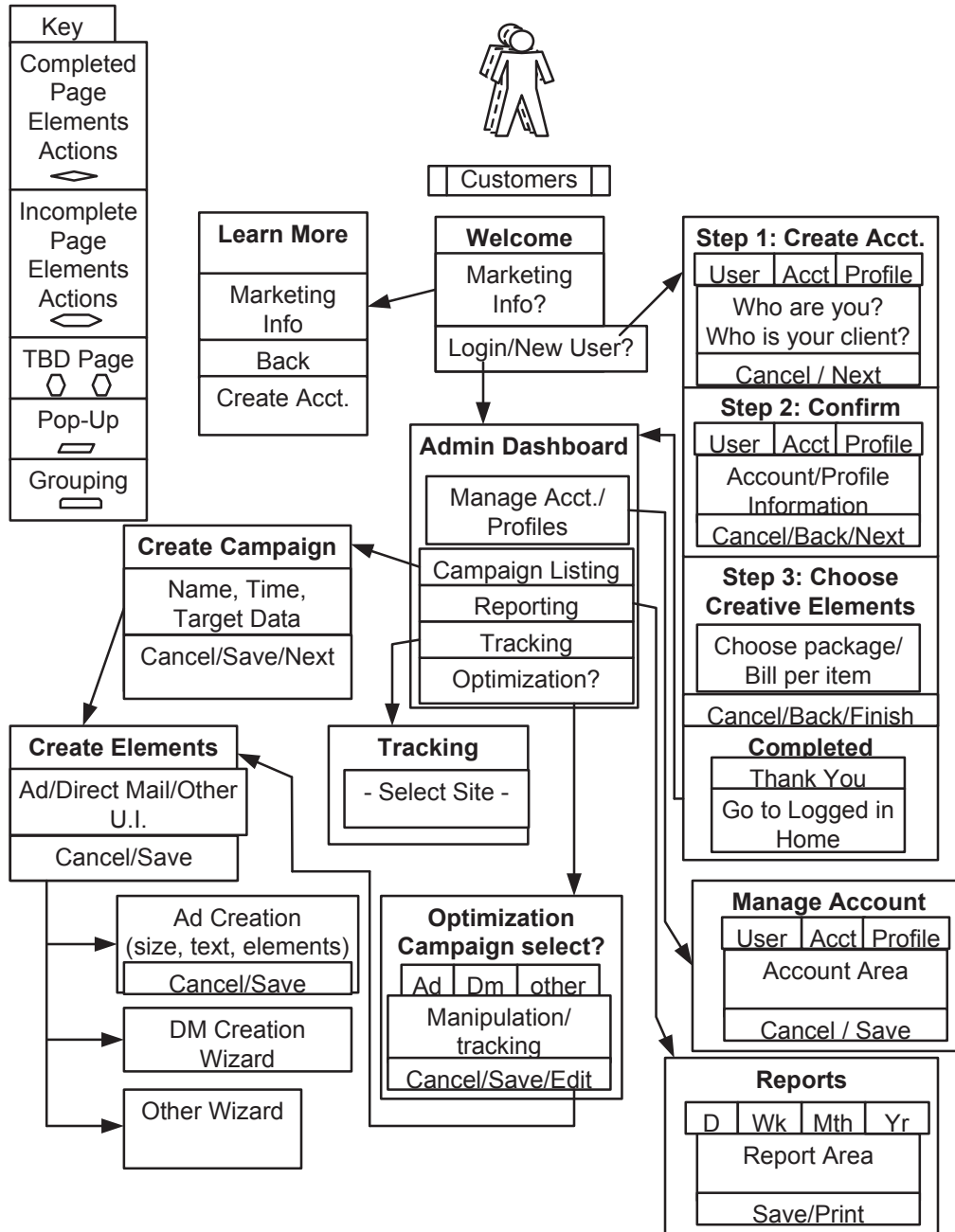


Fig. 71A



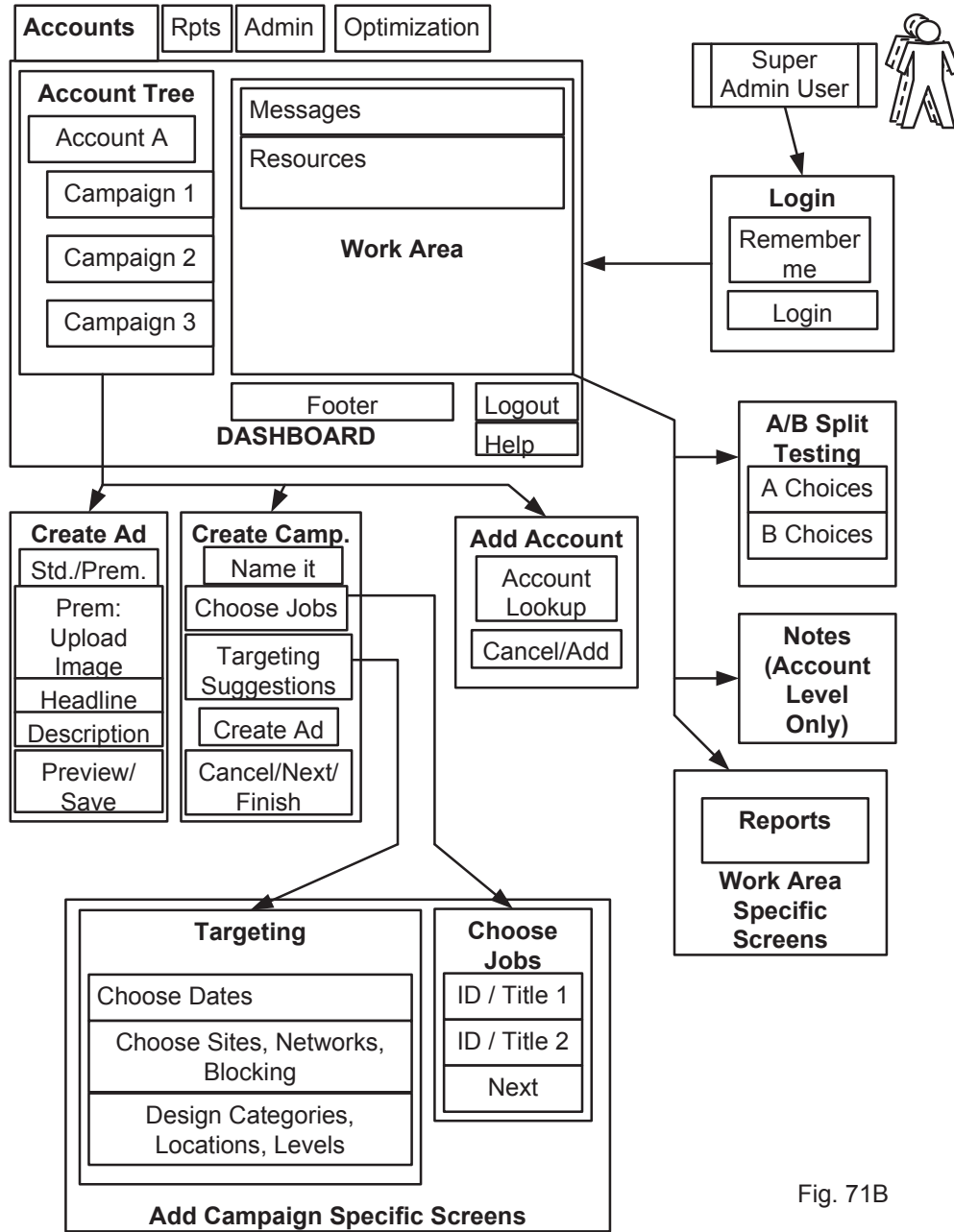
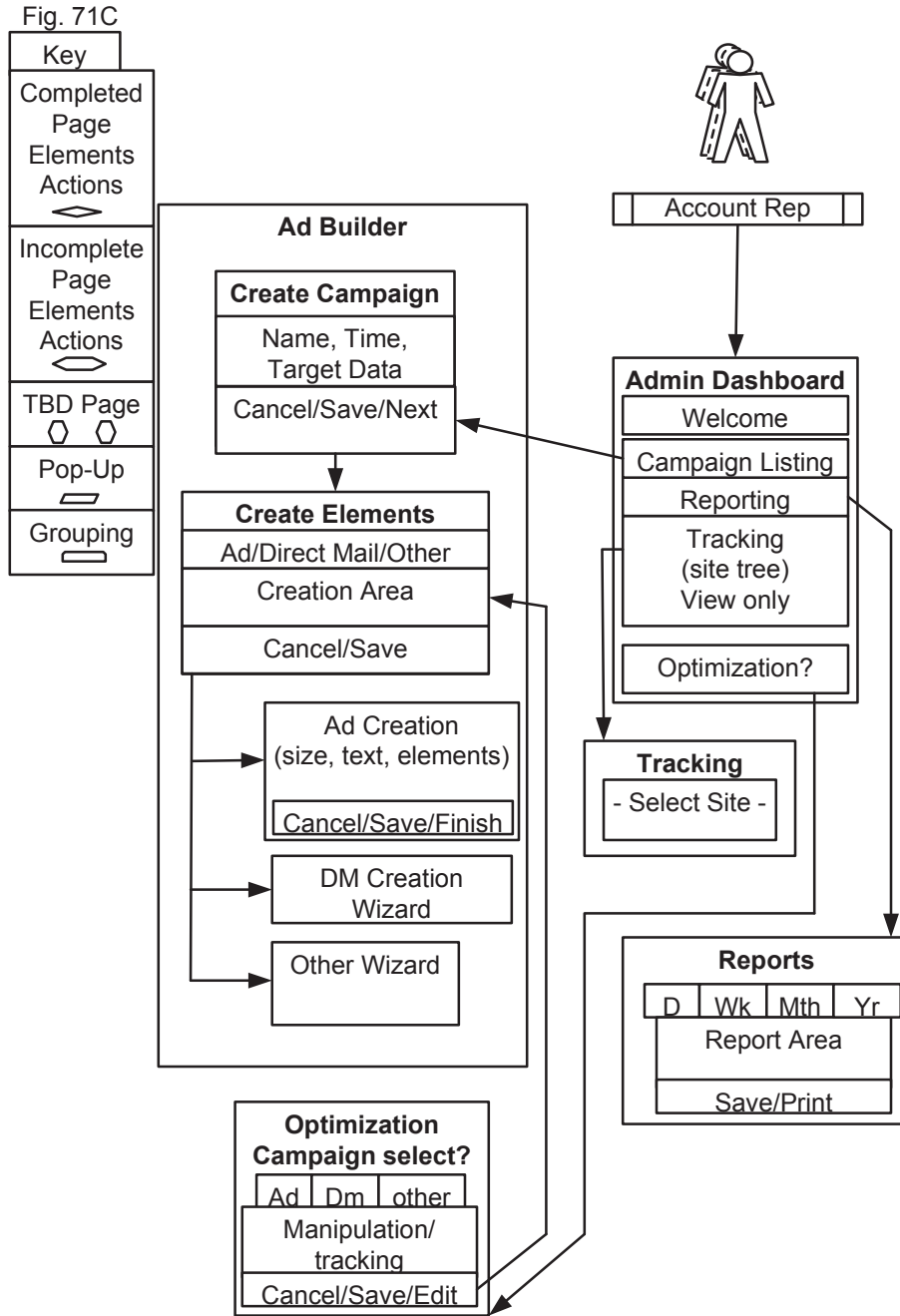


Fig. 71B



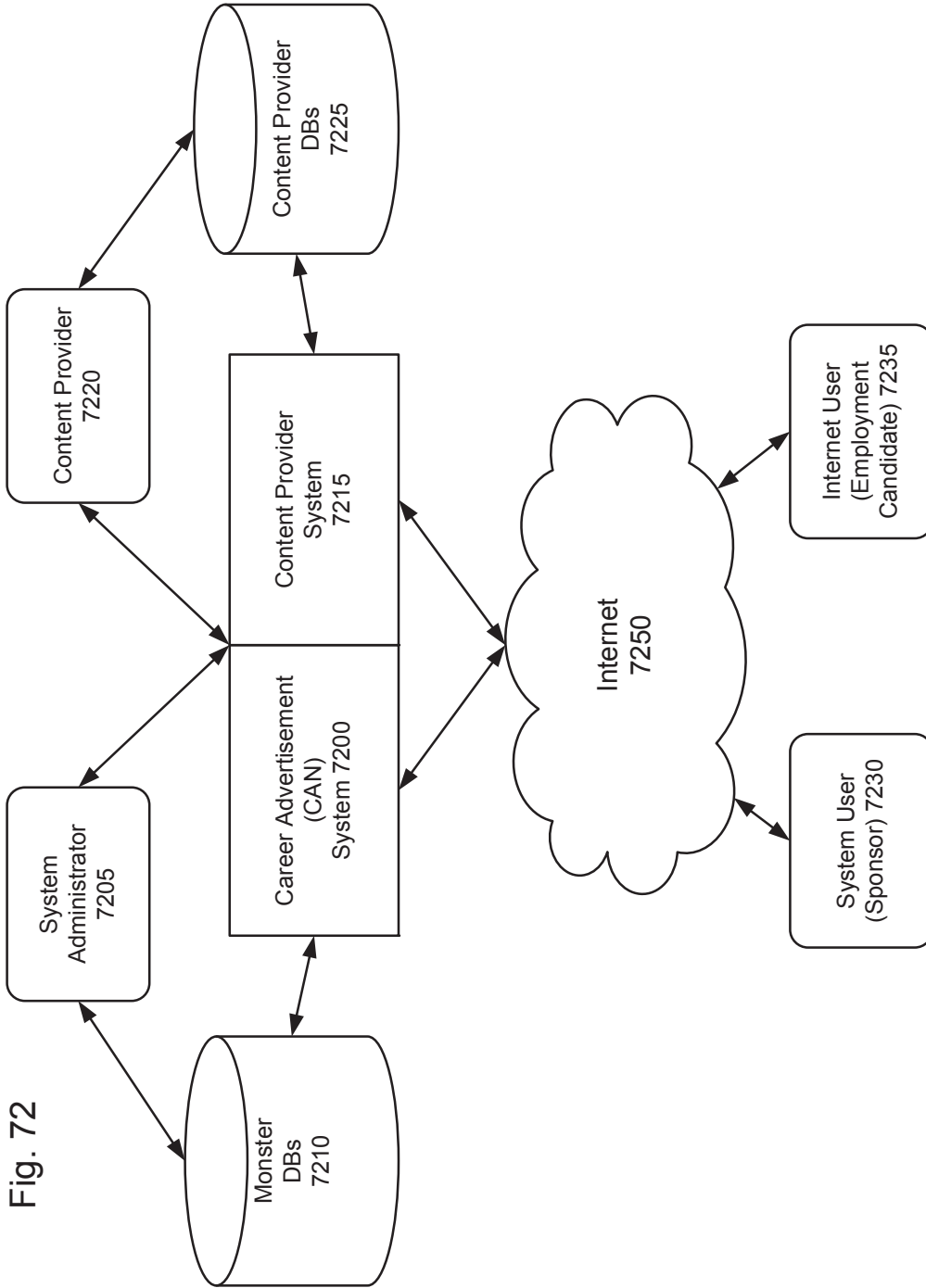


Fig. 72

Fig. 73

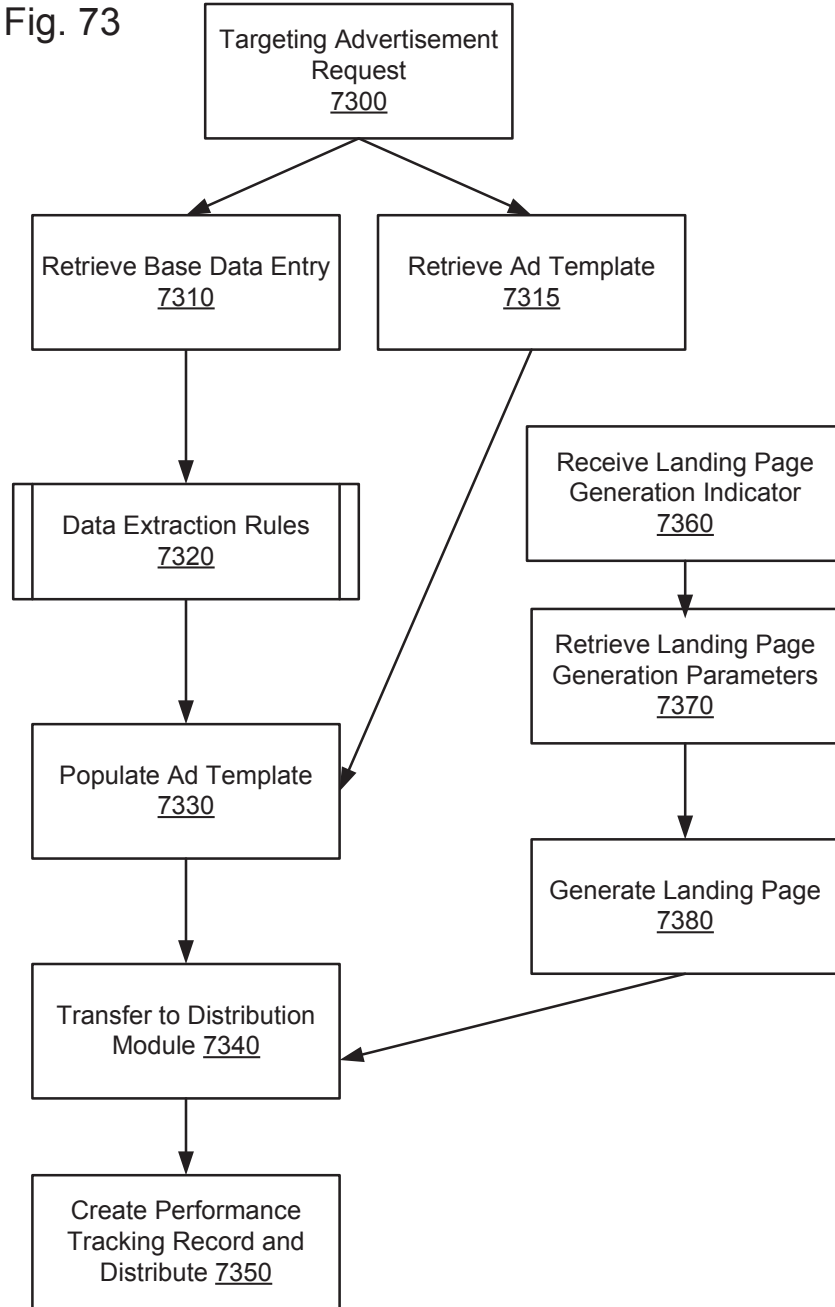


Fig. 74A

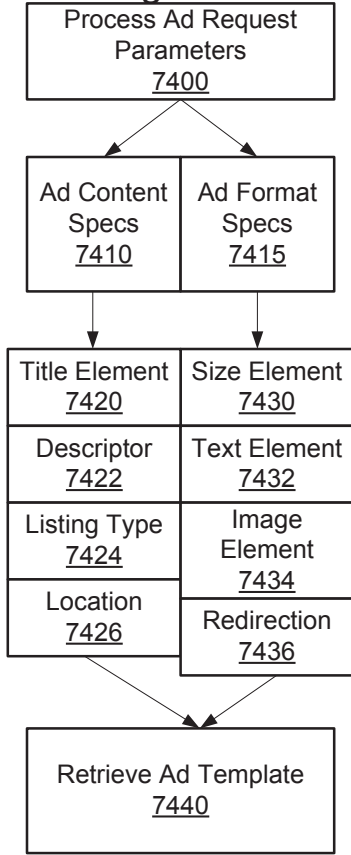


Fig. 74C



Fig. 74D

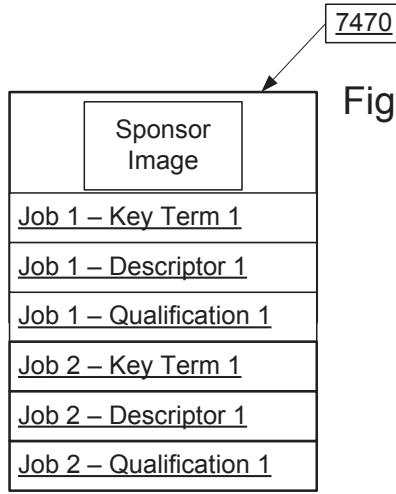


Fig. 74E

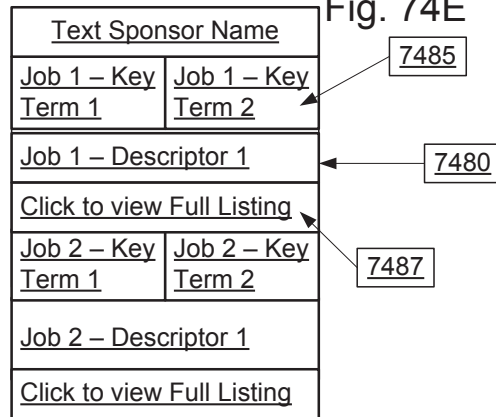


Fig. 74B

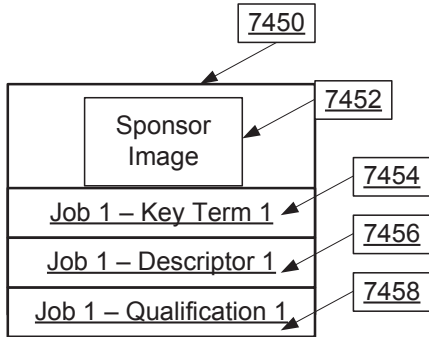


Fig. 75A

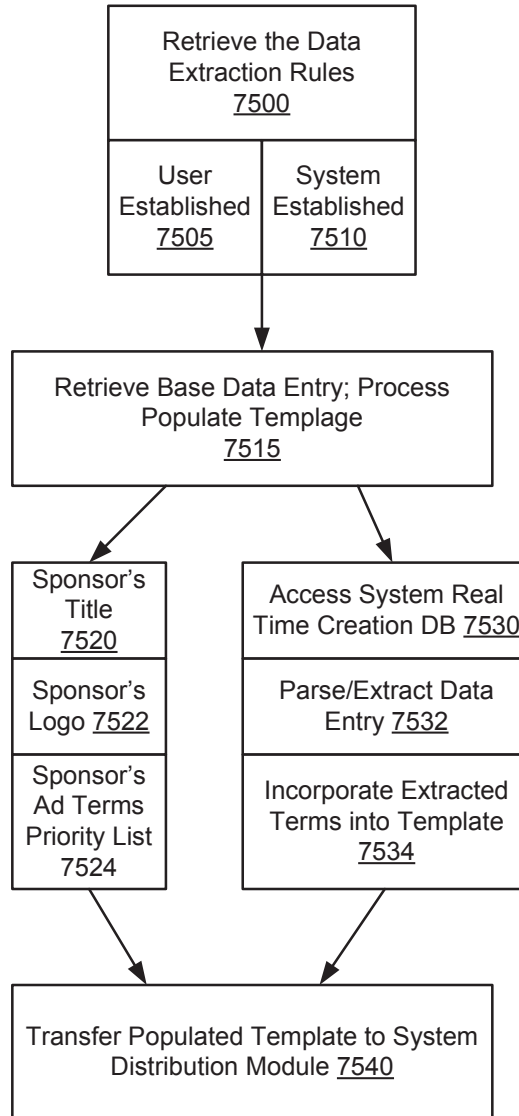


Fig. 75B

Job Search Site		Search Jobs	My Monster	For Employers						
7555	→	Software Engineer								
7560	→	Starting Pay: \$100,000								
7565	→	Job Location: New York, New York								
7570	→	Job Description: Will design, develop and maintain software on Microsoft .Net platform. Serve as technical lead on development efforts, including participating in requirements analysis and design.								
7575	→	Qualifications: <ul style="list-style-type: none"> * Two or more years of extensive experience with C# and ASP.NET is ideal. * Knowledge in OO development principles, multi-tier web application design and relational database principles required. * Strong Analytical Skills and good problem solving skills. * Experience working in a regionally distributed development group. * Prior use of NAnt, NUnit, NDoc and CruiseControl.Net, or similar products would be a plus. 								
		The Benefits: <table style="width: 100%; border: none;"> <tr> <td>Health care:</td> <td>Life Insurance</td> </tr> <tr> <td>Vision care;</td> <td>Short Term and Long</td> </tr> <tr> <td>Flex Spending Account</td> <td>Term Disability Plan</td> </tr> </table>			Health care:	Life Insurance	Vision care;	Short Term and Long	Flex Spending Account	Term Disability Plan
Health care:	Life Insurance									
Vision care;	Short Term and Long									
Flex Spending Account	Term Disability Plan									
Job Search Site		<u>APPLY NOW</u>								

ABC Software Co.

Fig. 75C

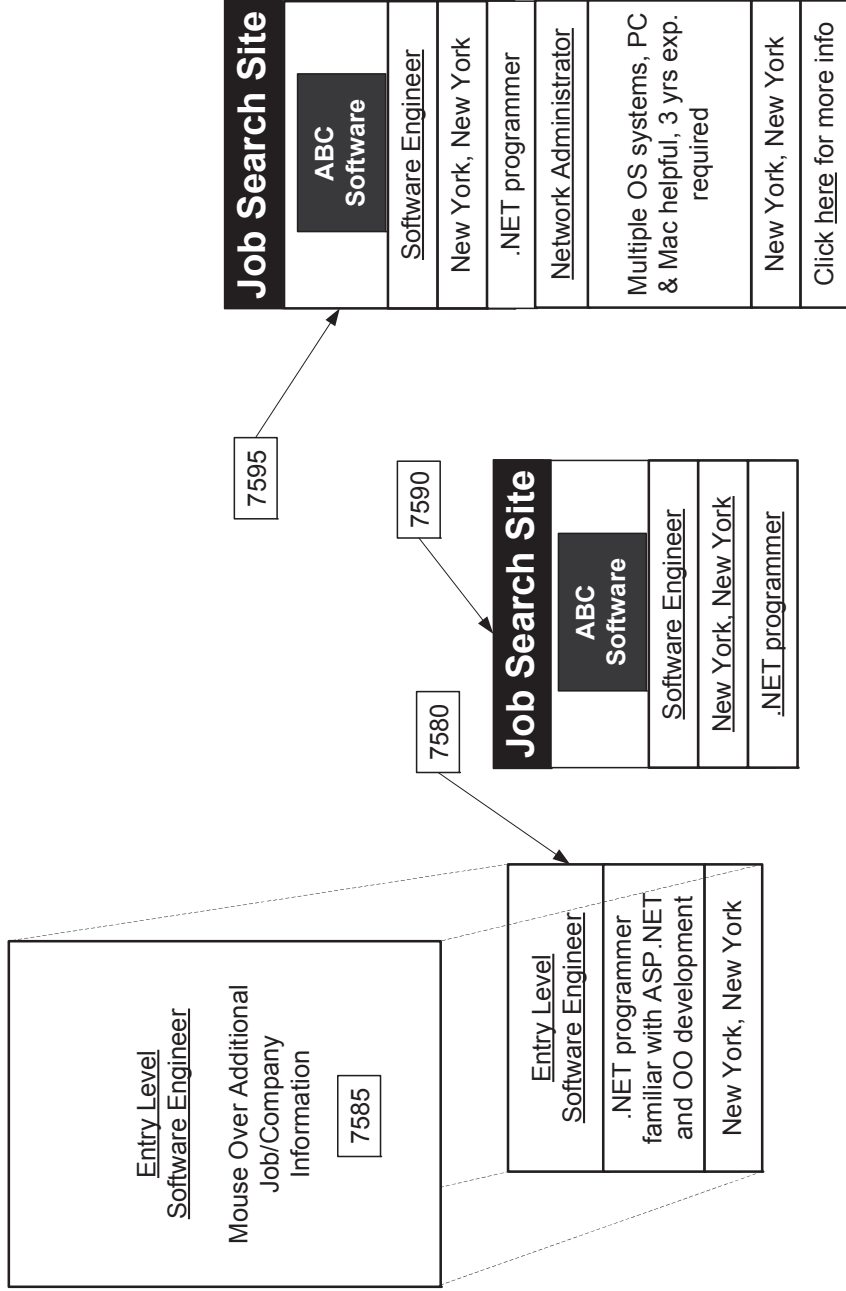
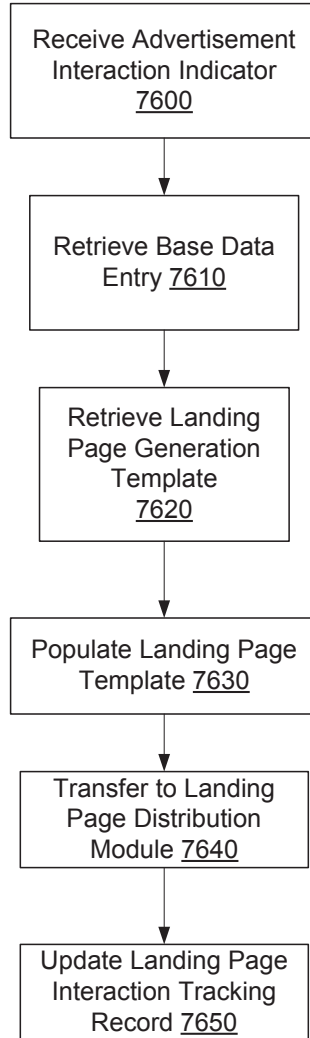
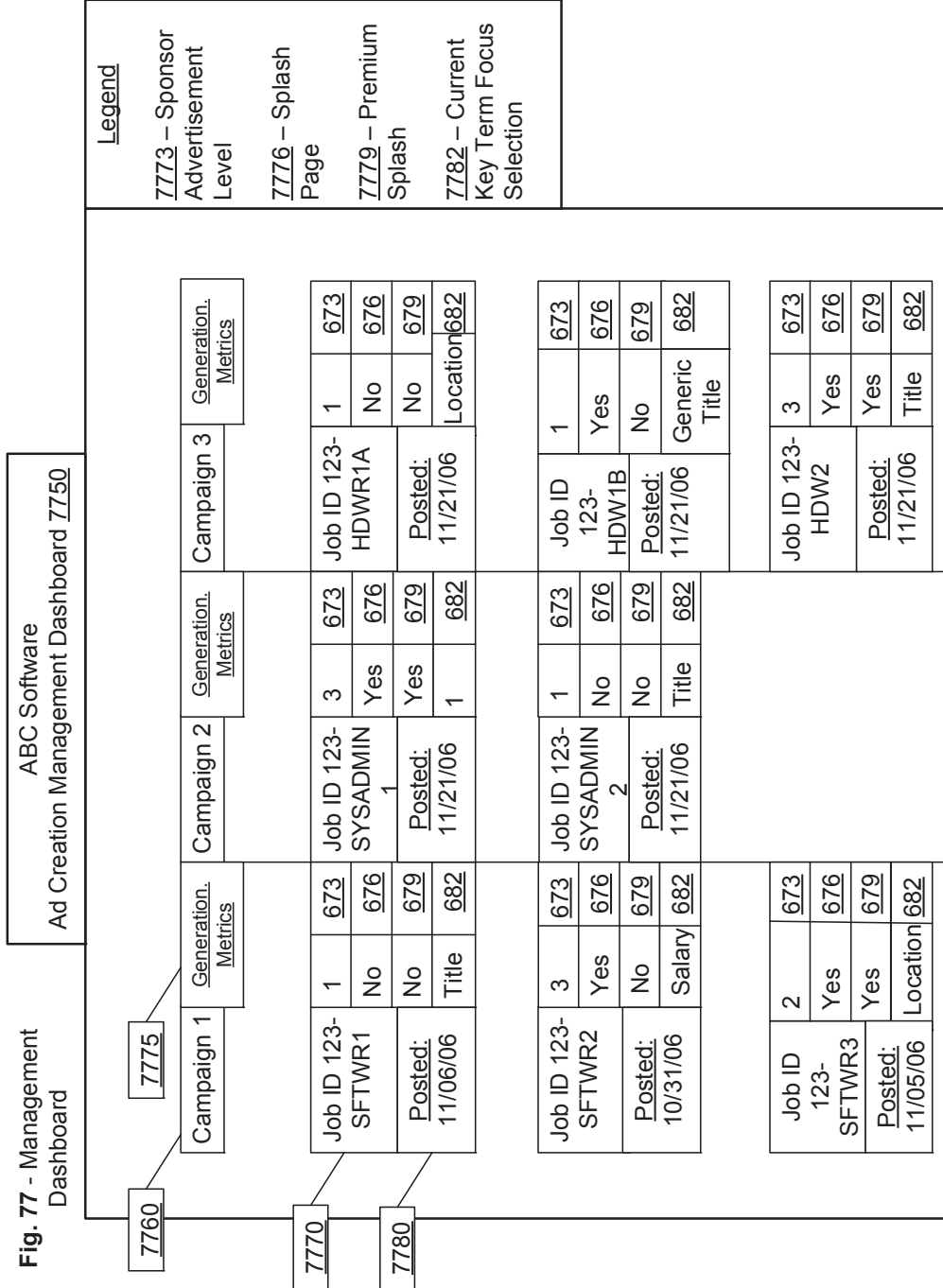


Fig. 76A





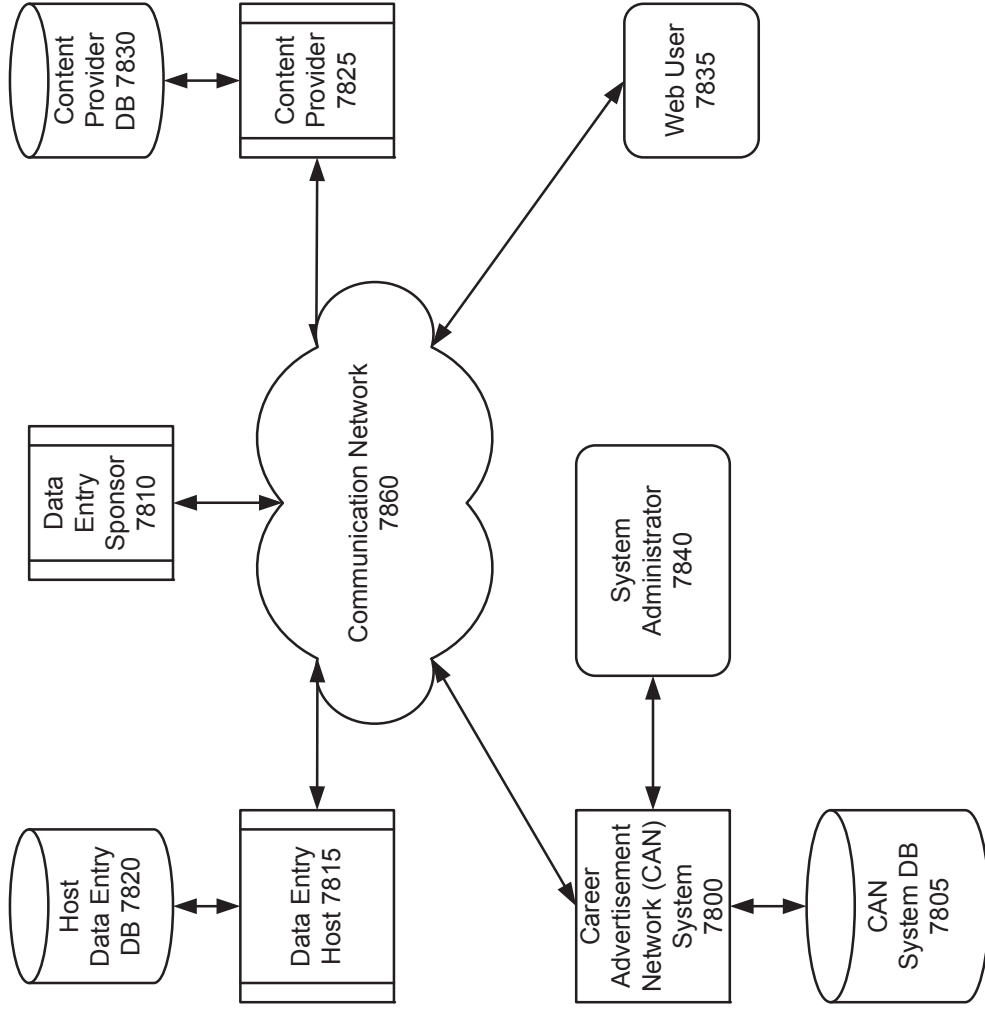
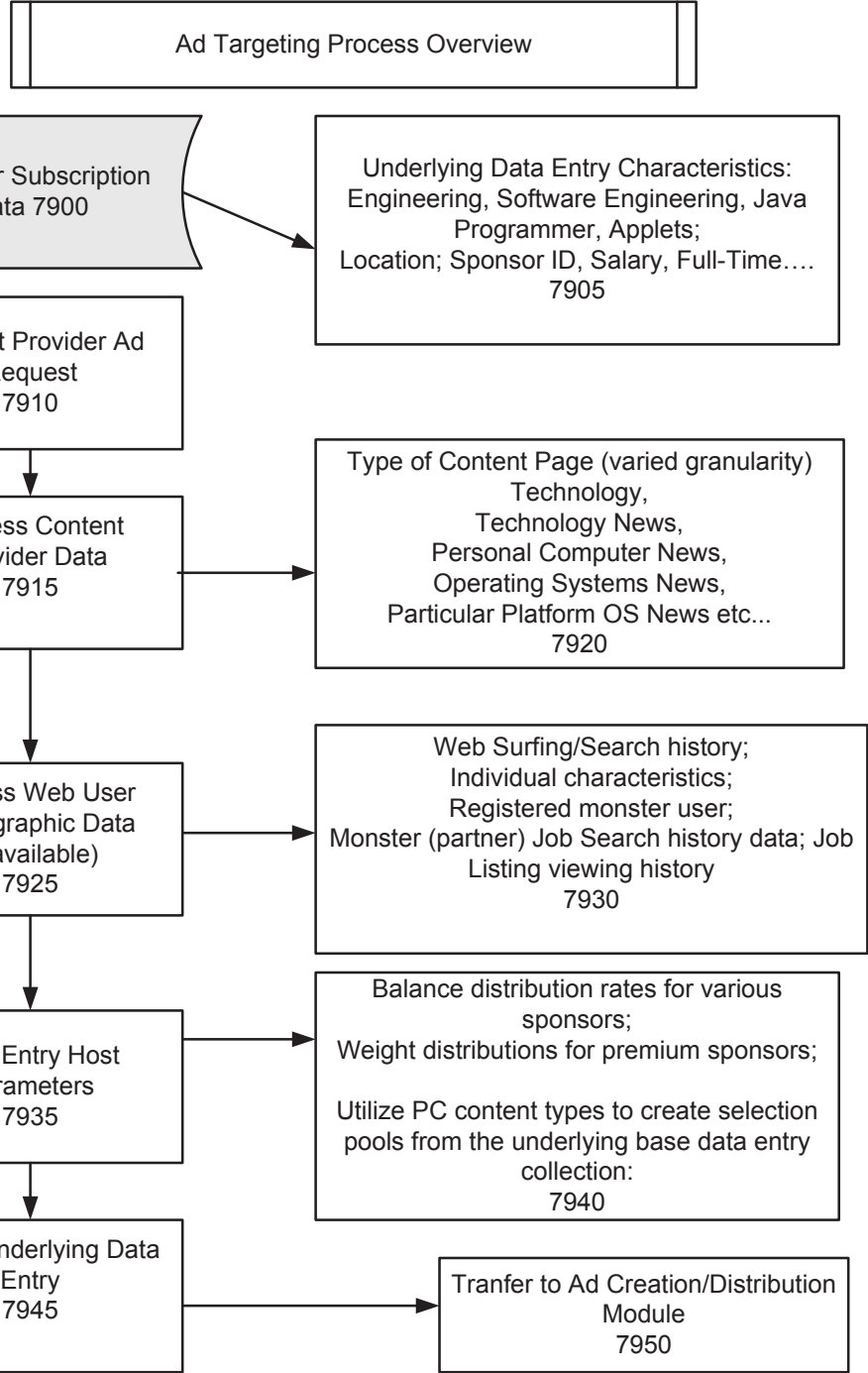
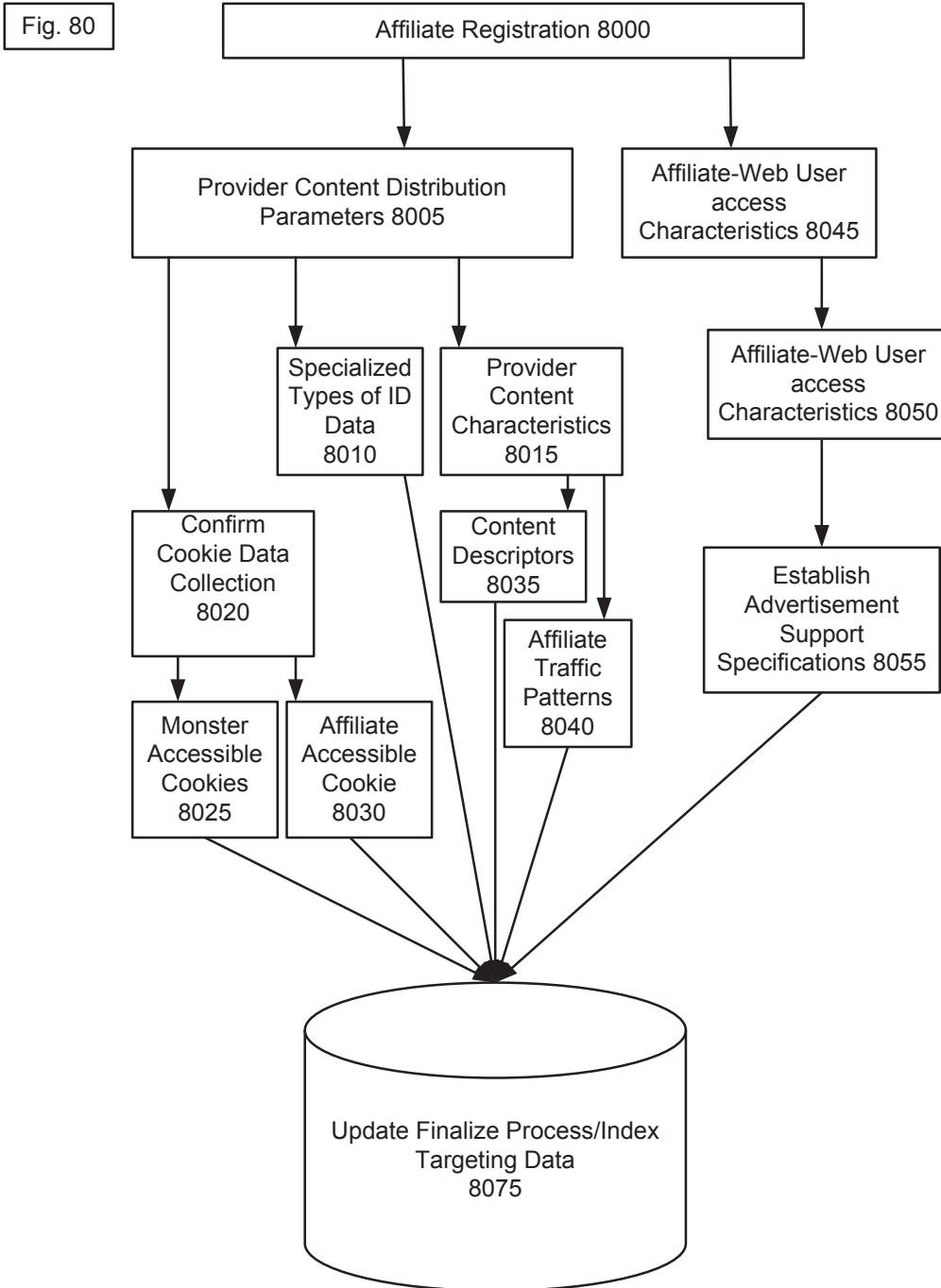
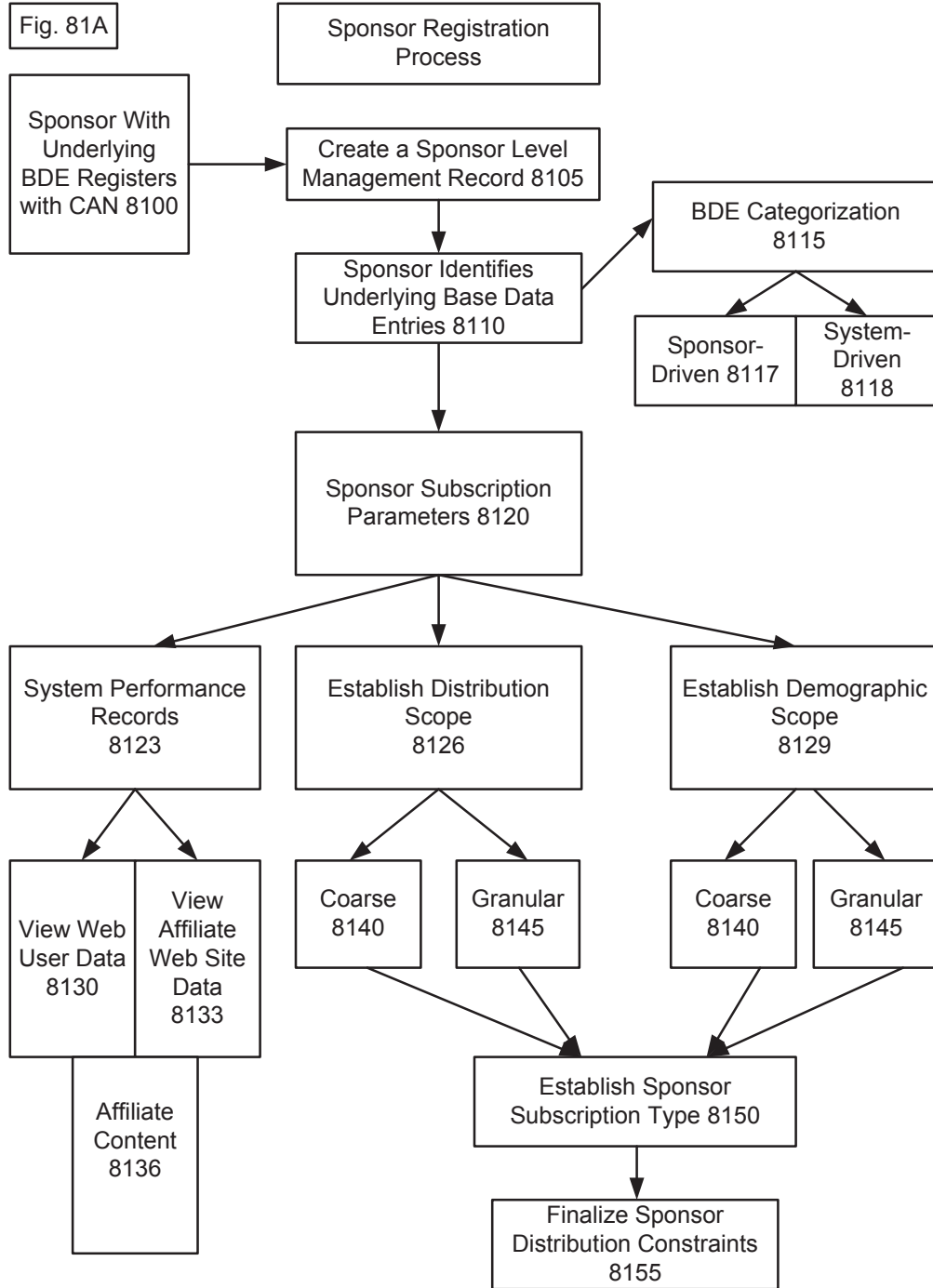


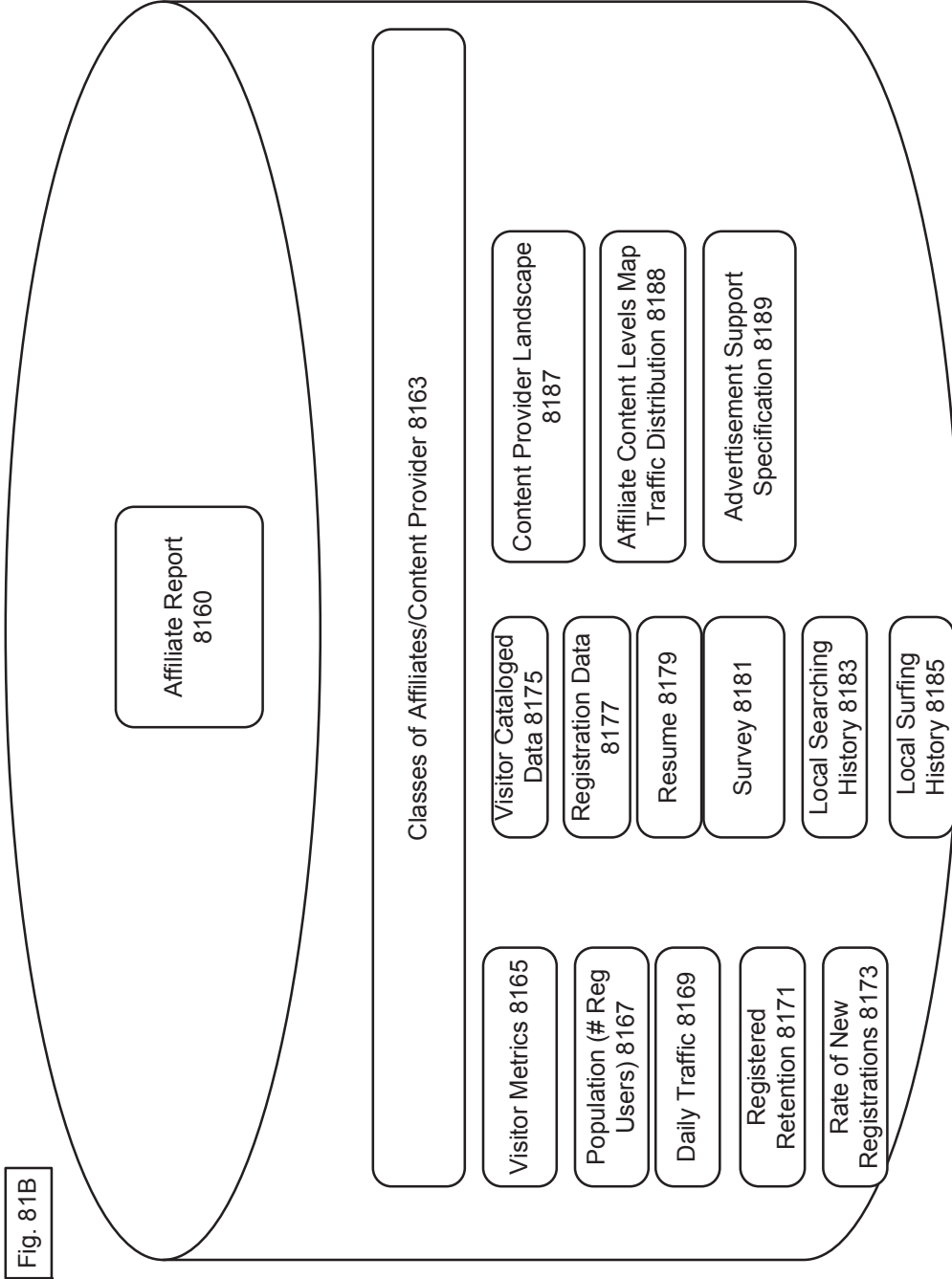
Fig. 78

Fig. 79









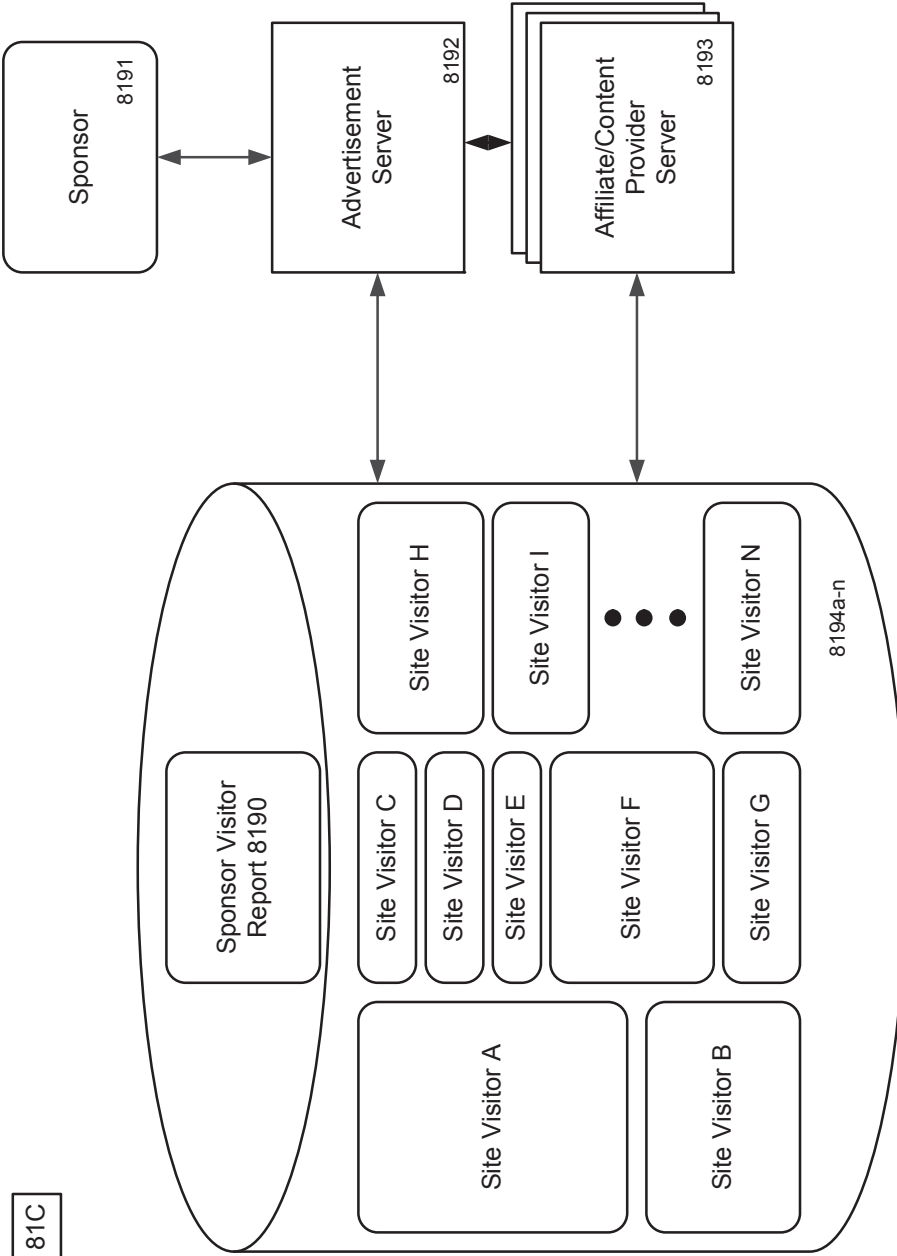
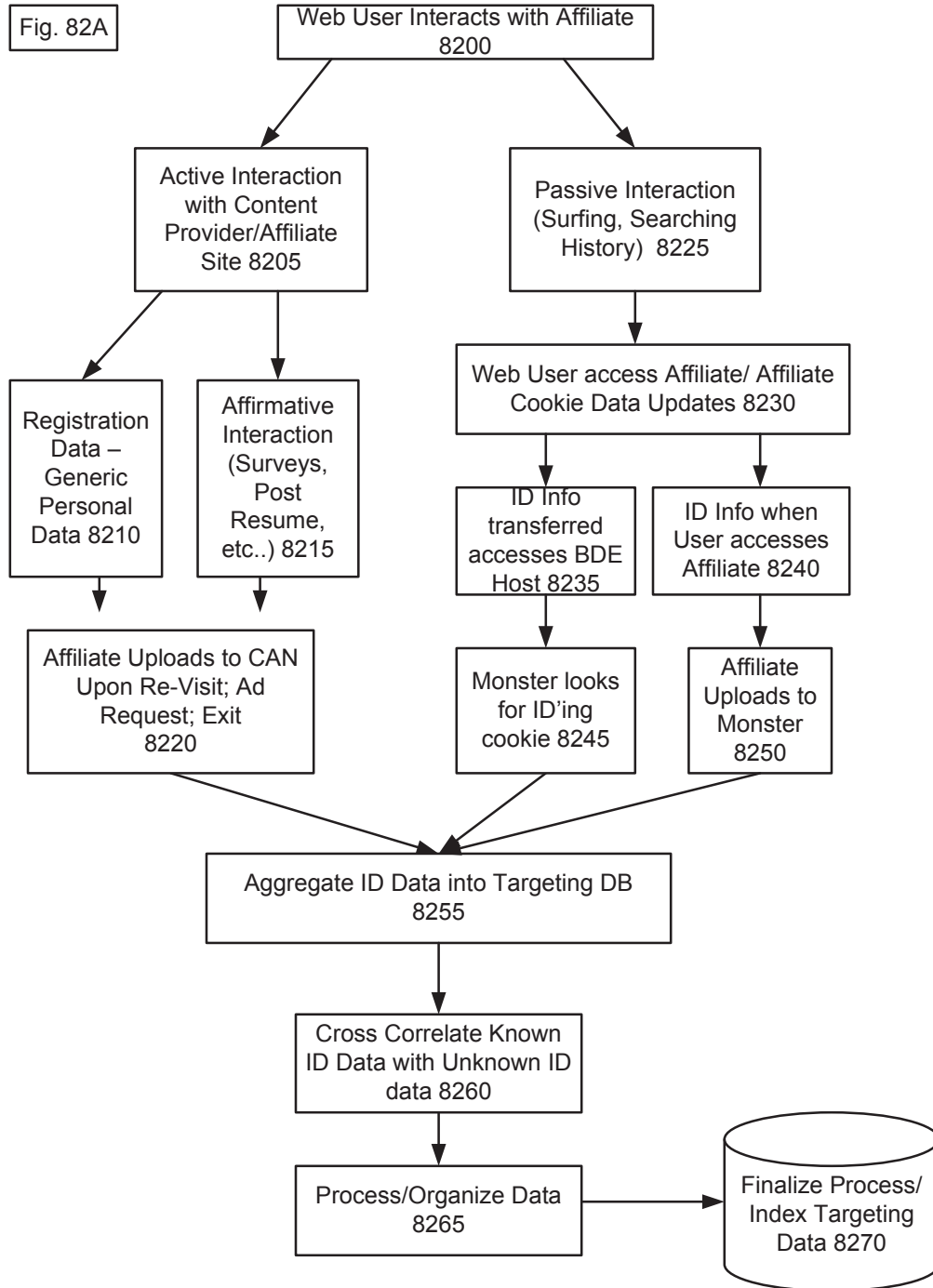
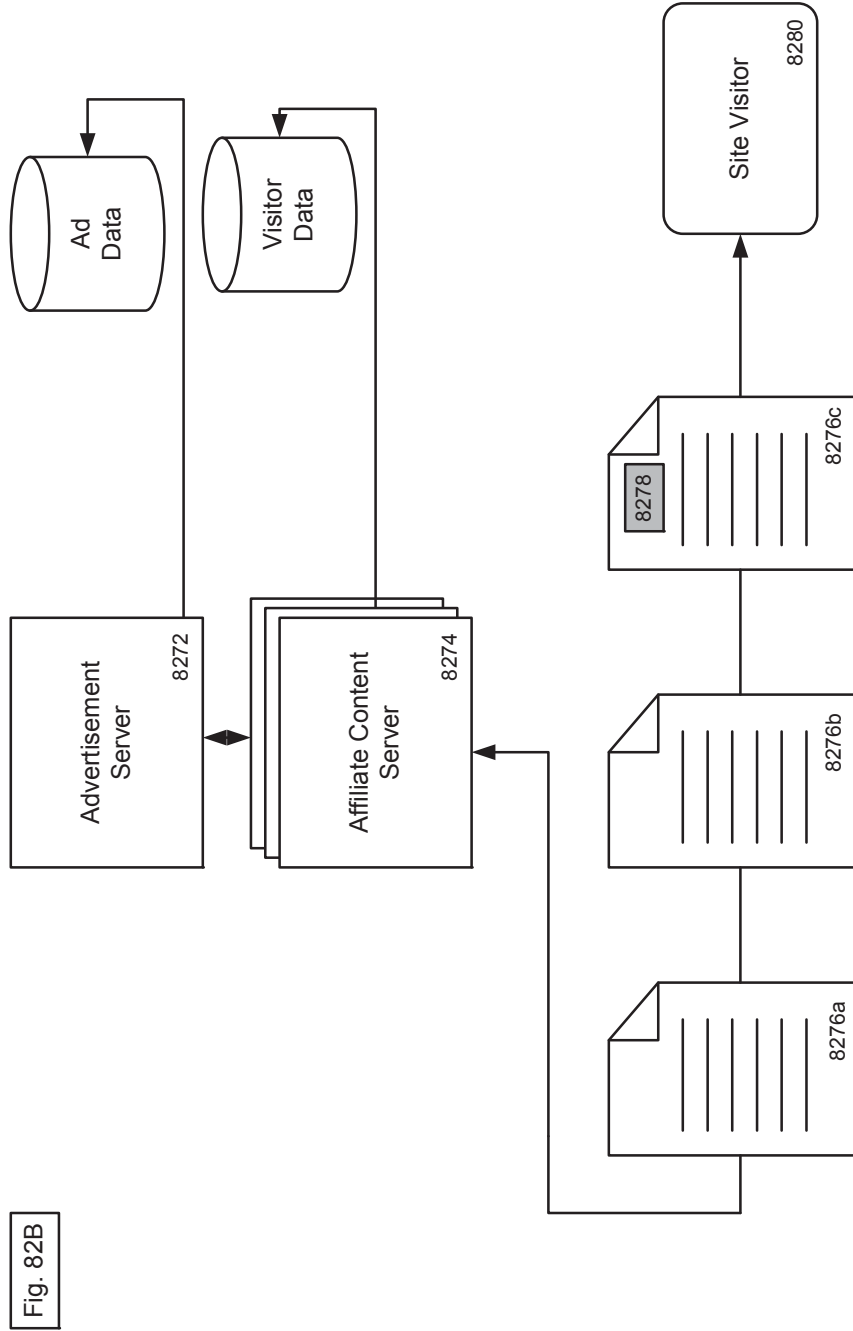
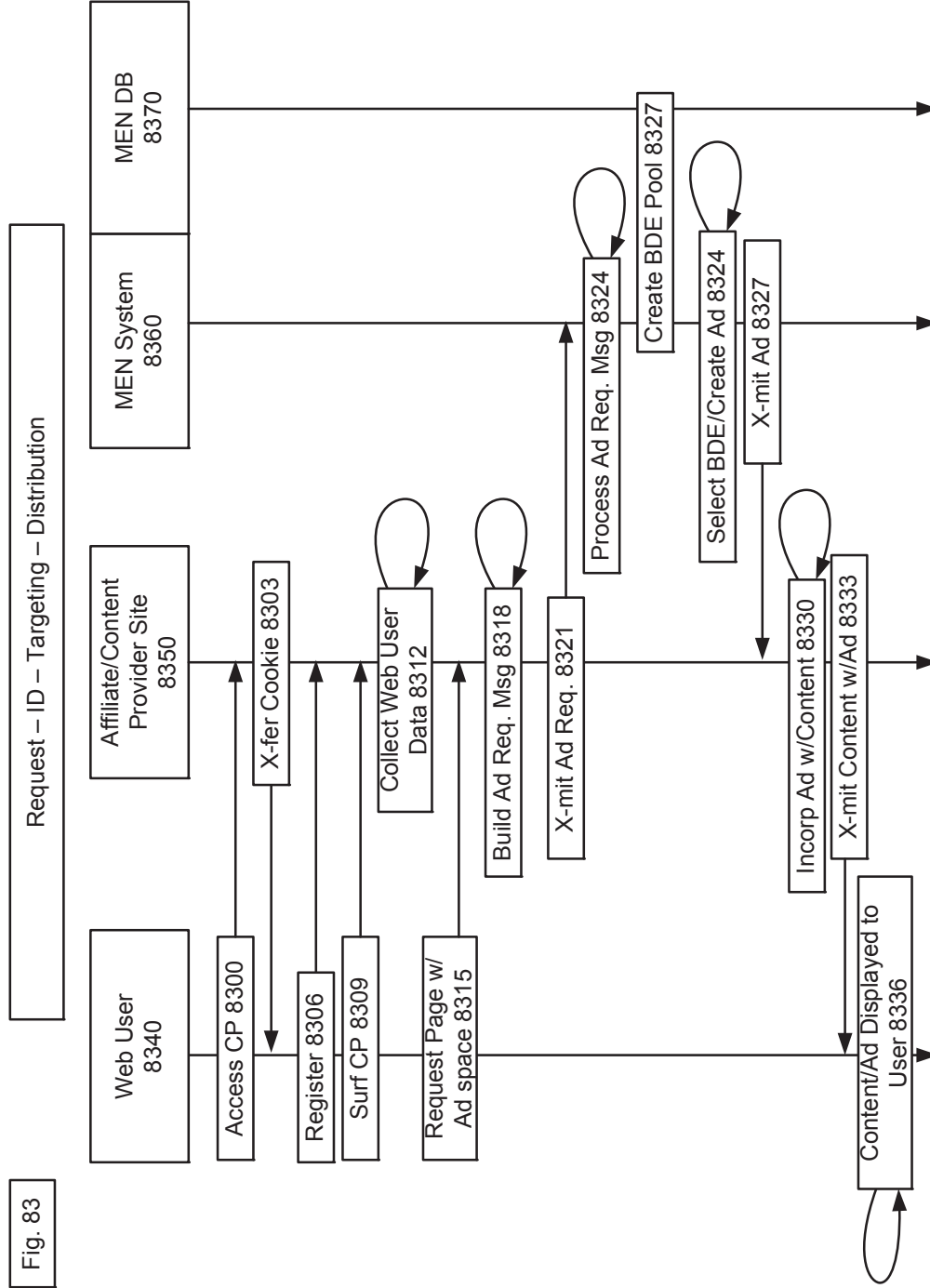
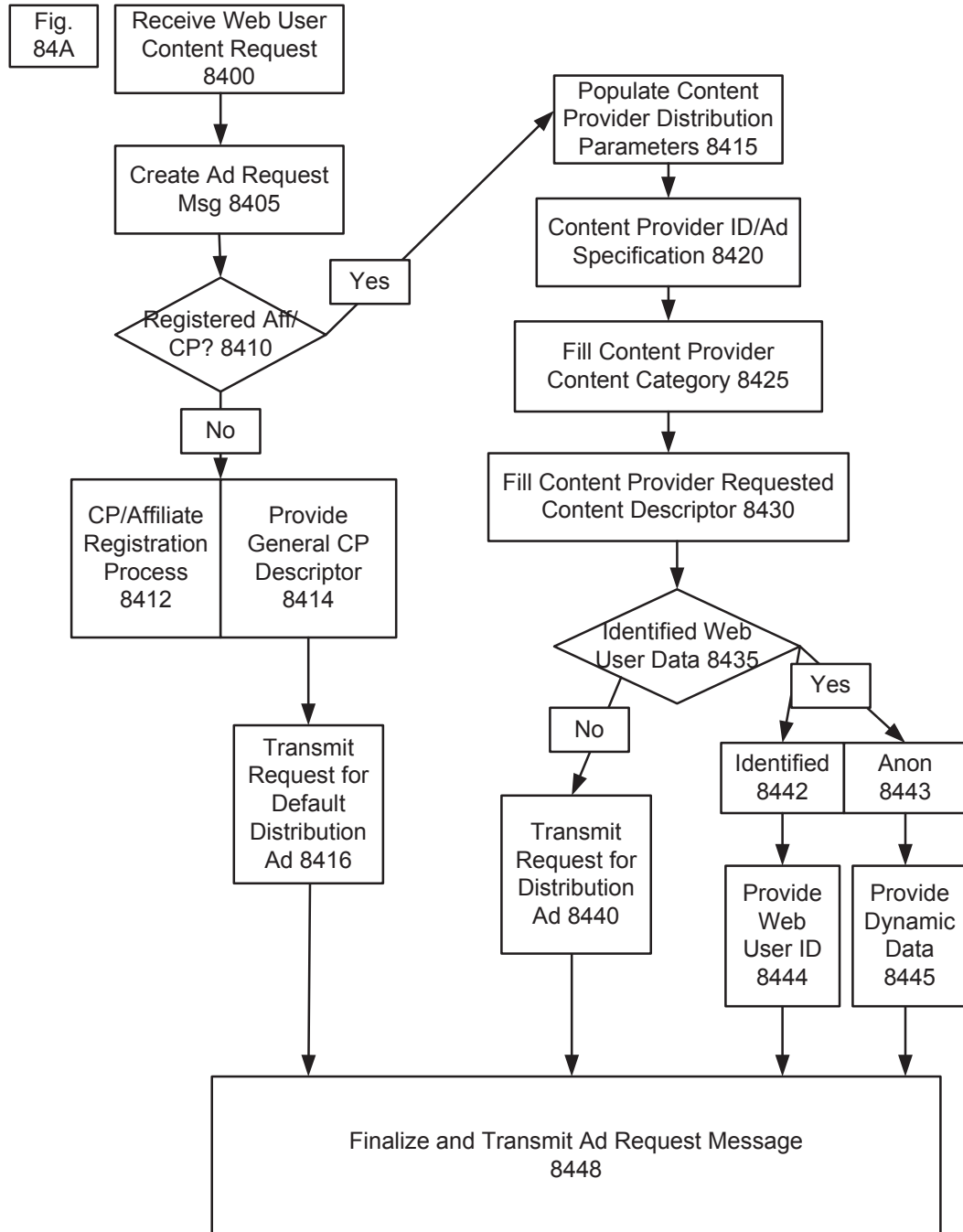


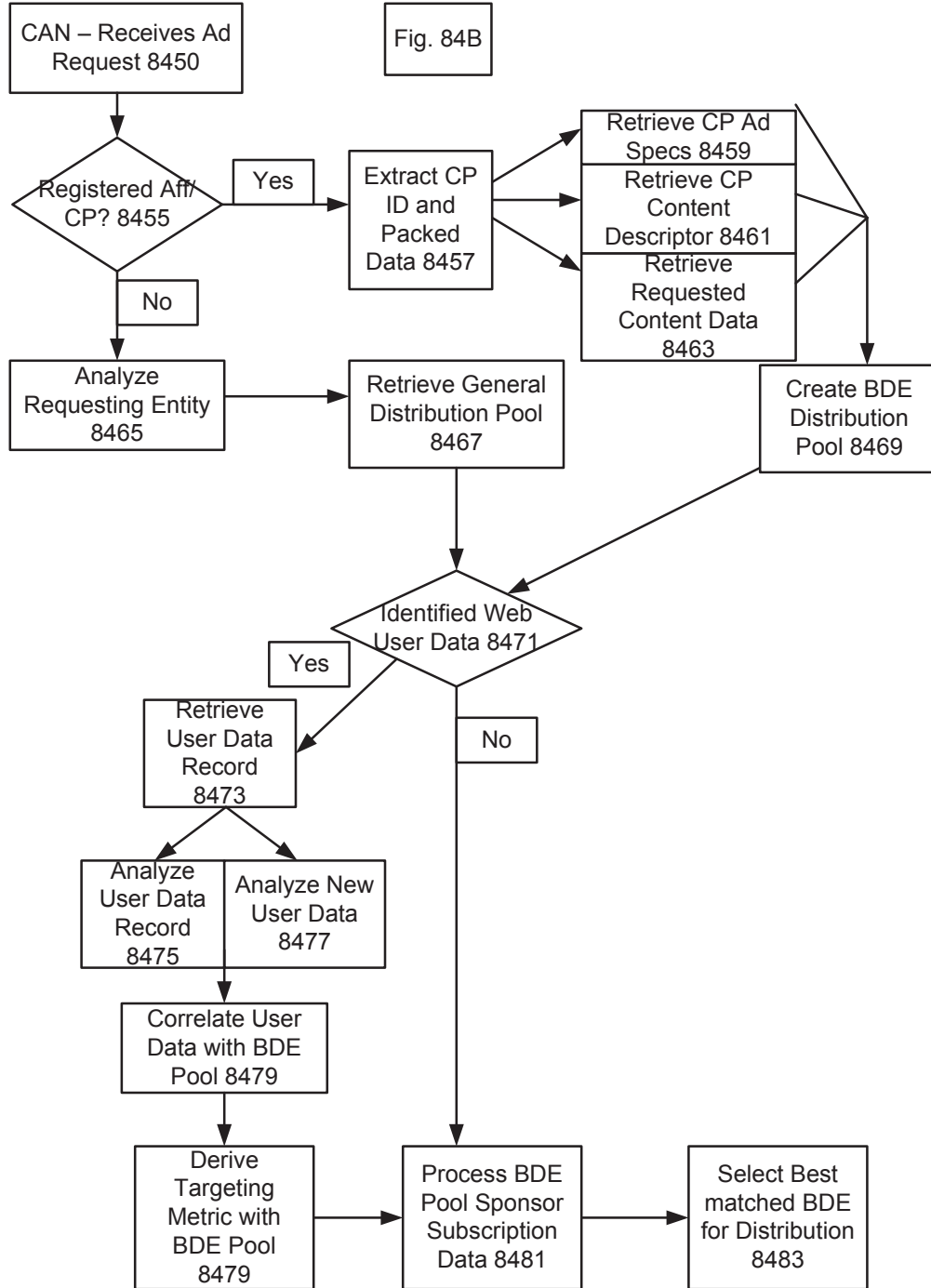
Fig. 81C











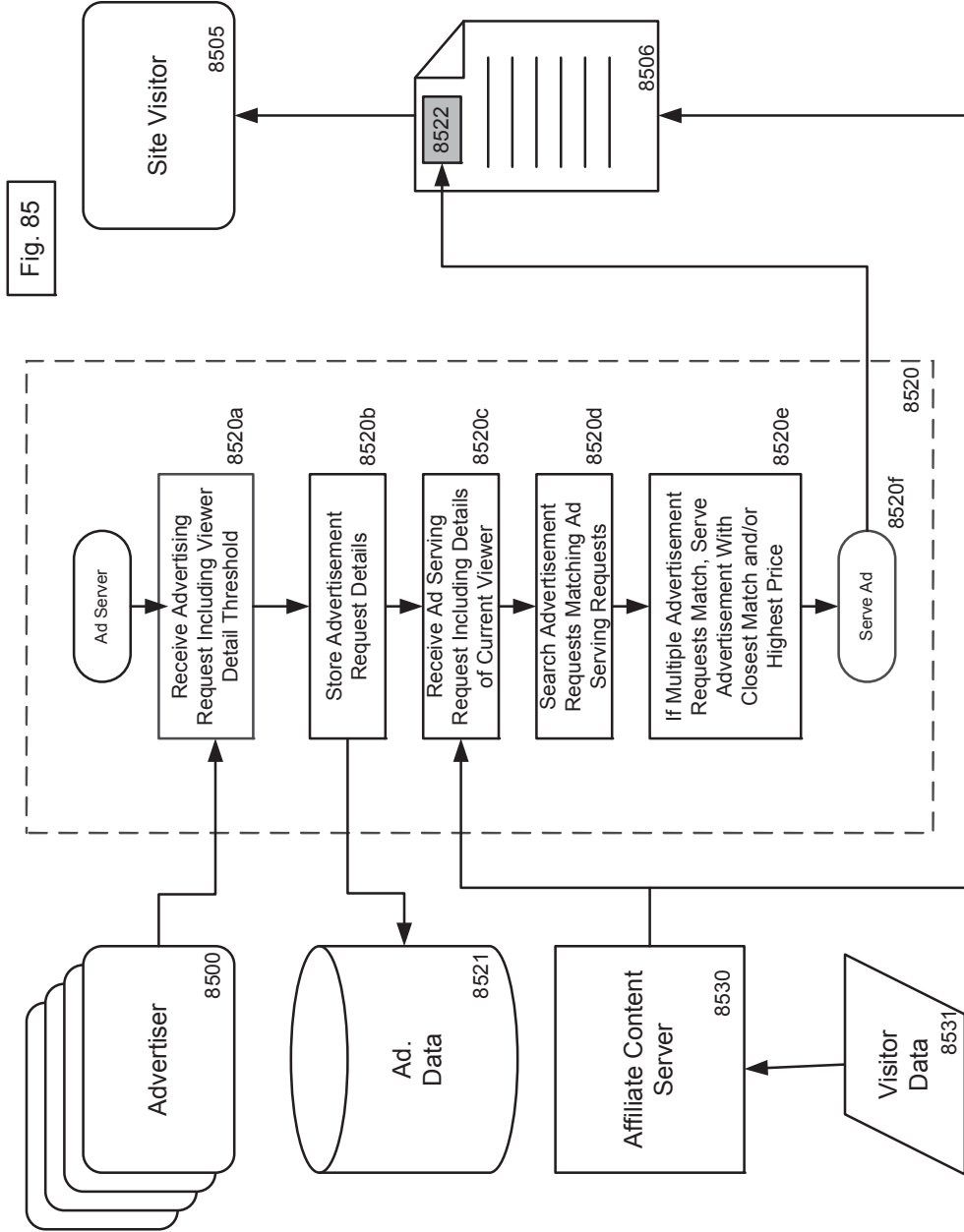


Fig. 85

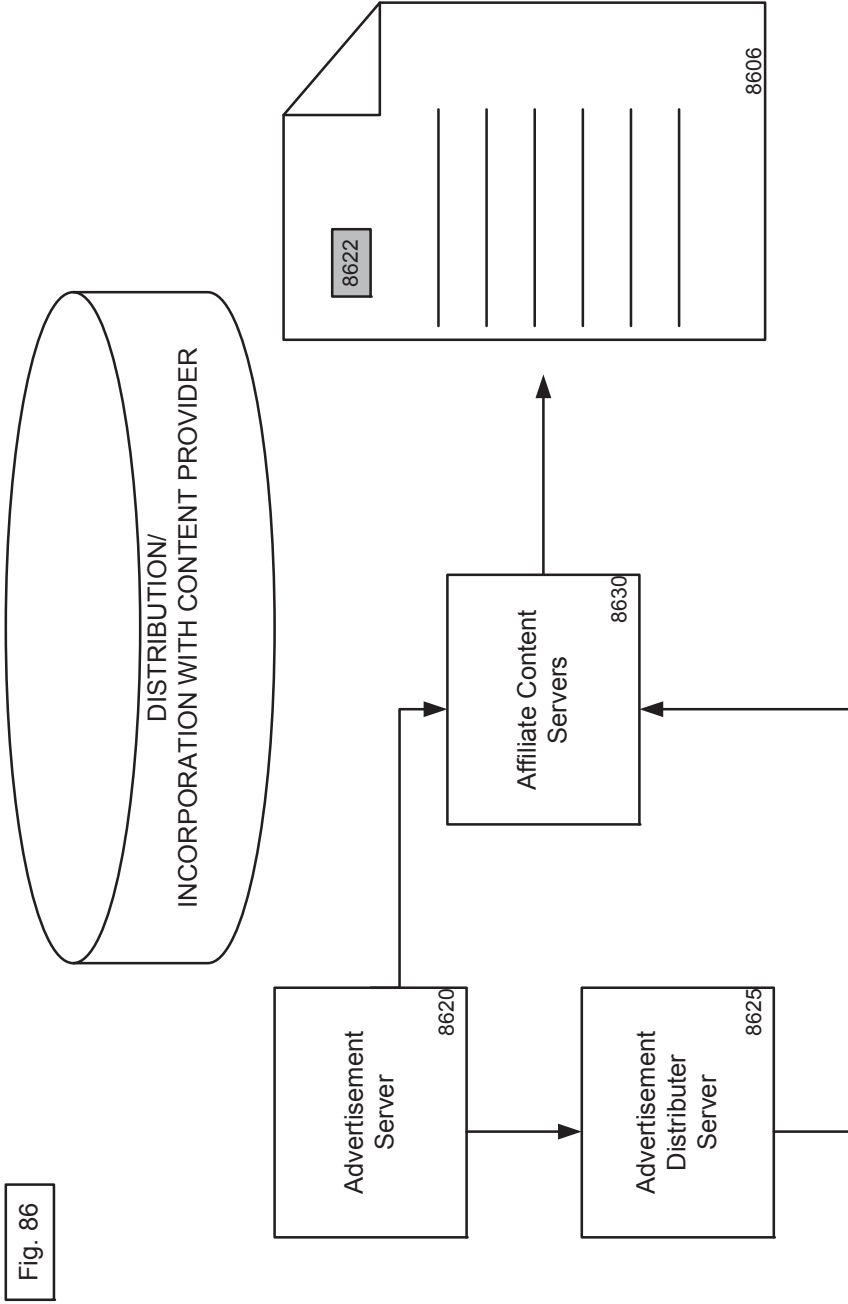
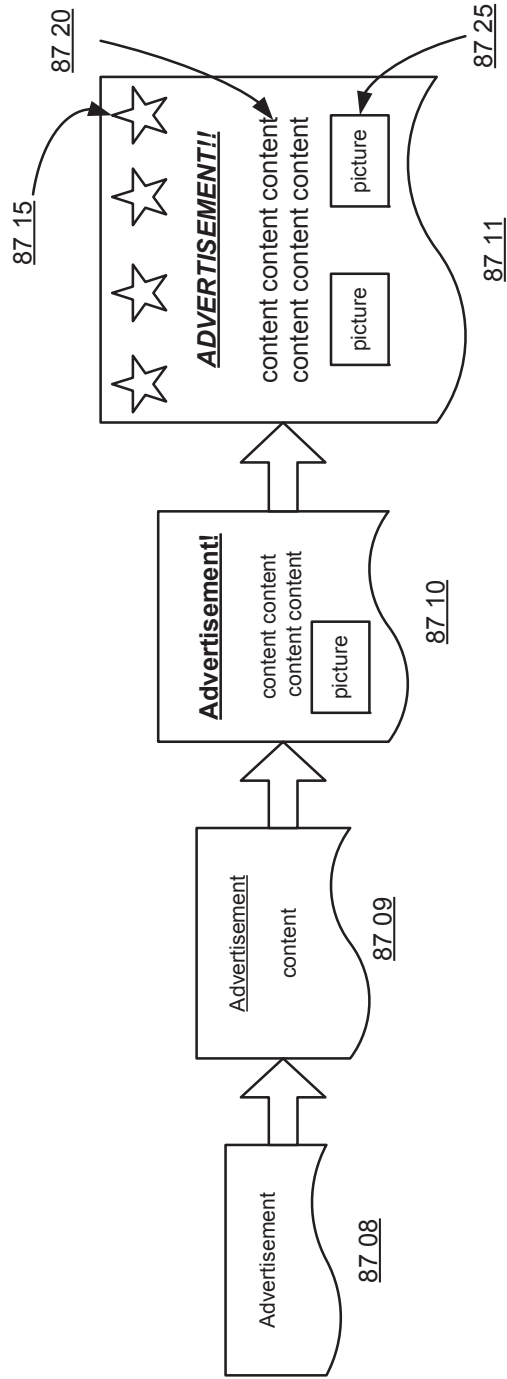


FIGURE 87



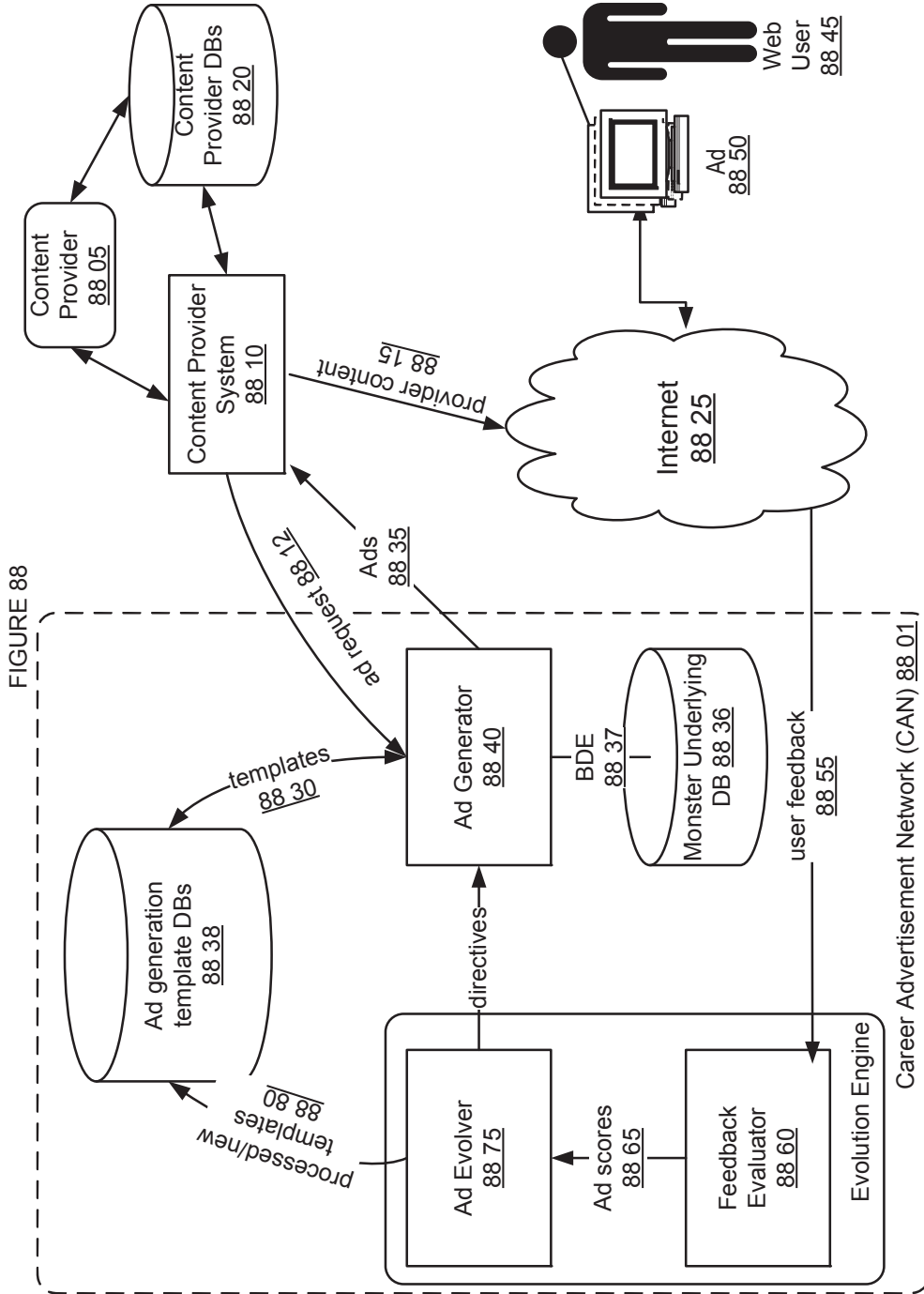


FIGURE 89

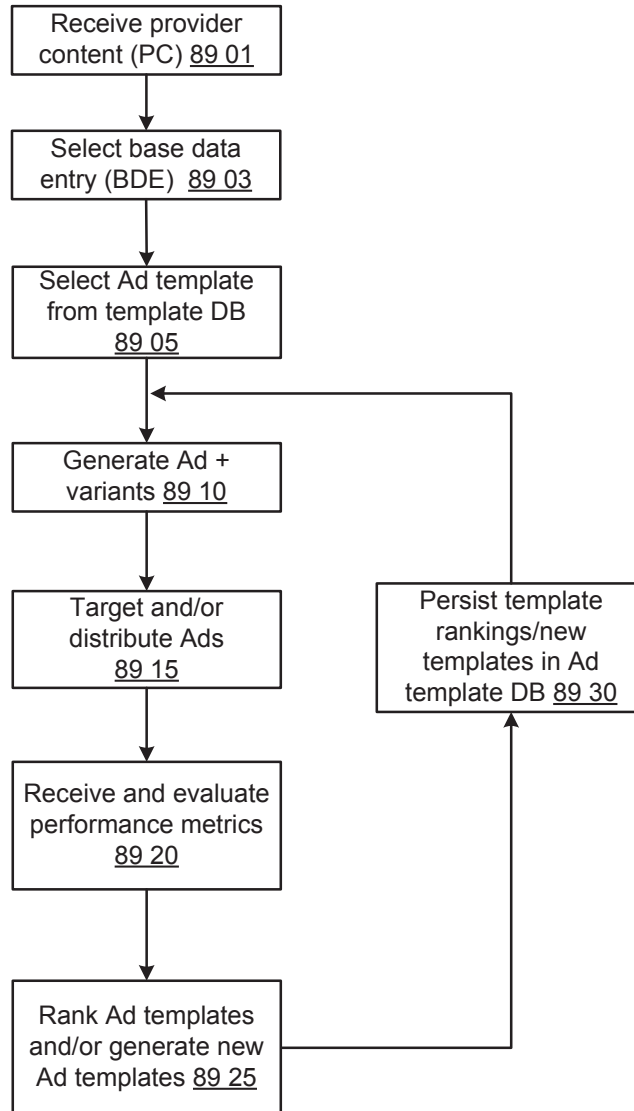
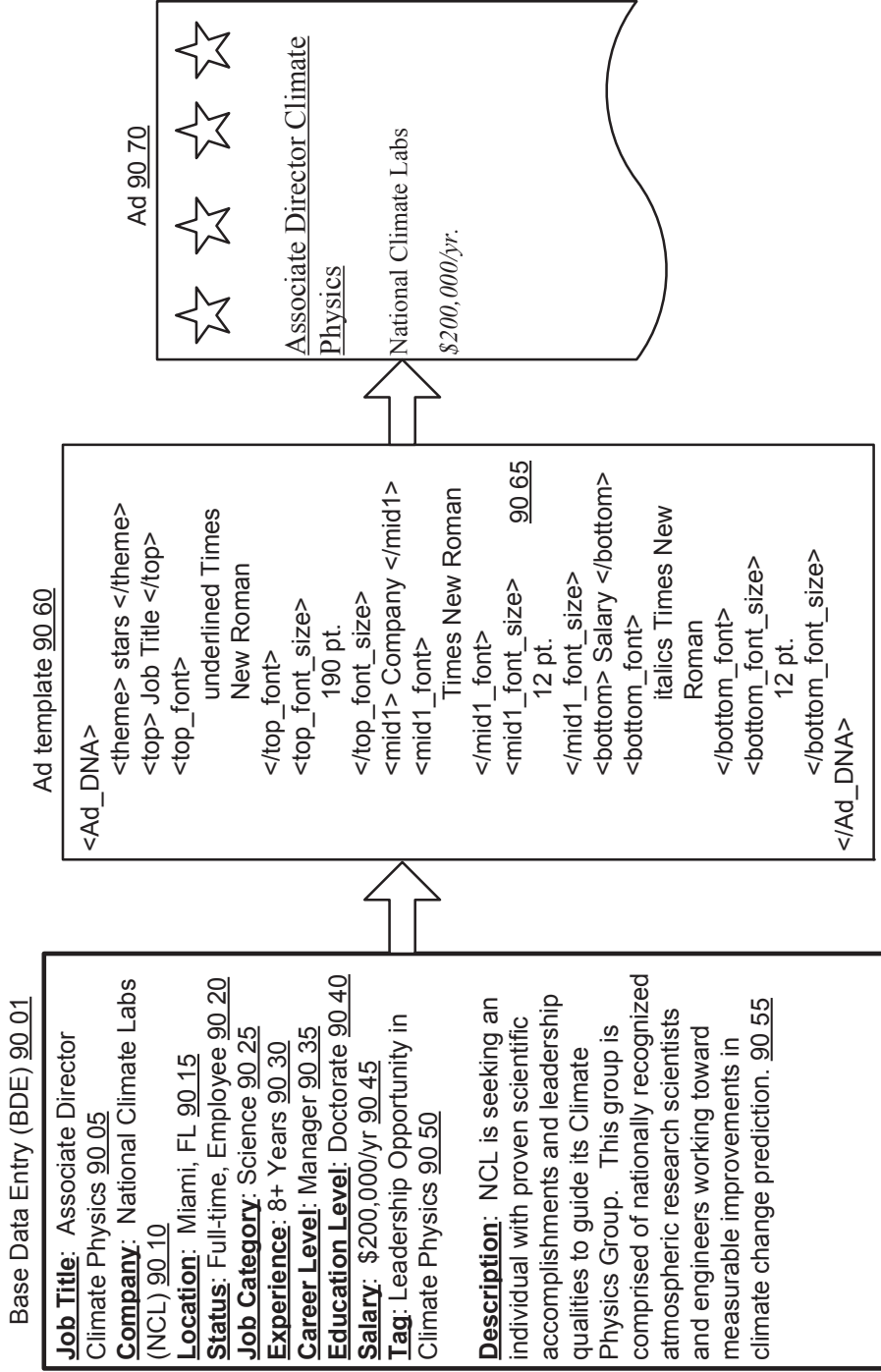


FIGURE 90



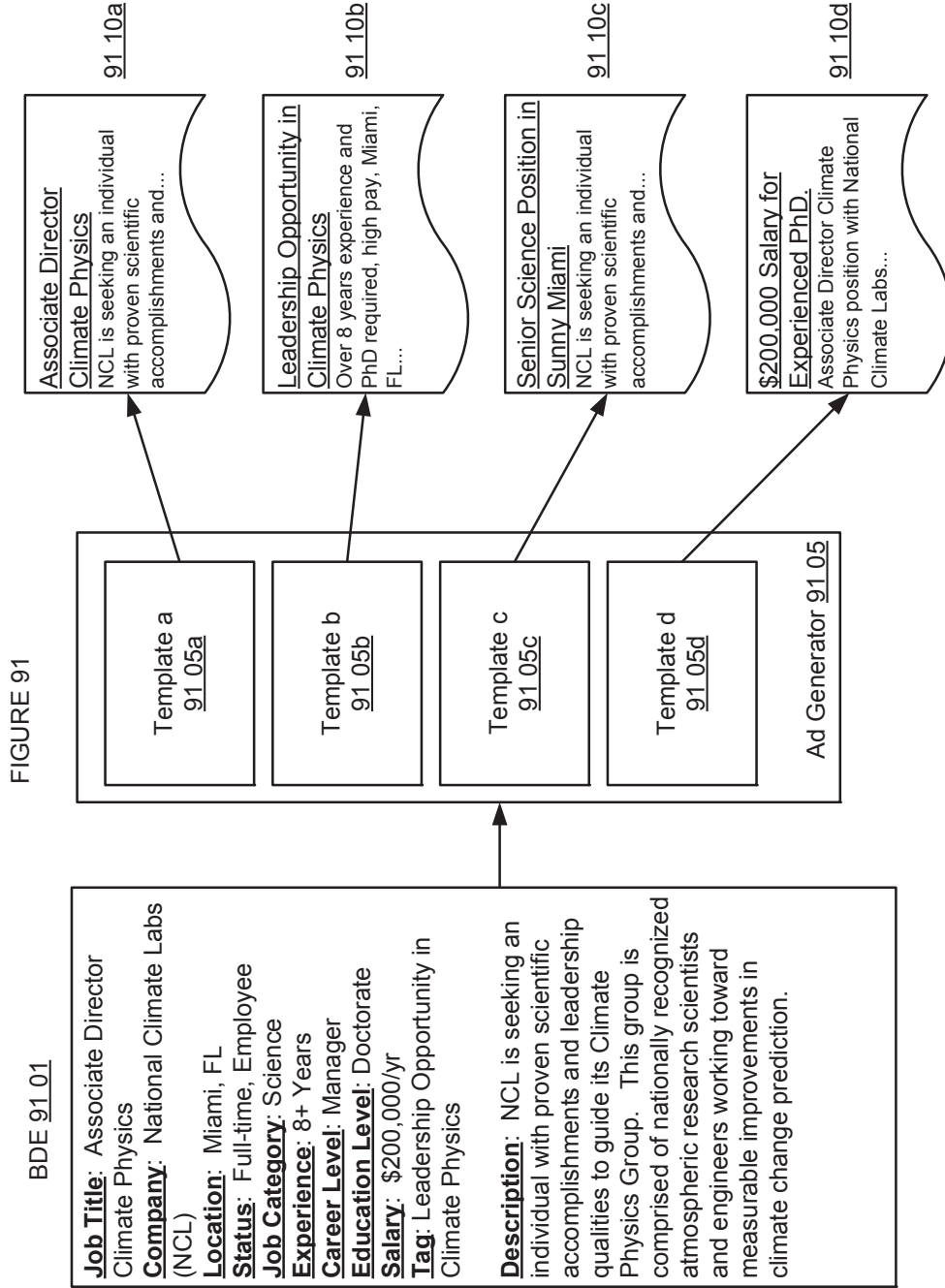


FIGURE 92a

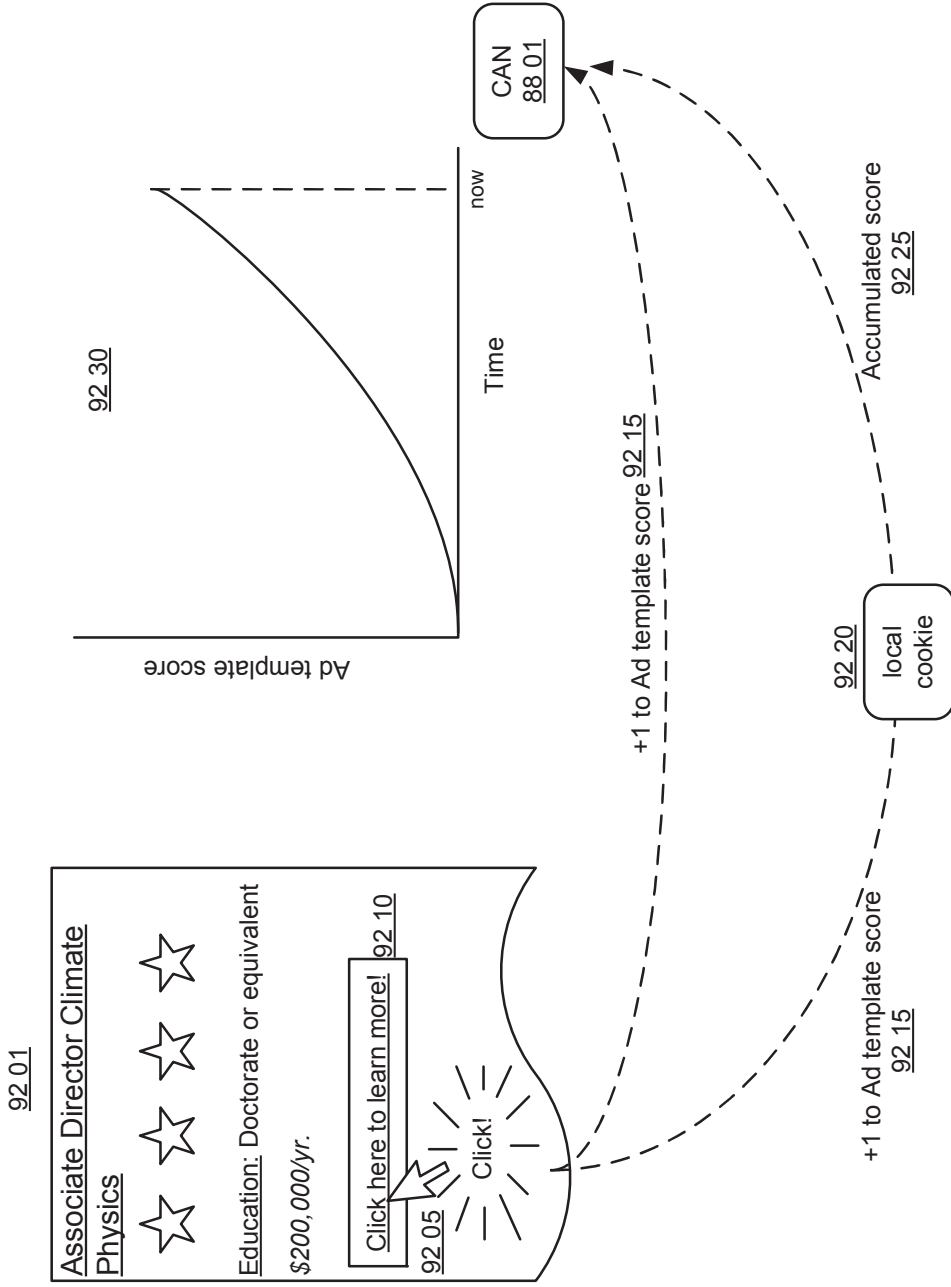


FIGURE 92c

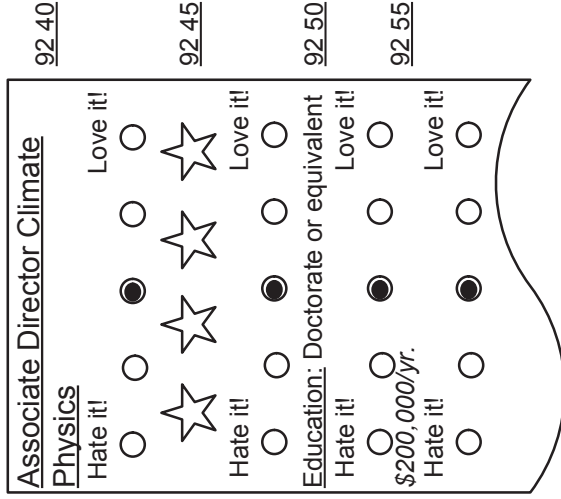


FIGURE 92b

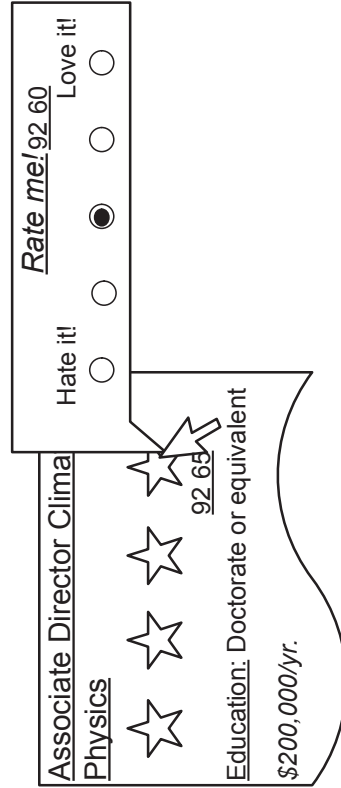
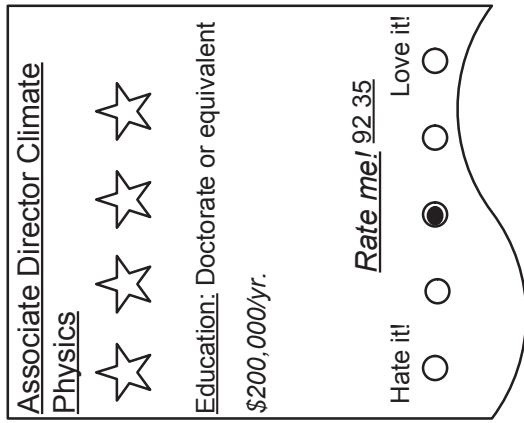


FIGURE 92d

FIGURE 93a

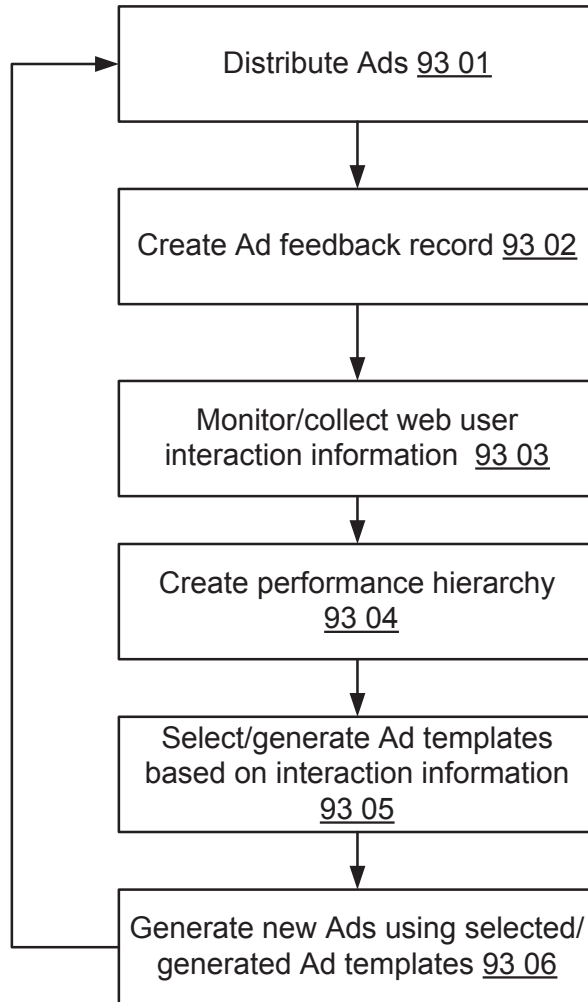


FIGURE 93b

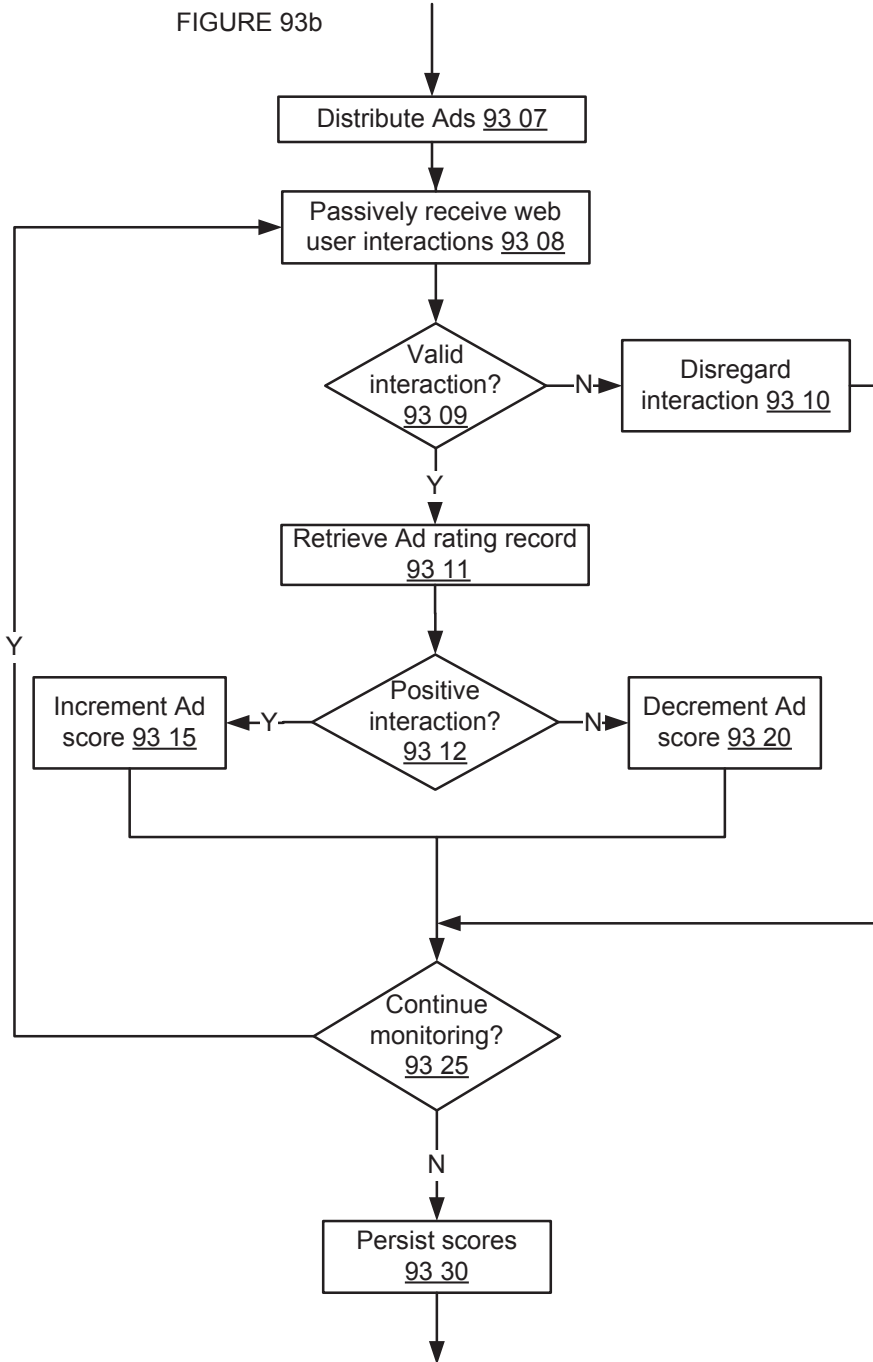


FIGURE 93c

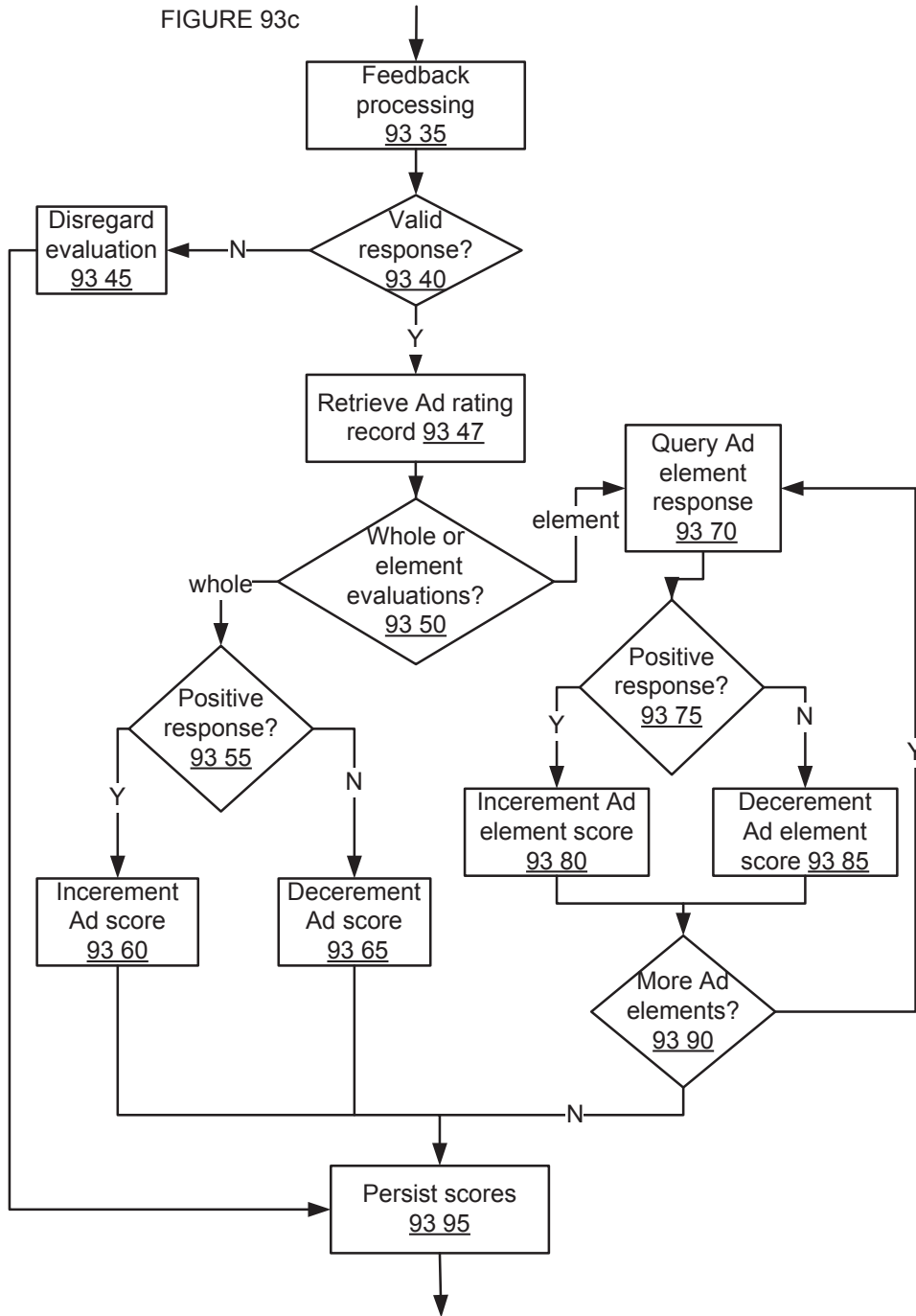


FIGURE 94

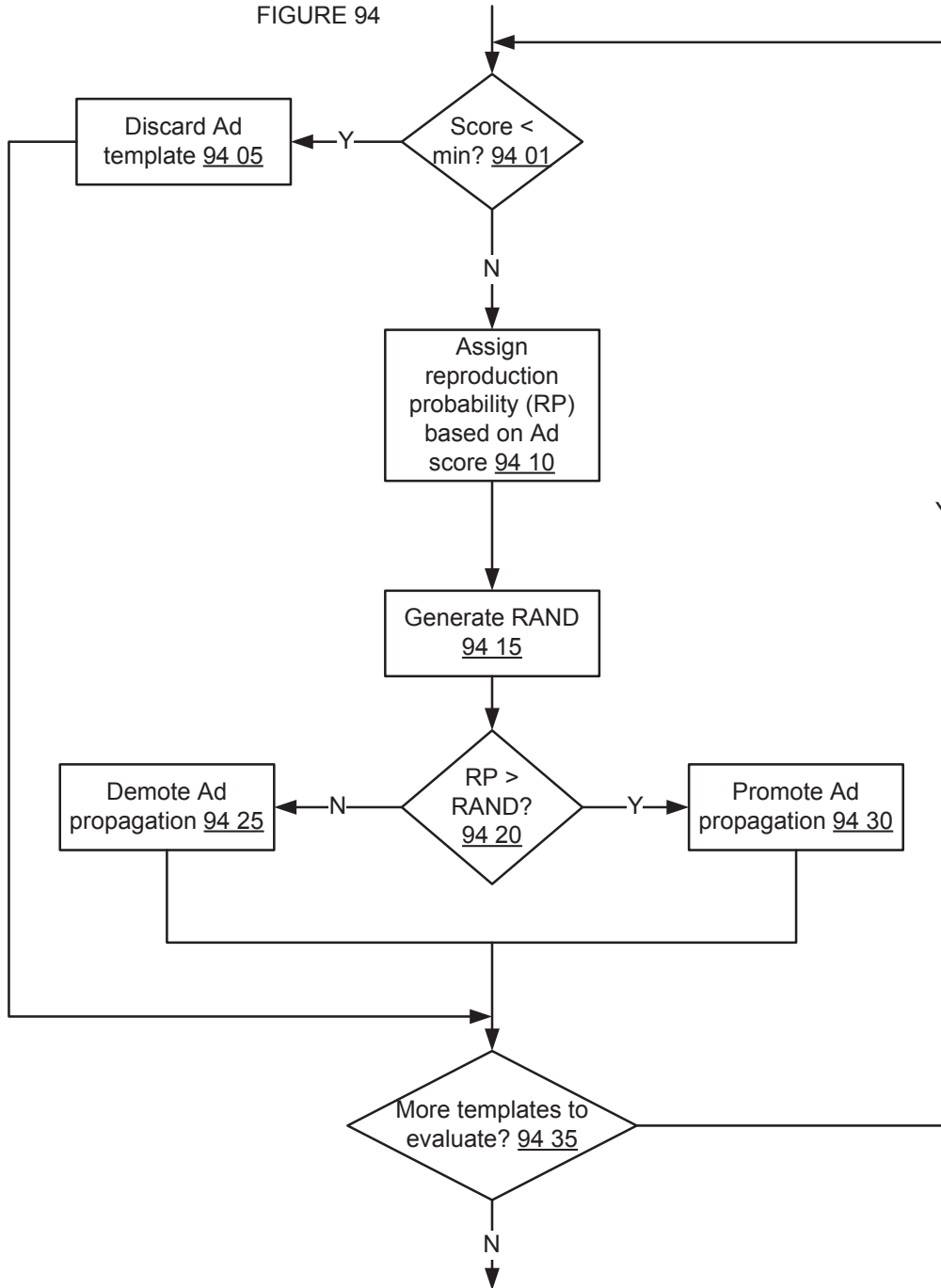


FIGURE 95
1st generation 95.01

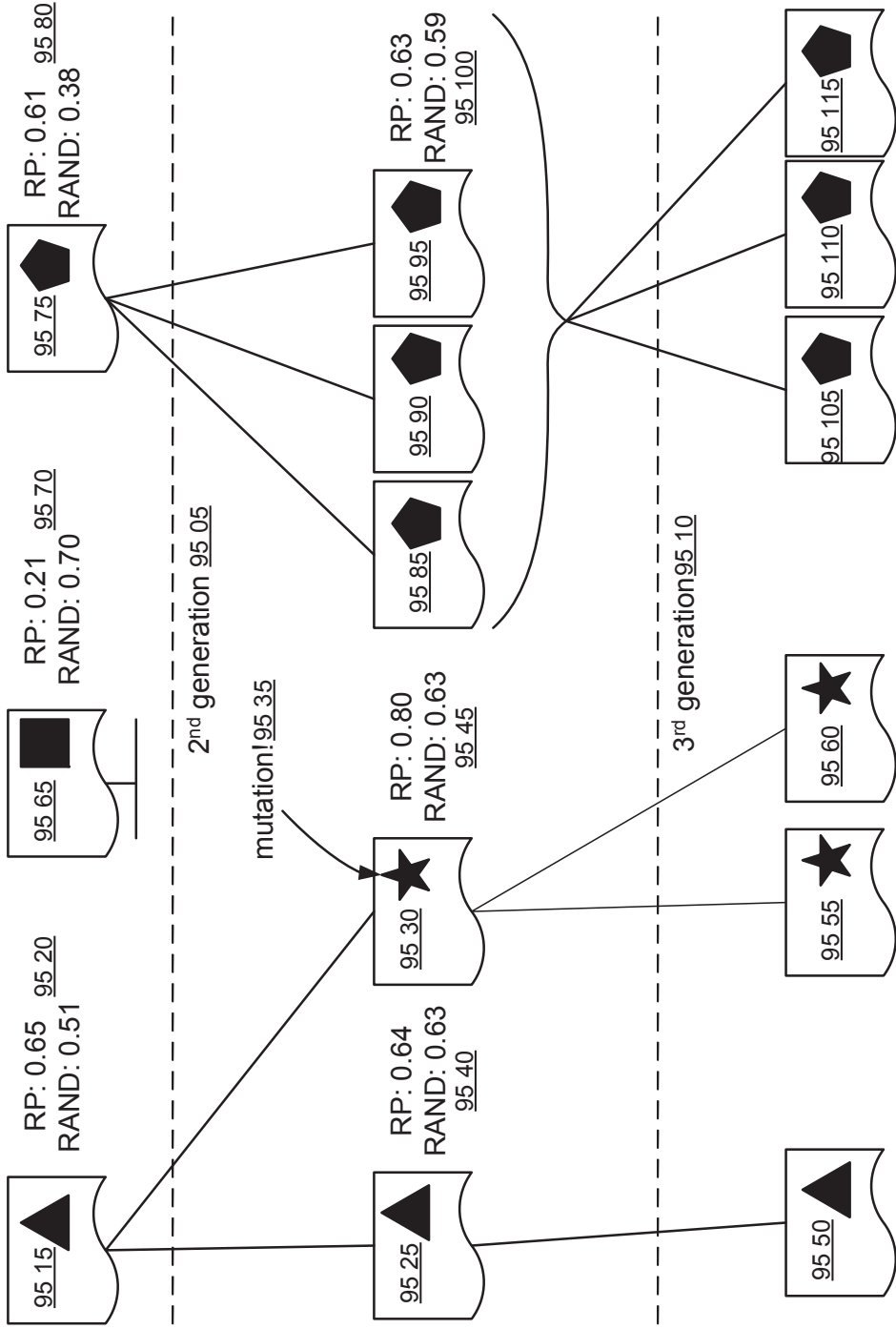


FIGURE 96

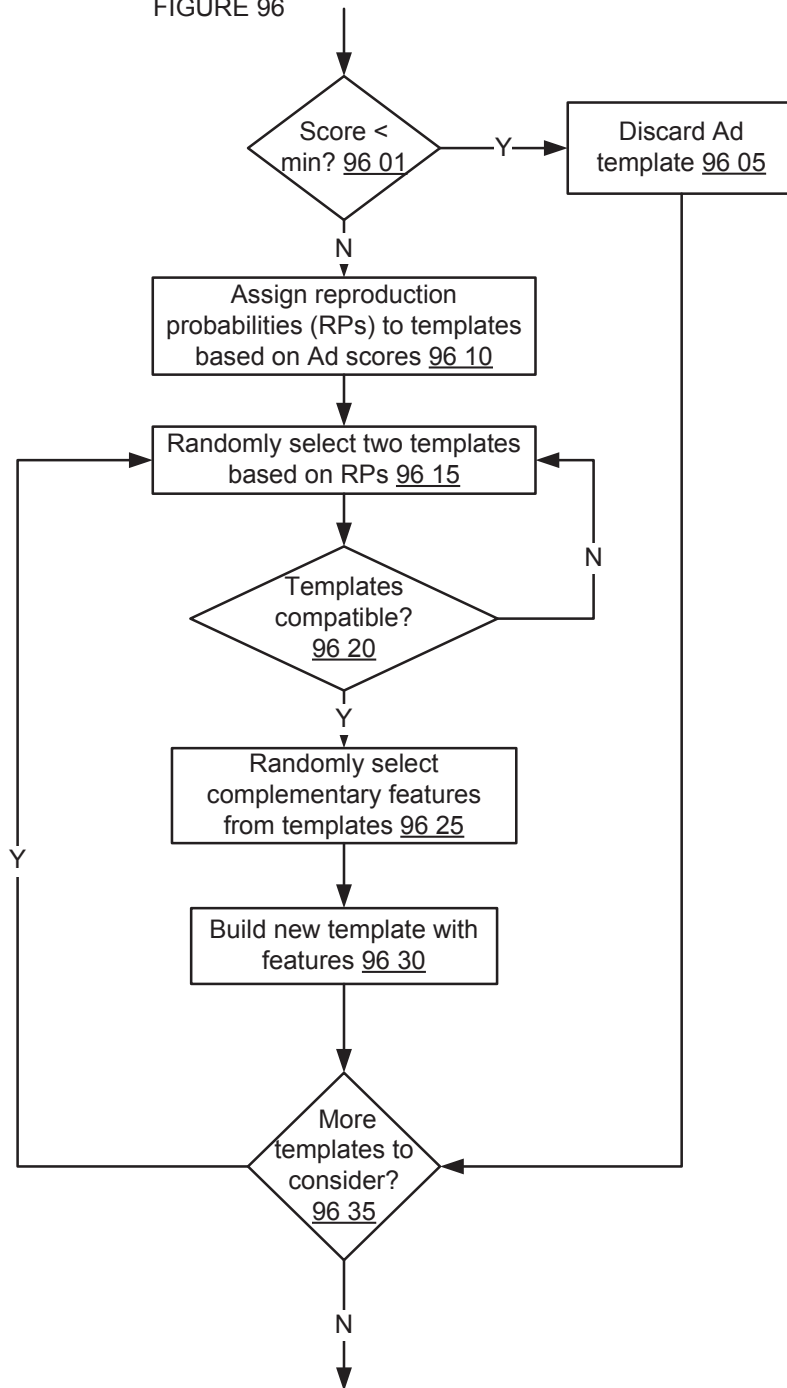


FIGURE 97

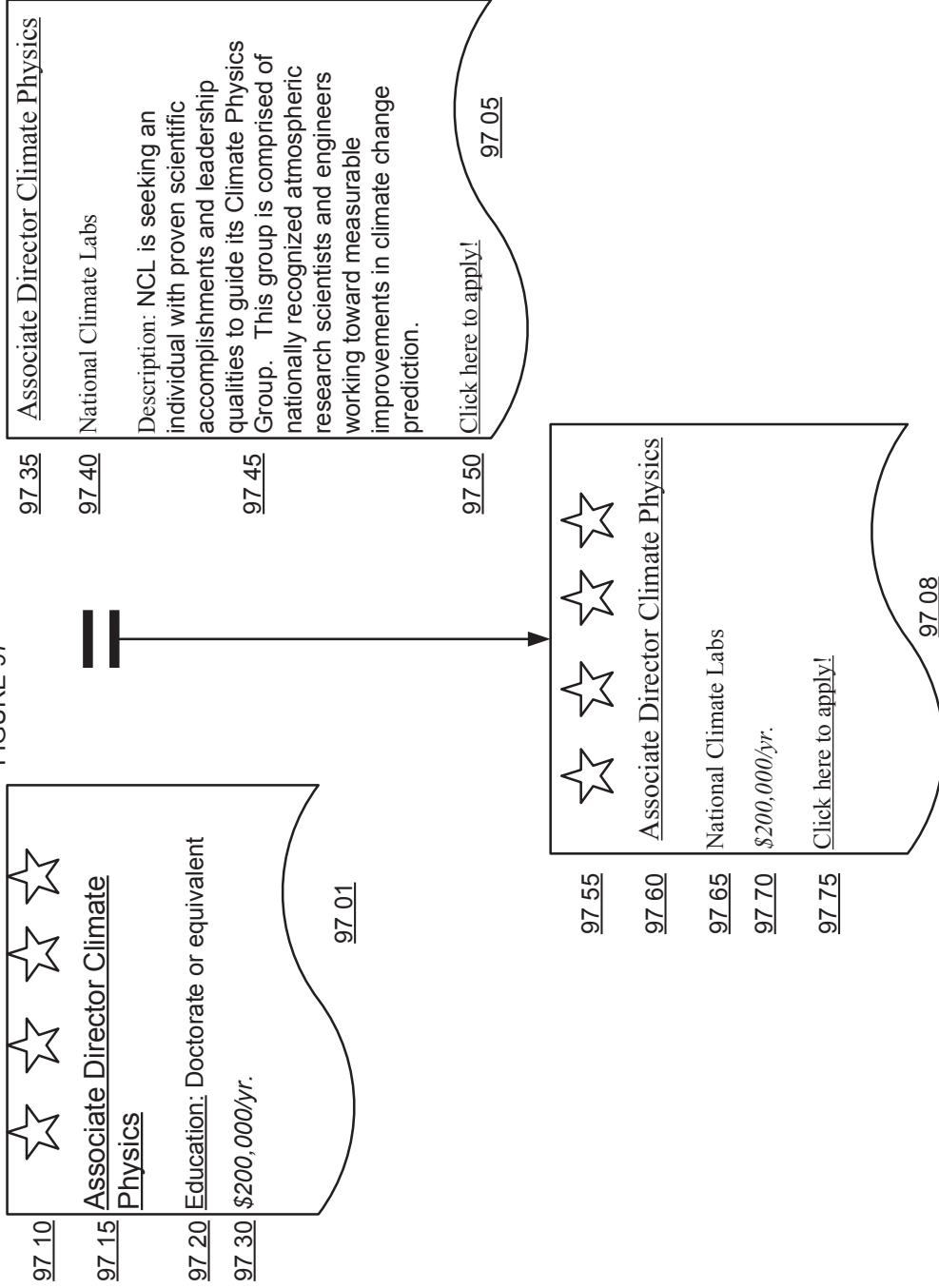


FIGURE 98

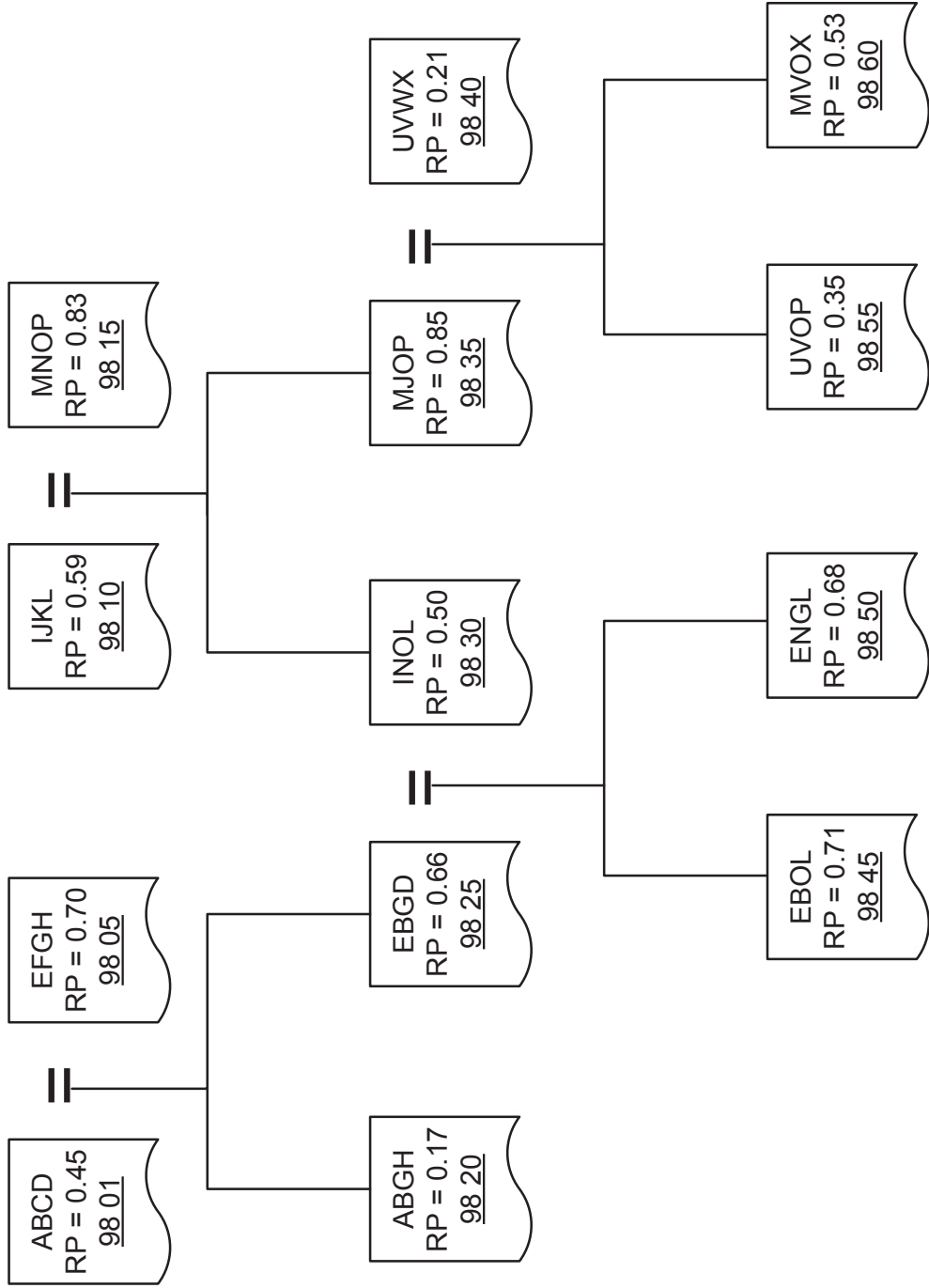
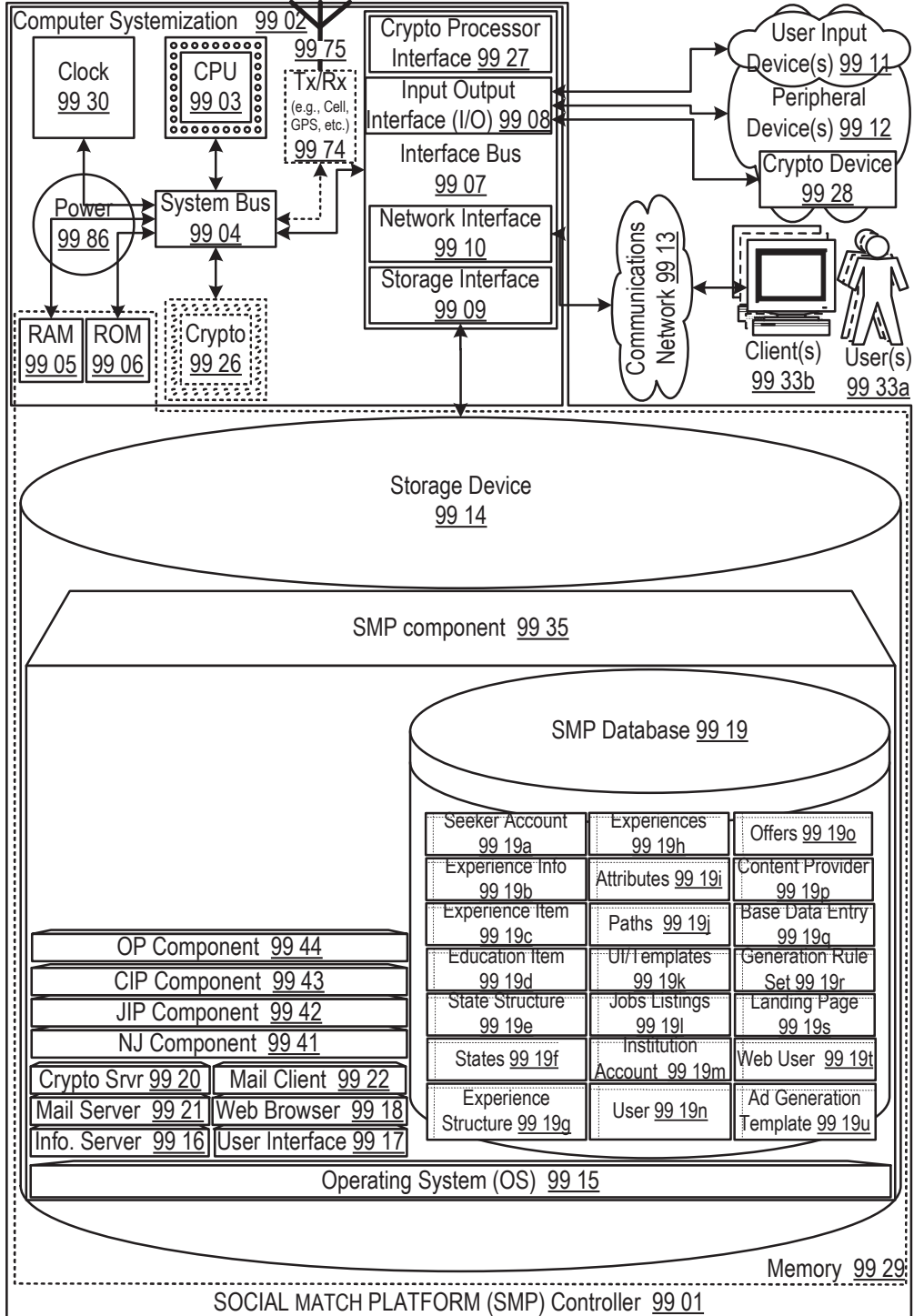


FIGURE 99



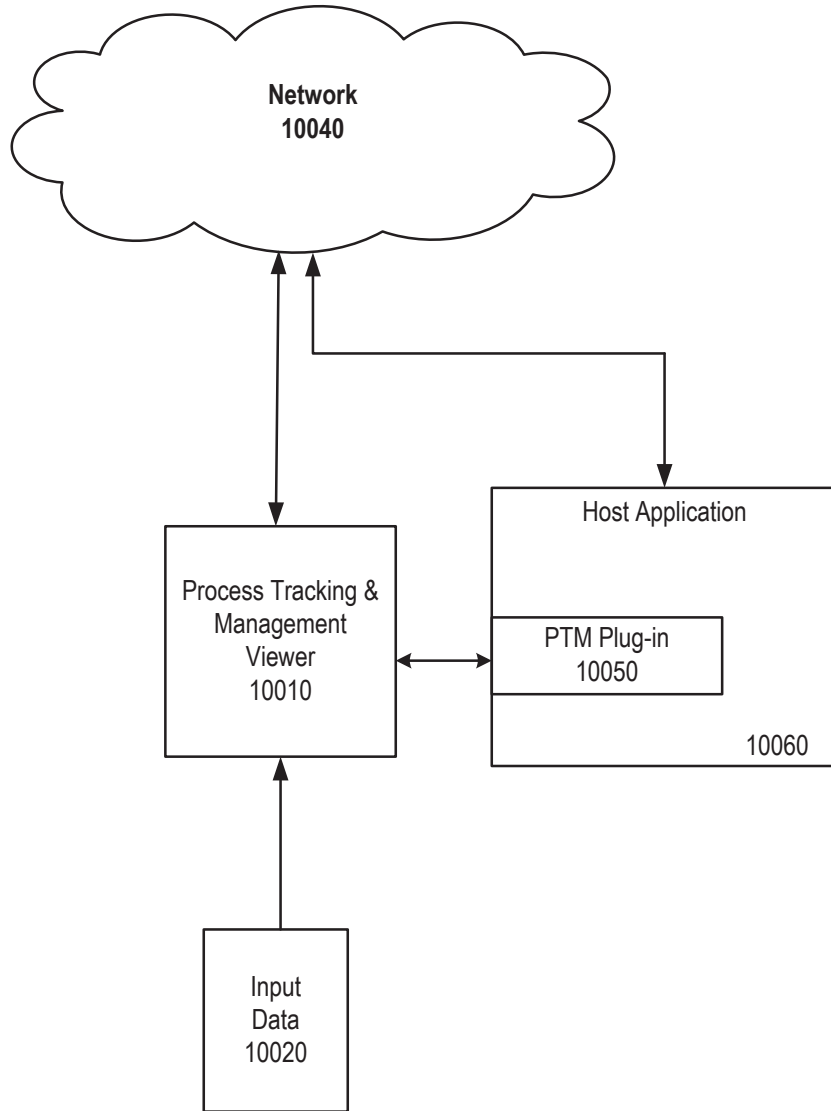


FIG. 100

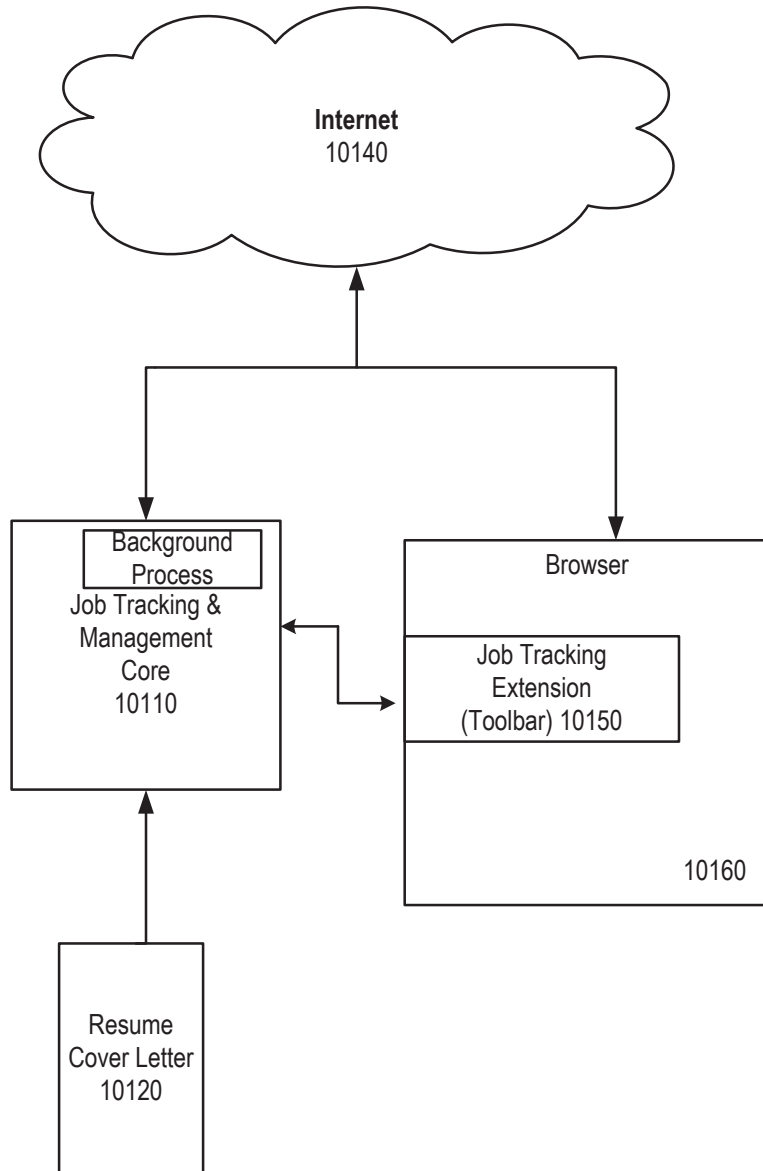


FIG. 101

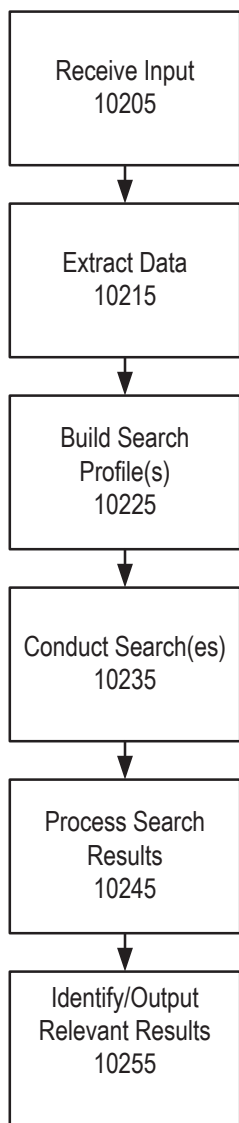


FIG. 102A

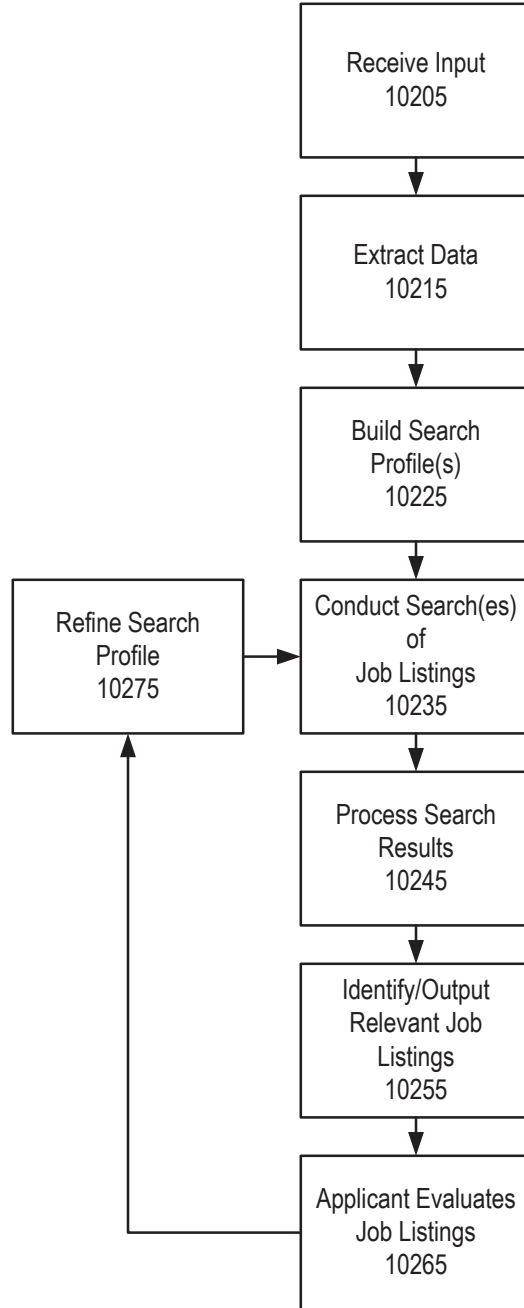


FIG. 102B

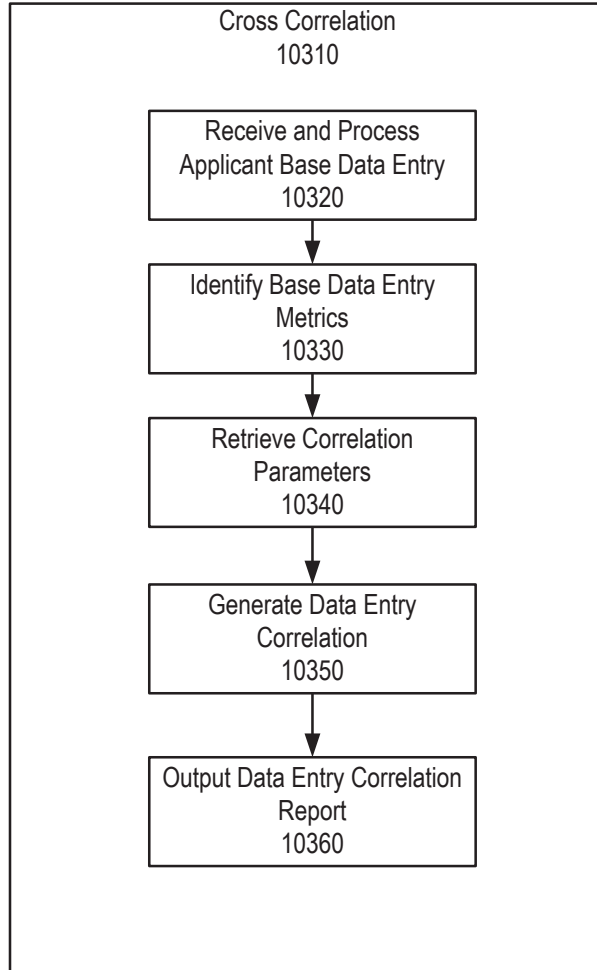


FIG. 103A

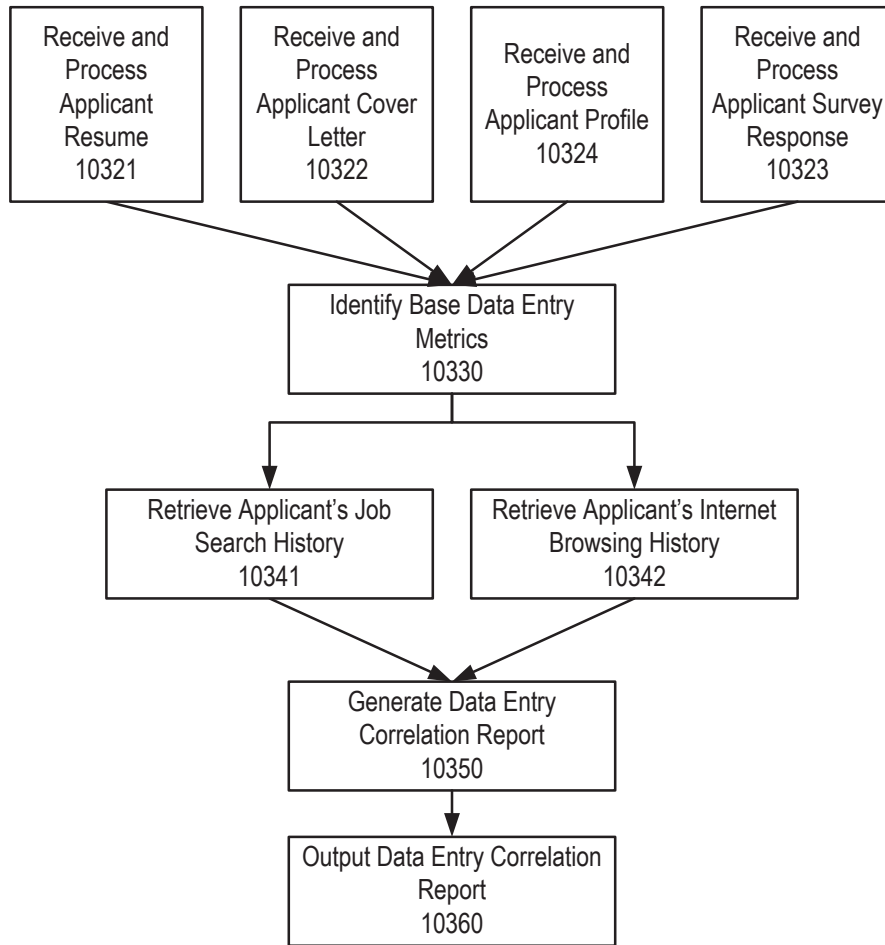
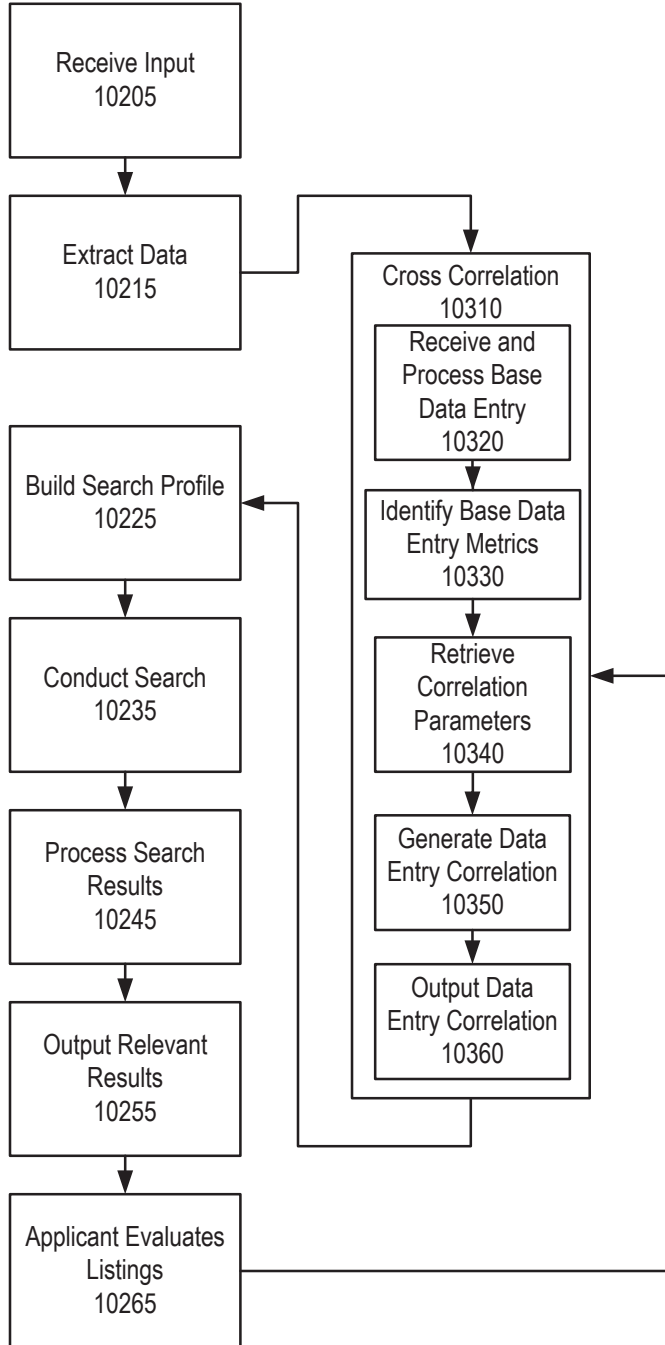


FIG. 103B



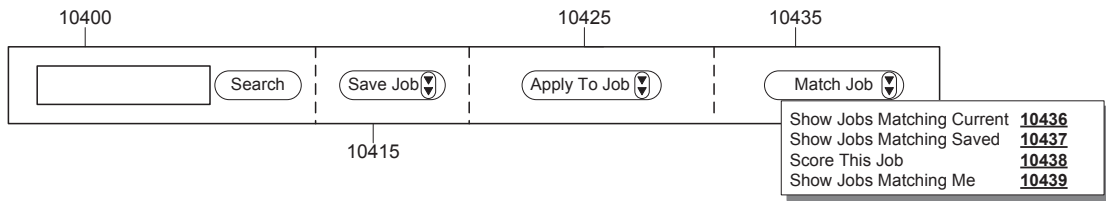


FIG. 104A

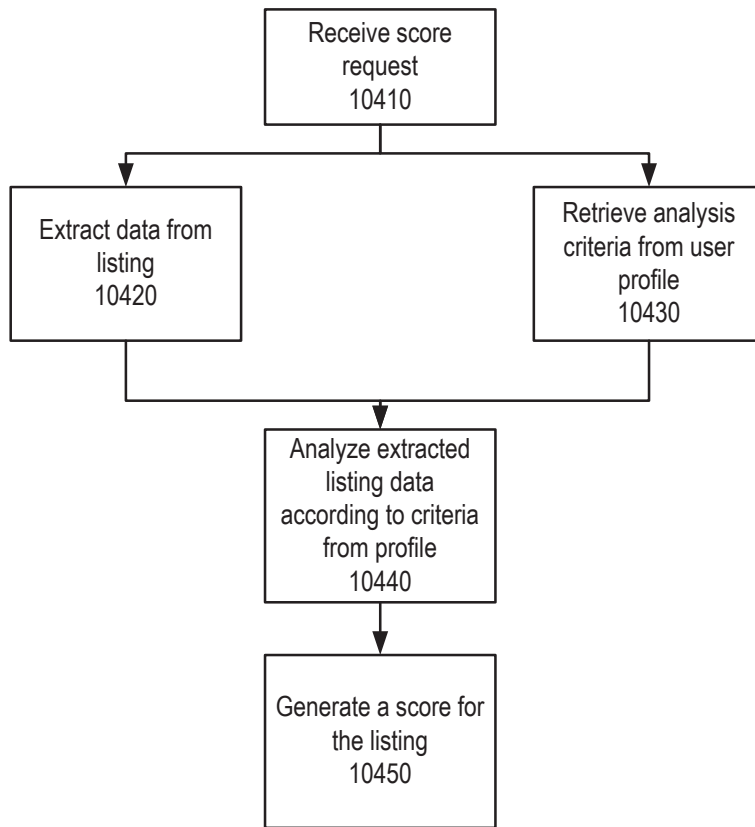


FIG. 104B

145/167

10400

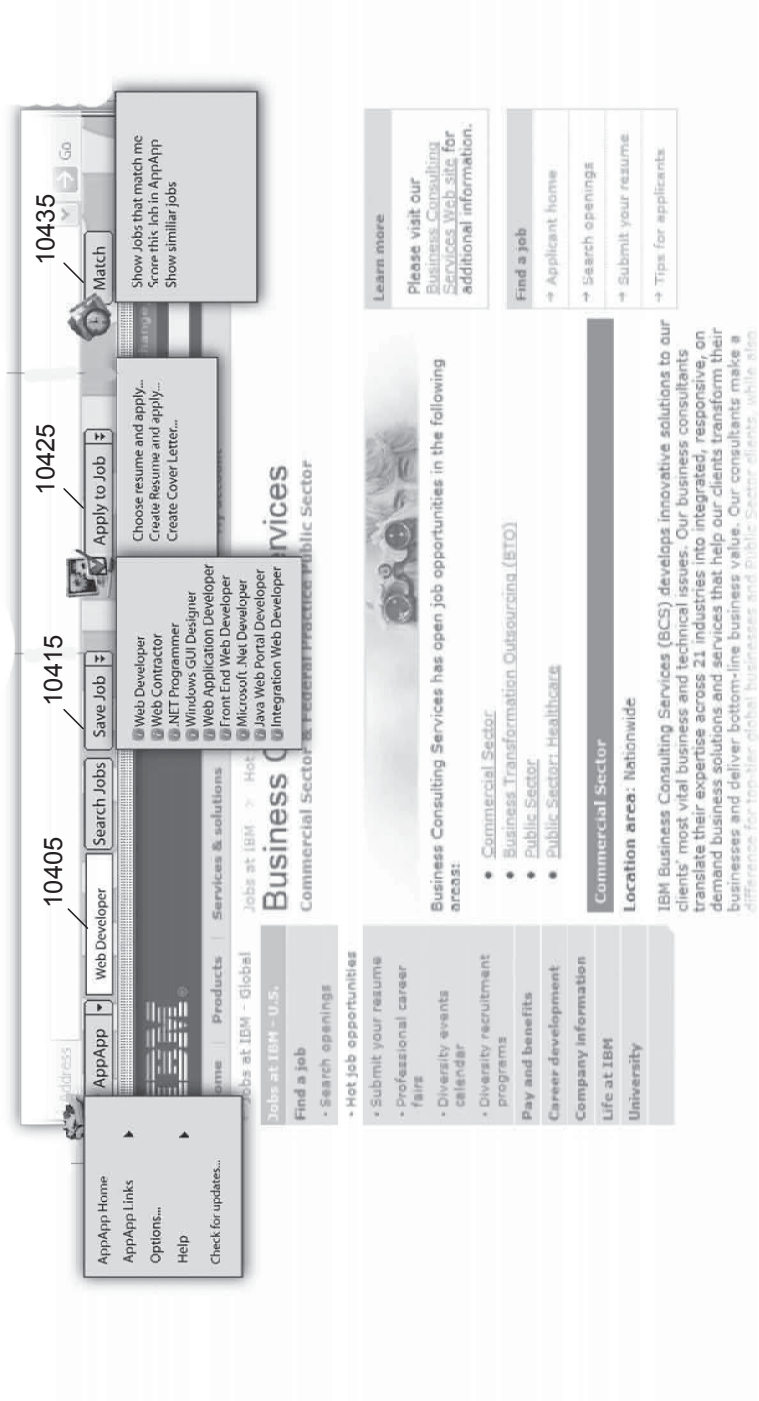


FIG. 104C

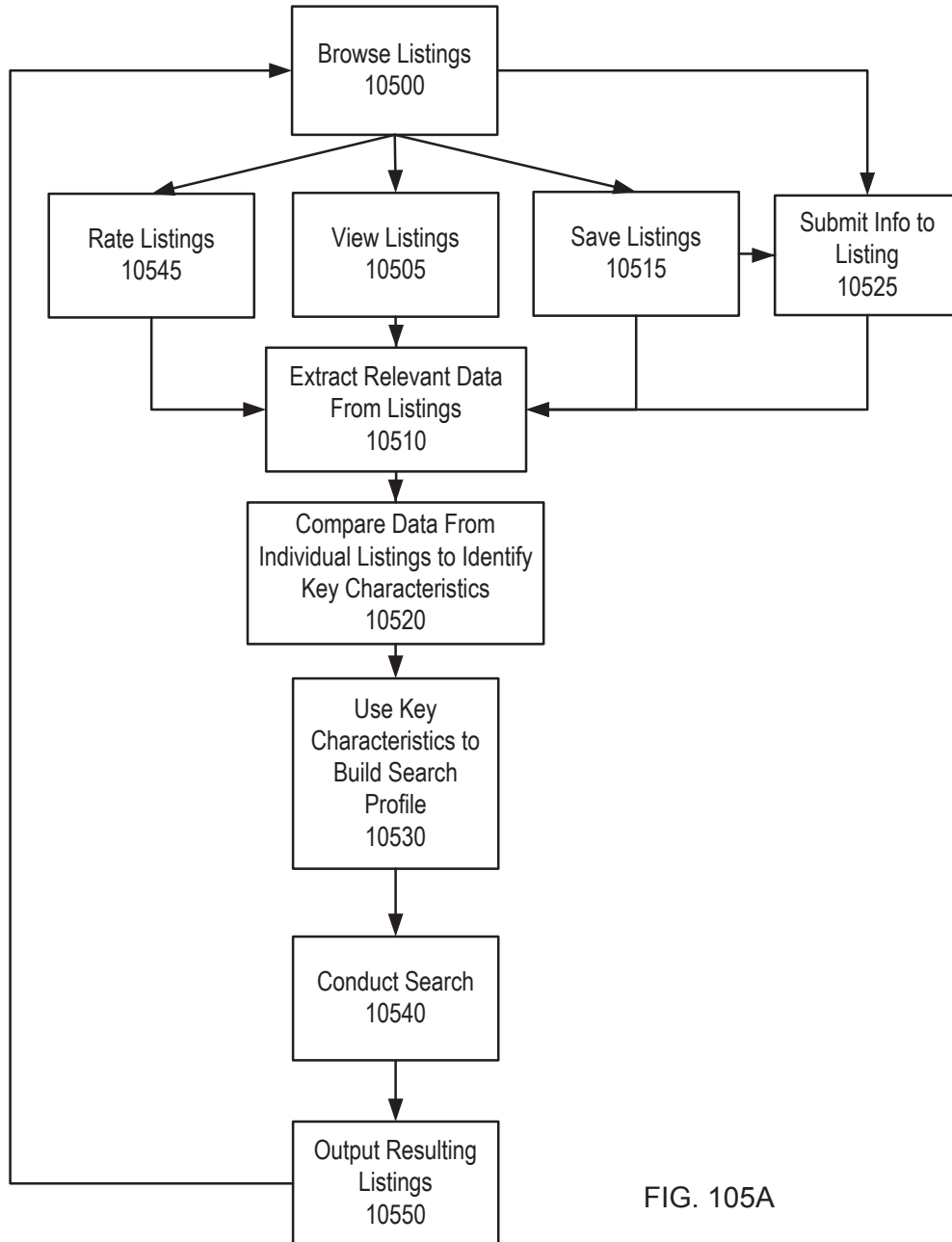


FIG. 105A

147/167

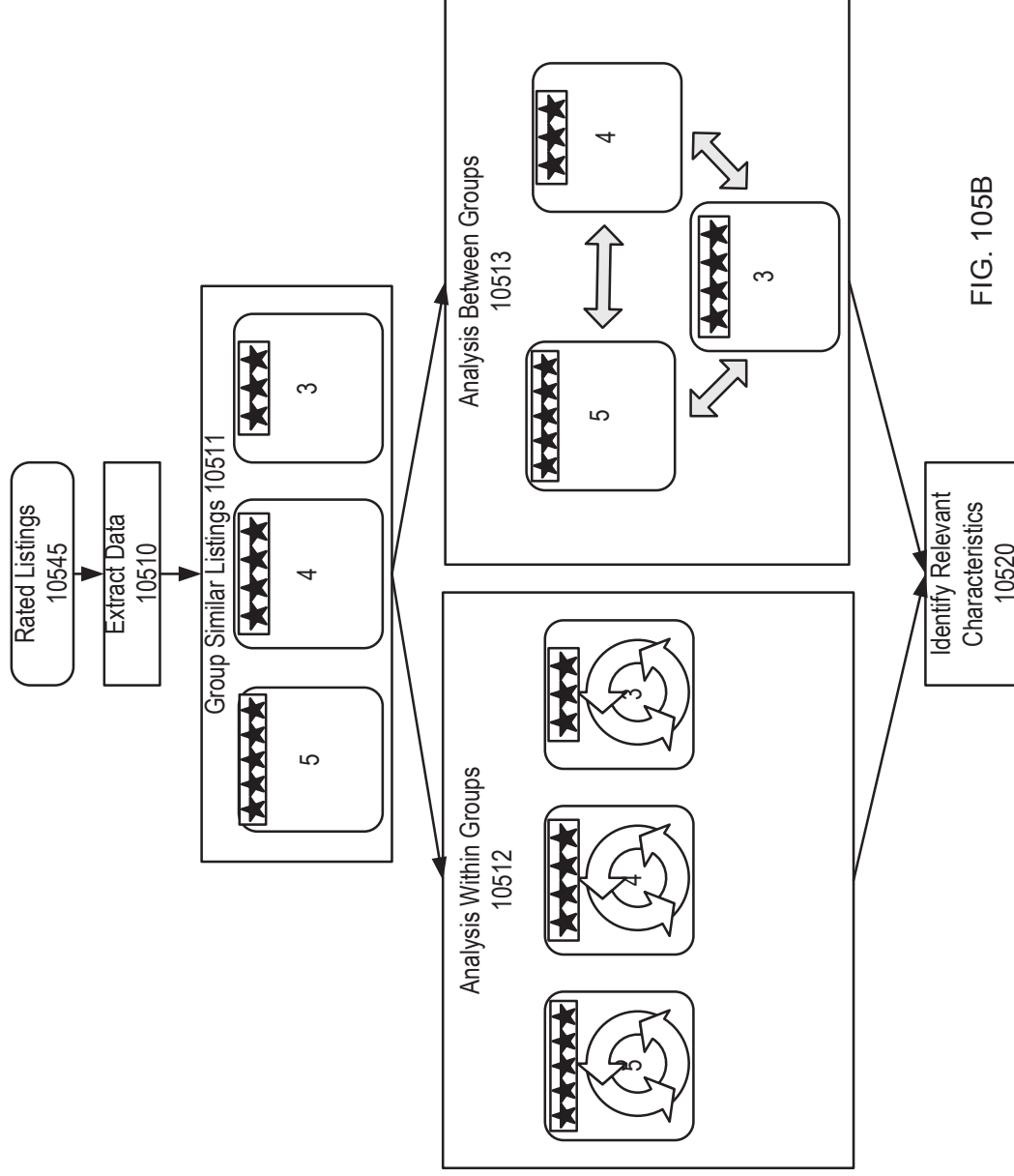


FIG. 105B

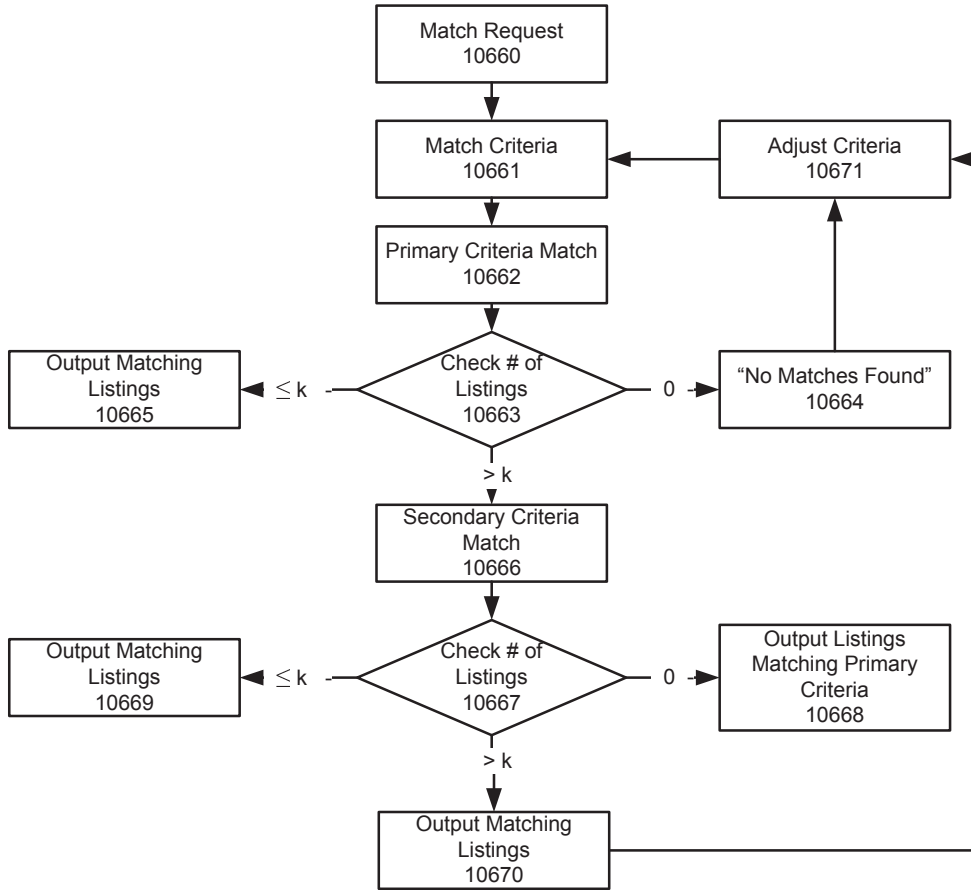


FIG. 106A

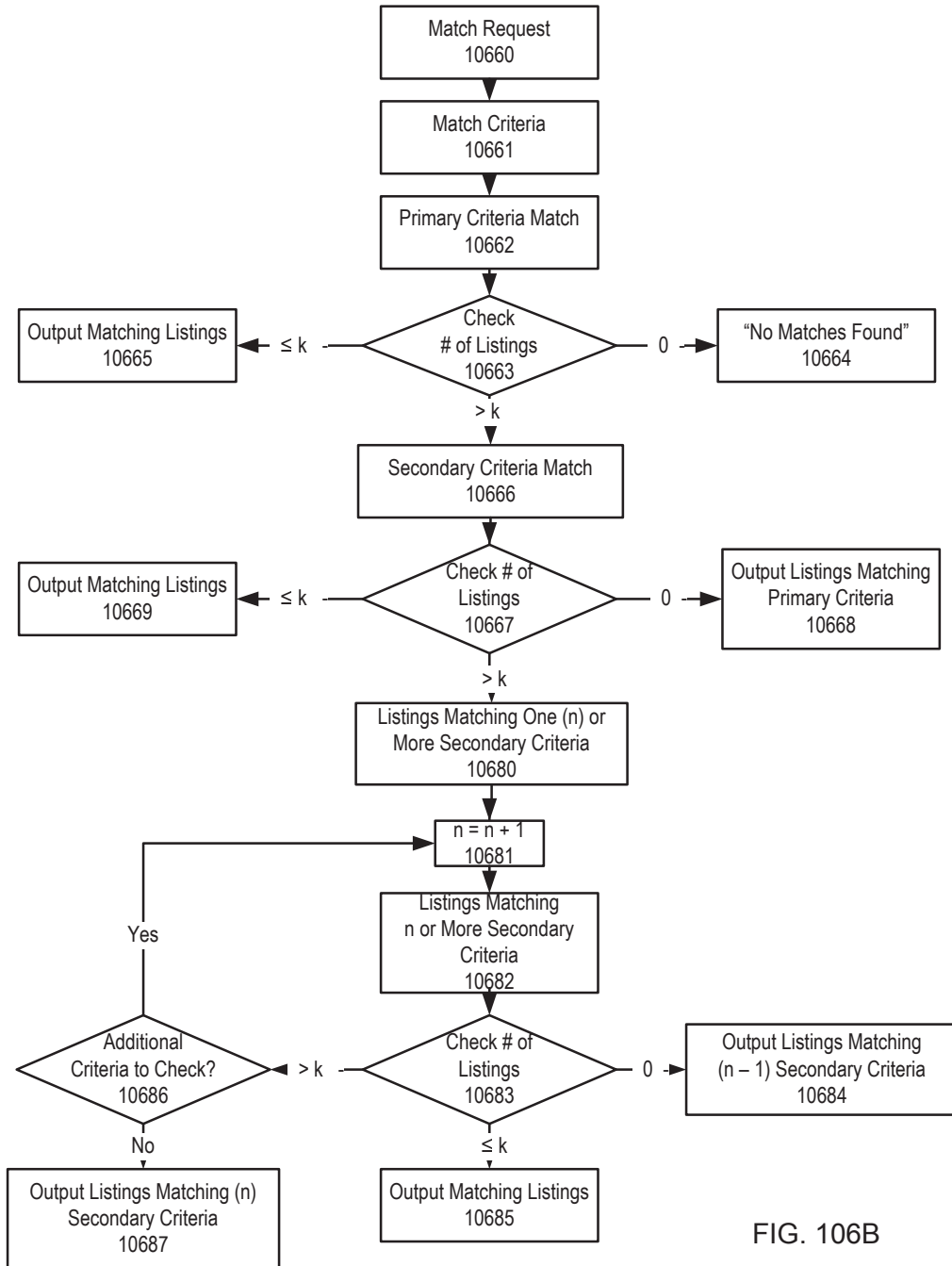


FIG. 106B

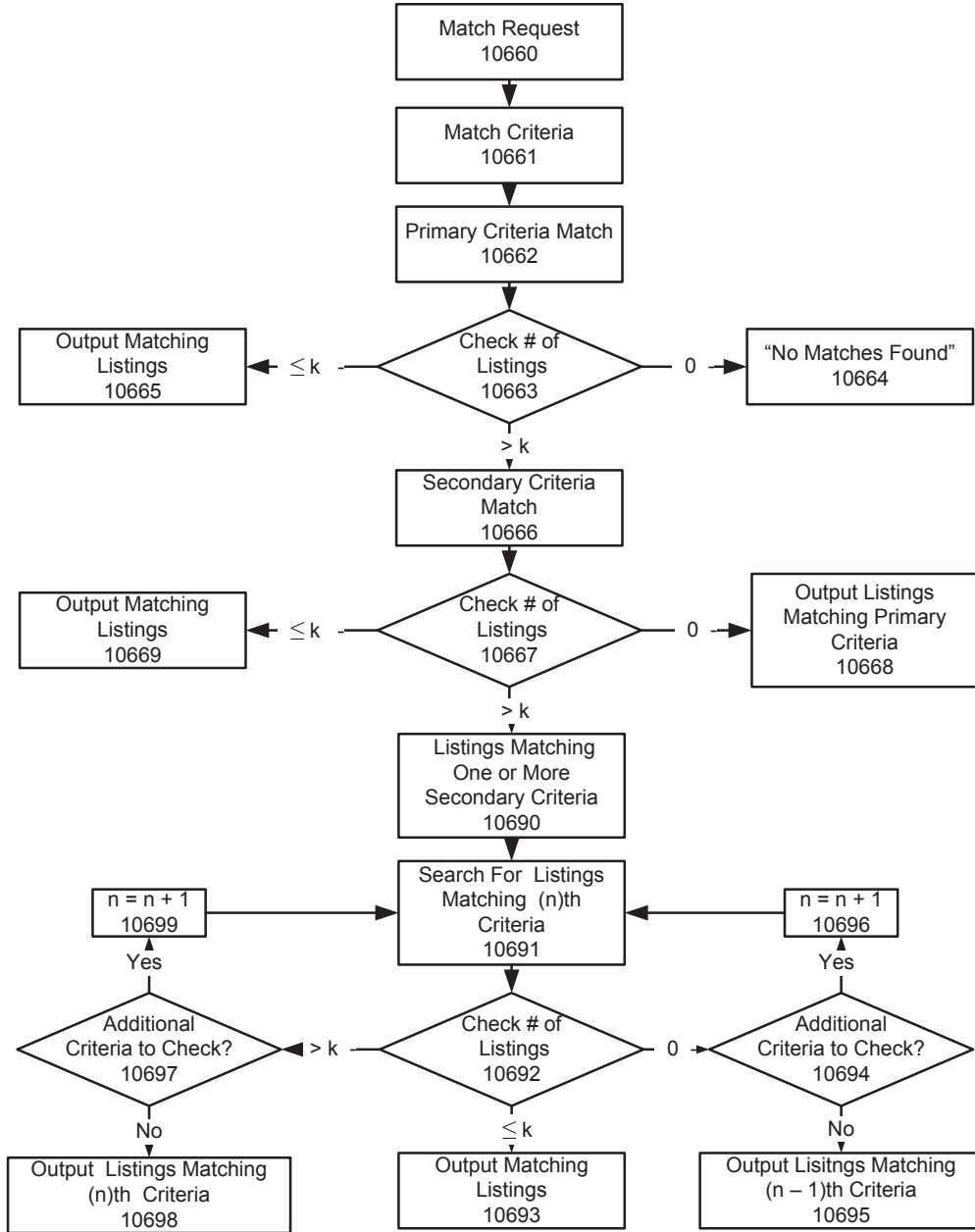


FIG. 106C

151/167

10701 Job Title	10702 Company	10703 Location	10704 Date Applied	10705 Corres.	10706 Status	10707 Score
Web Developer	Monster	Maynard, Ma	1/9	10	Offer	95
Web Developer	Monster	Maynard, Ma	1/10	7	HR Rev.	95
Web Developer	Monster	Maynard, Ma	1/12	3	2 nd Int.	95
Web Developer	Monster	Maynard, Ma	2/15	2	Opened	95
Web Developer	Monster	Maynard, Ma	3/15	1	Received	95
Web Developer	Monster	Maynard, Ma	-----	0	-----	95
Web Developer	Monster	Maynard, Ma	-----	0	-----	95
Web Developer	Monster	Maynard, Ma	-----	0	-----	95
Web Developer	Monster	Maynard, Ma	-----	0	-----	95
Web Developer	Monster	Maynard, Ma	-----	0	-----	95
Web Developer	Monster	Maynard, Ma	-----	0	-----	95
Web Developer	Monster	Maynard, Ma	-----	0	-----	95
Web Developer	Monster	Maynard, Ma	-----	0	-----	95

FIG. 107A

10715 User Rating	10701 Job Title	10702 Company	10703 Location	10704 Date Applied	10705 Corres.	10706 Status	10707 Score
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	1/9	10	Offer	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	1/10	7	HR Rev.	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	1/12	3	2 nd Int.	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	2/15	2	Opened	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	3/15	1	Received	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	-----	0	-----	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	-----	0	-----	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	-----	0	-----	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	-----	0	-----	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	-----	0	-----	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	-----	0	-----	95
☆☆☆☆☆☆	Web Developer	Monster	Maynard, Ma	-----	0	-----	95

FIG. 107C

AppApp Application History

Screen employers for a better match.

- Manage your resumes, and applications.
- View status and responses to your applications.
- Read and review companies you've applied too.
- Track sent, received and opened email.

Logged in as: DaveD.
Company: Monster
Total Applies: 5
Last Log In: 9-3

You have 3 new responses

My Tools:

- Account Preferences
- Reports
- Hotlist
- User Data
- Change Layout
- Change Content

Dynamic "tripwire"/upsell/special offer. Drives eBiz contextual to the user.

My Applications

Job Title	Company	Opened	Viewed	Responses	Location	Date Applied
Web Developer	Monster	No	YES	3	Maynard, MA	9-3
Web Developer	Monster	YES	YES	3	Maynard, MA	9-3
Web Developer	Monster	YES	YES	9	Maynard, MA	9-3

Reviews of this Company

Reviewer	Company	Opened	Viewed	Responses	Location	Date Applied
AppApp Reviewer Username #1	Monster	No	YES	3	Maynard, MA	0
AppApp Reviewer Username #2	Monster	YES	YES	3	Maynard, MA	0
AppApp Reviewer Username #3	Monster	YES	YES	3	Maynard, MA	0
AppApp Reviewer Username #4	Monster	YES	YES	3	Maynard, MA	0

Job Posting ALERT

Position Filled
Monster.com
CareerBuilder.com
HotJobs.com
Job Updated
Job Removed

December

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Last Correspondence: 17 DAYS AGO

FIG. 107B

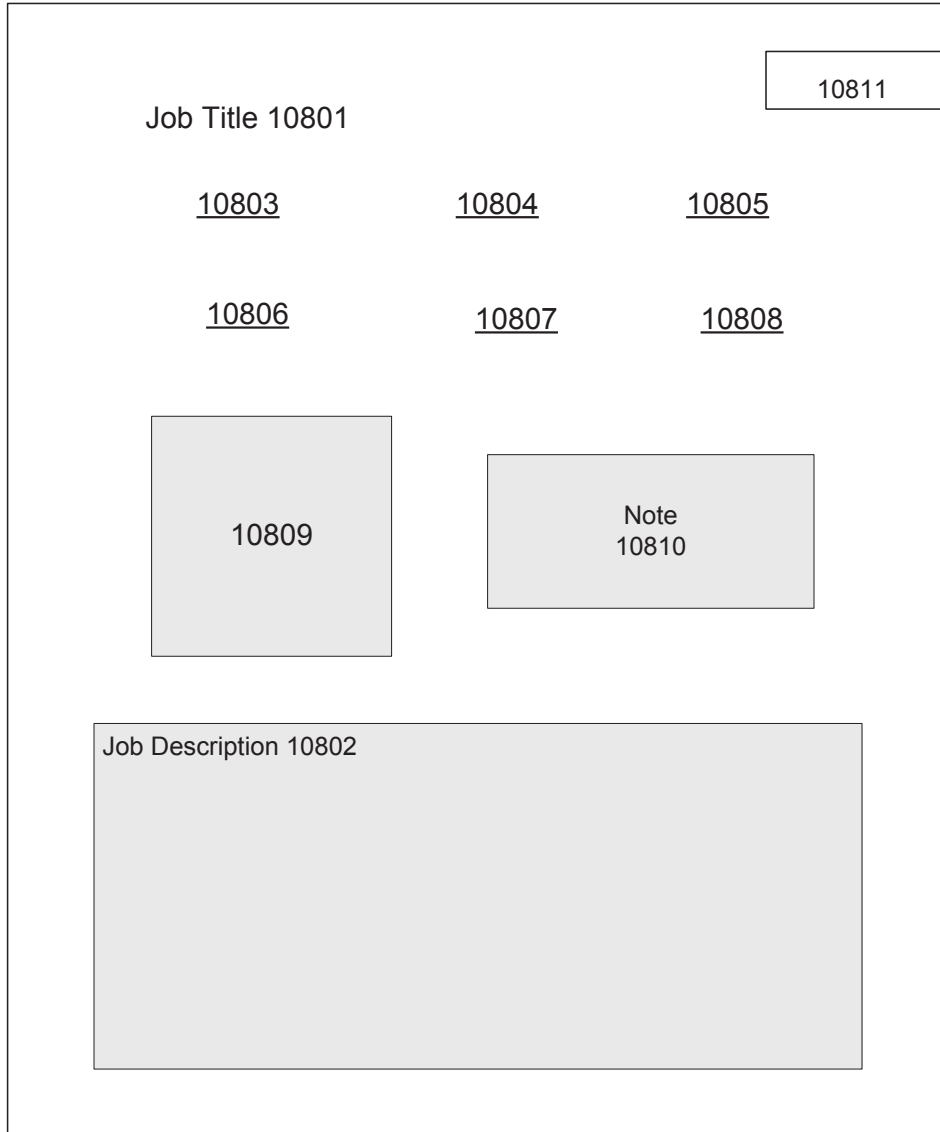


FIG. 108A

AppApp Job View

View saved job postings.

- View job freshness.
- View status on jobs posted.
- View job posting statistics.
- Find similar jobs, similar responsibilities and different titles that match this job.

Job Posting View

View the same job posting from other sites:

Flash Designer / Developer

Company: Pokémon USA, Inc.	Location: New York, New York
Status: Full Time, Employee	Job Category: Internet/Commerce
Required Work Experience: 2+ to 5 Years	Career Level: Experienced
Education Level: Bachelor's Degree	

Apply Now

Logged in as: DaveD.
Company: Monster
Total Applies: 5
Last Log In:

You have **3** new responses

My Tools:

- Account Preferences
- Reports
- Hotlist
- User Data
- Change Layout
- Change Content

This job has been posted **7** times in the last year

Active for **22** days.

Last Contact

- 9 Responses to this job
- Company Data
- Posting Status Alerts
- Find Similar Jobs

Save Notes here on this Job. **save notes**

Job Description

Pokémon USA, Inc., the New York Based subsidiary of The Pokémon Company in Japan, manages and oversees the Pokémon franchise in North and Latin America, which includes licensing, merchandising, TV animation, theatrical releases, home video entertainment, the official Pokémon website and PokemonCenter.com, an e-commerce site. Since its launch in North America in 1998, Pokémon has evolved into a global cultural phenomenon generating over \$15 billion in worldwide retail sales.

Pokémon USA, Inc. is seeking to fill an intermediate level Flash Designer/Developer in our Creative Department. This position reports to the Creative Director.

Responsibilities:

- Plan, design, develop, and implement Flash experiences for our fan website, e-commerce website, clients, customers and the organization.
- Work on multiple projects from requirements, generation through design, specification, coding, animation and debugging.
- Responsible for the daily upkeep of www.pokemon.com, game development, flash banners, animation and mini-site development.
- You must be comfortable working independently or as part of a team with creative and technical divisions.

Requirements:


- Strong design skills using Photoshop, Illustrator or comparable graphics products.

December

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Last Correspondence: **17 DAYS AGO**

FIG. 108B



[Search Jobs](#)
[My Monster](#)
[For Employers](#)

★★★★★★
10812

☆☆★★★
10804

☆★★★★★
10802

★★★★★★
10807

★★★★★★
10808

☆☆★★★★★
10813

Software Engineer

ABC Software Co.

Starting Pay: \$100,000

Job Location:
New York, New York


Job Description:
Will design, develop and maintain software on Microsoft .Net platform. Serve as technical lead on development efforts, including participating in requirements analysis and design. Will have responsibility for developing solutions for all tiers of the application; code-behinds, controls, middle-tier business components, database layer, etc. Assist in formulating standards and development practices across multiple related teams.

Qualifications:

- * Two or more years of extensive experience with C# and ASP.NET is ideal.
- * Knowledge in OO development principles, multi-tier web application design and relational database principles required.
- * History of delivering XML-based, Webservice, Component-Based and Multi-Threaded applications to production is desired.
- * Experience working with large project team development environments.
- * Good fundamental and hands on knowledge of with MS SQL Server.
- * Strong Analytical Skills and good problem solving skills.
- * Experience working in a regionally distributed development group.
- * Prior use of NAnt, NUnit, NDoc and CruiseControl.Net, or similar products would be a plus.

The Benefits:

<p>Health benefits: Blue Cross and Blue Shield PPO Program Guardian DMO and Dental PDO Plans Vision care Flexible Spending Accounts Supplement life insurance for employees and their families Discounted auto and homeowners insurance</p>	<p>Employer provided benefits: Life Insurance Short Term and Long Term Disability Plans Employee Assistance Plan Competitive Paid Time Off program Education Tuition Reimbursement Plan for Continuing Education</p>
--	---



[APPLY NOW](#)

Fig. 108C

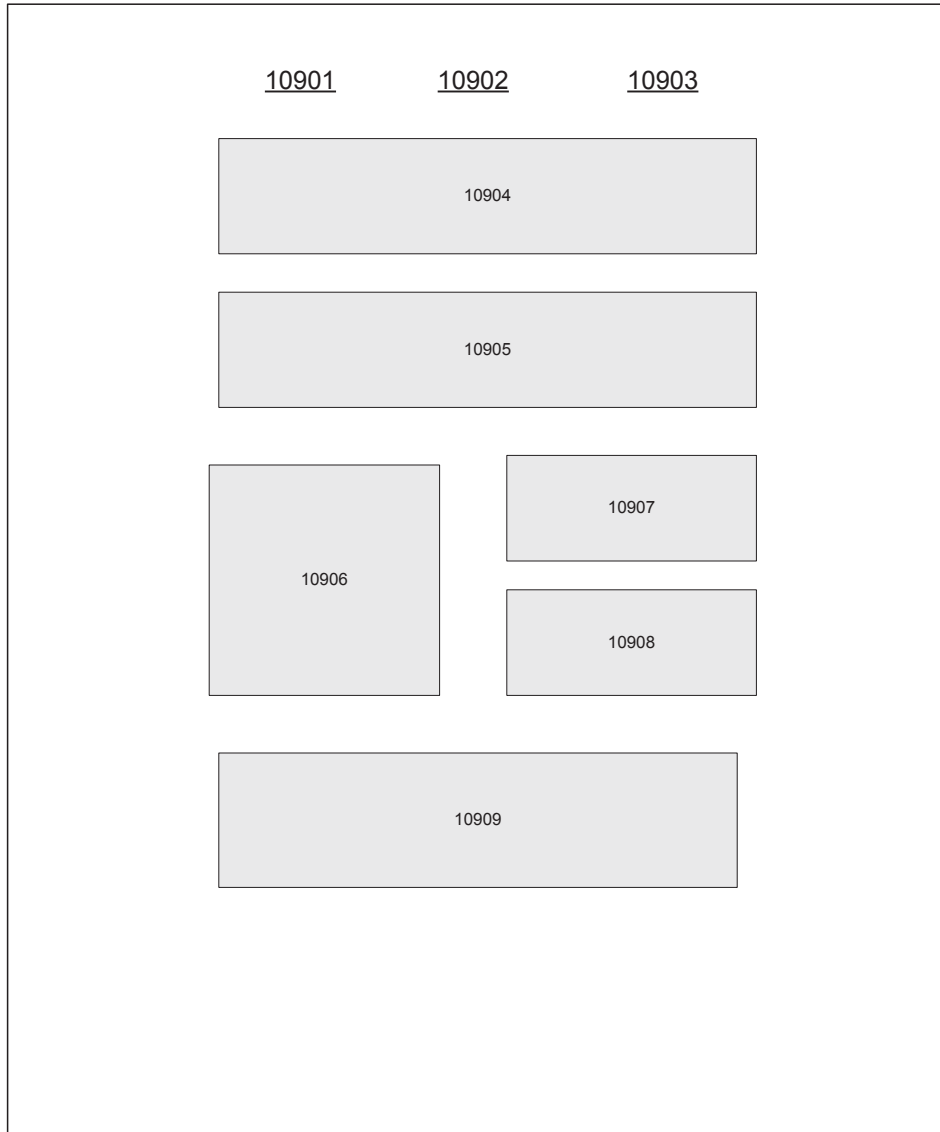


FIG. 109A

157/167



FIG. 109B

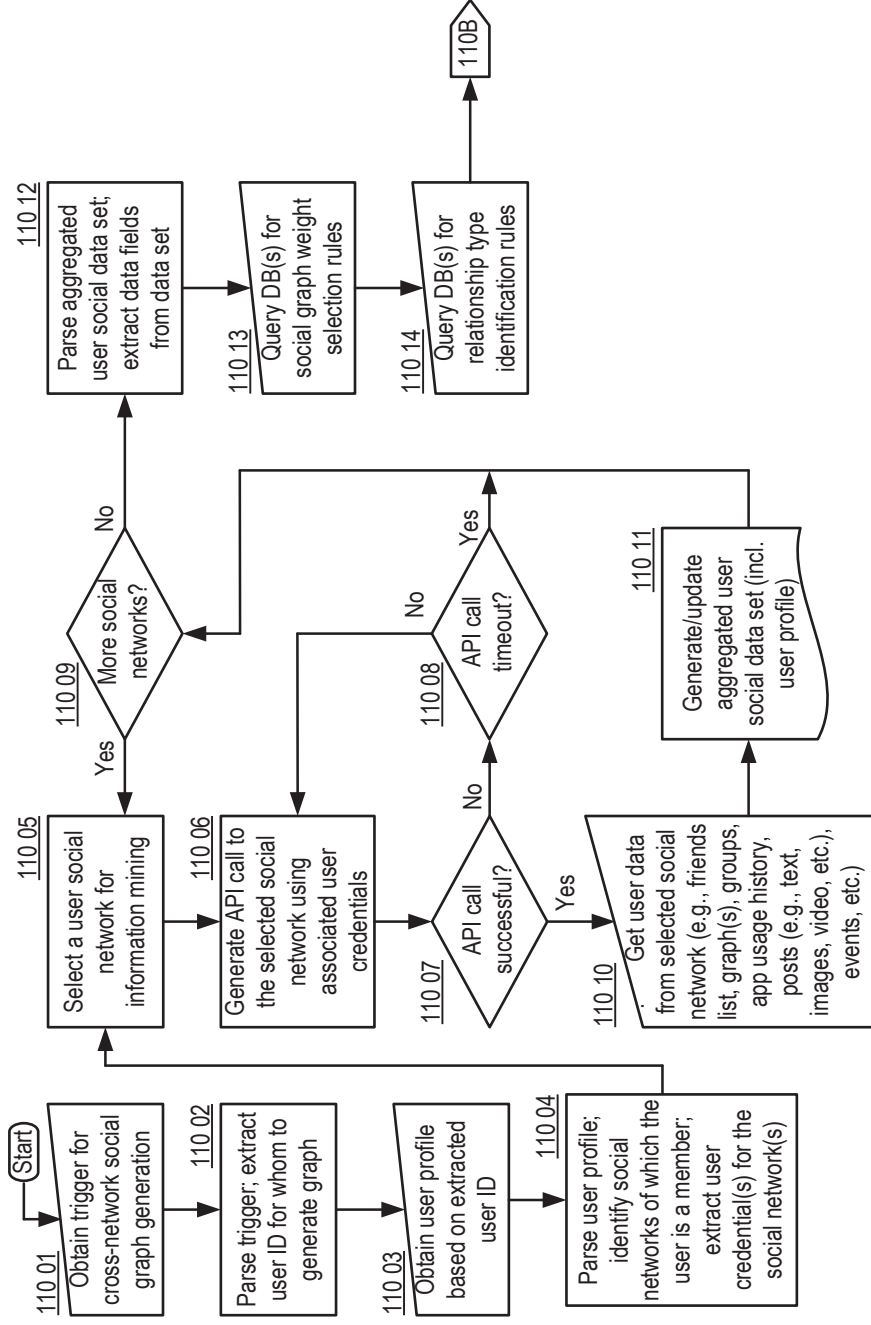


FIGURE 110A

Example Logic Flow: Cross-Network Social Graph Updating (CN-SGU) component

159/167

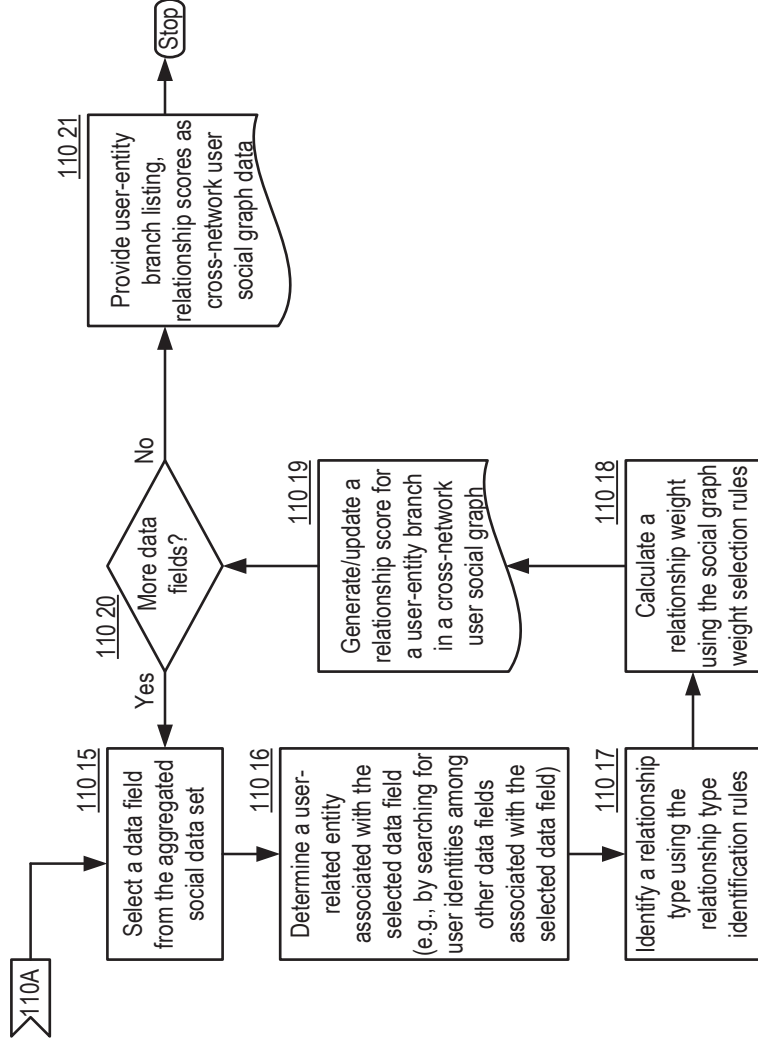


FIGURE 110B

Example Logic Flow: Cross-Network Social Graph Updating (CN-SGU) component

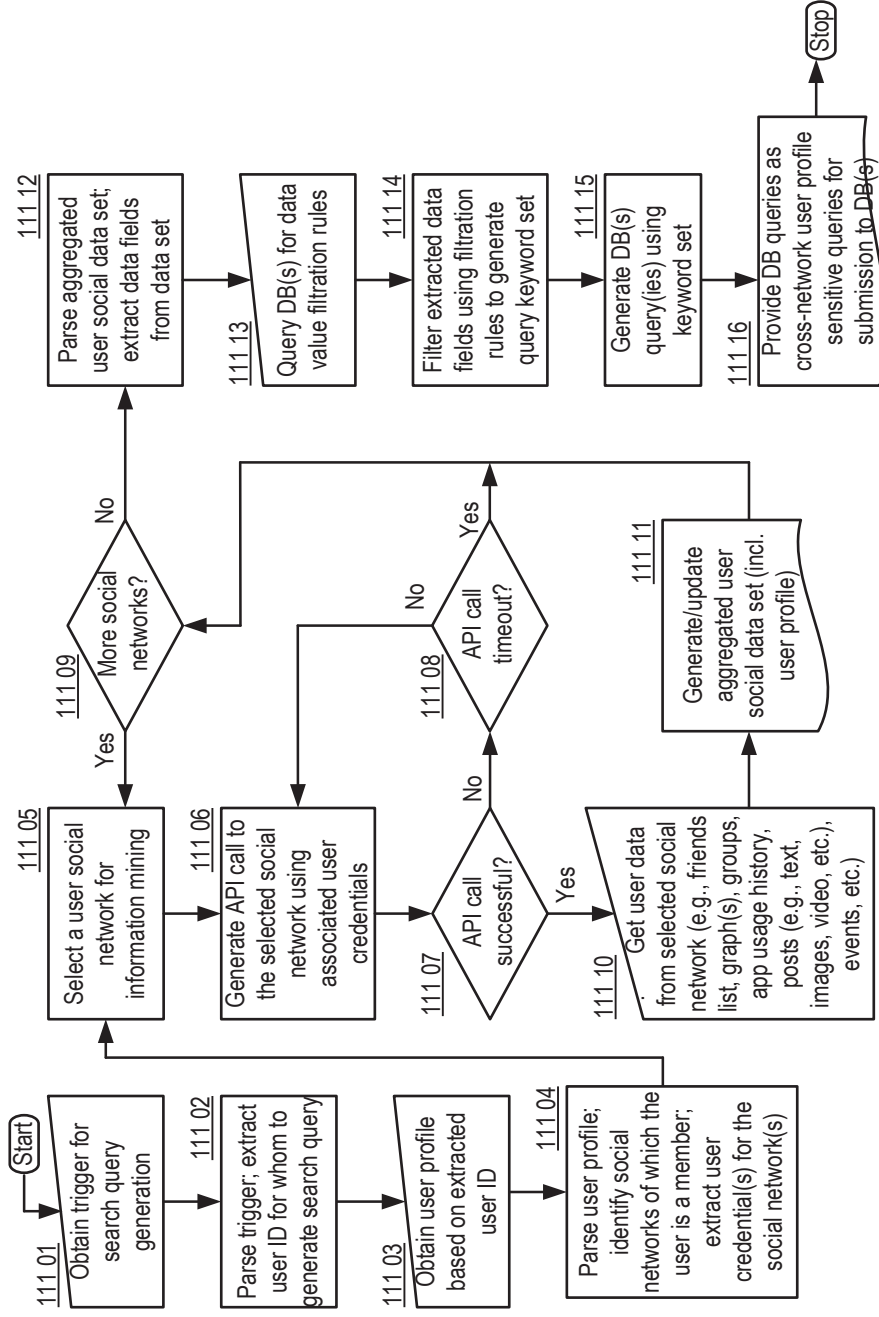


FIGURE 111

Example Logic Flow: Cross-Network User Profile Sensitive Query Generation (CN-UPSQG) component

FIGURE 112

facebook Search Home Profile Find Friends Account

BeKnown™

The Professional Networking App on Facebook

Join BeKnown

Discover BeKnown

Use BeKnown to connect professionally, even if you're not Facebook friends.

More Convenience - Build professional relationships on Facebook.

More Control - Keep your professional life separate from Facebook friends.

More Connections - Expand your network of colleagues and companies.

Want to know more? Find out How it Works

FIGURE 113

facebook Search Home Profile Find Friends Account

BeKnown Messages Profile Network Jobs Post a Job Company Search

David Kelly
Master Electrician
Morrissey Electric
david.kelly@morrisseyllc.com

All Badges (7)

Professional Network Stats

- 522 Connections
- 43,456 2nd Degree Connections
- 86 Followed Companies
- 98,267 Companies

Endorse Your Connections

- Rick Steele Endorse
- Remee Holland Endorse
- Tony Peck Endorse

Your Professional Profile is 35% Complete

Like BeKnown and get to 40%

I'll do this later. Give me another task.

Invite Your Facebook Friends

Connect with these friends to grow your professional network.

- Lisa Mahmud Invite
- Marcy Newman Invite
- Matt Pittman Invite

Updates

David Kelly joined your network on BeKnown.

- + 384 connections added to your network.
- + Connected to following companies (+151 more).

1 hour ago

Network Updates - Jobs For You

Sponsored

Games

FIGURE 114

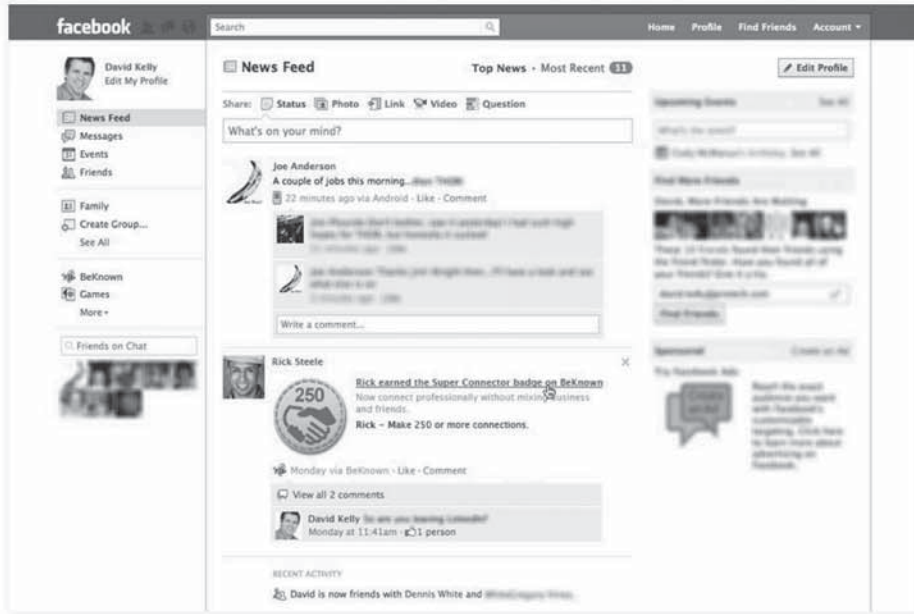


FIGURE 115

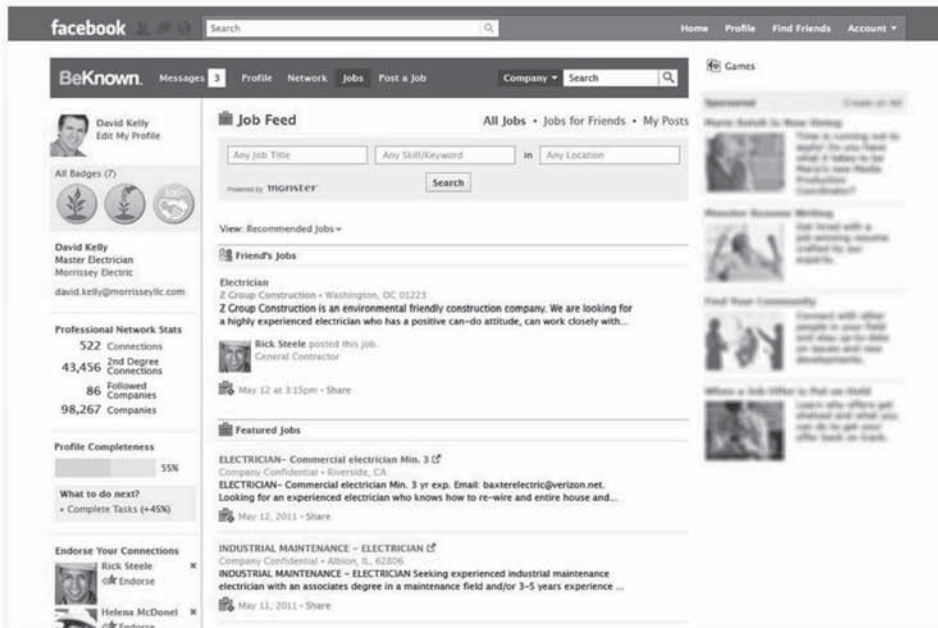


FIGURE 116

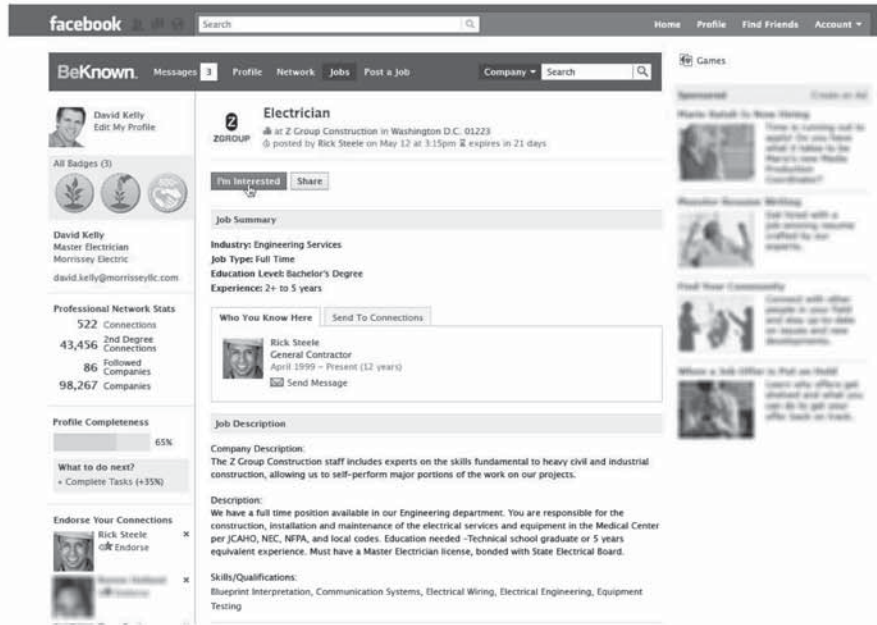


FIGURE 117

How it Works

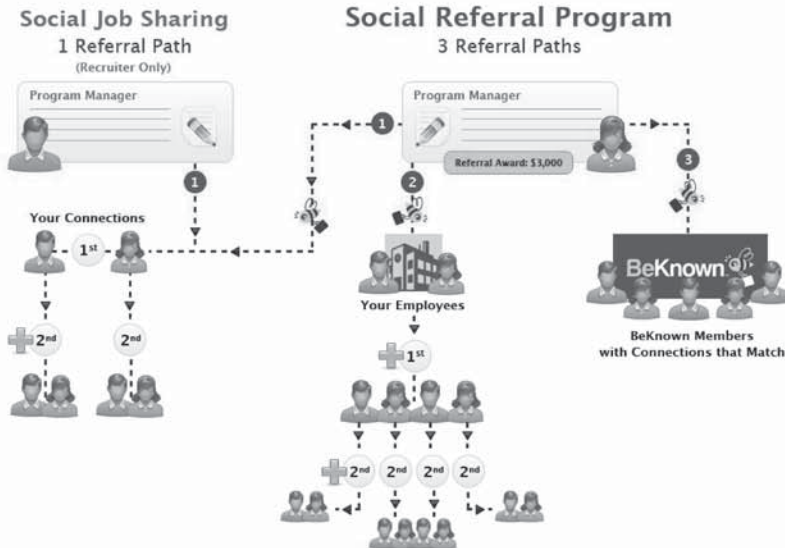


FIGURE 118

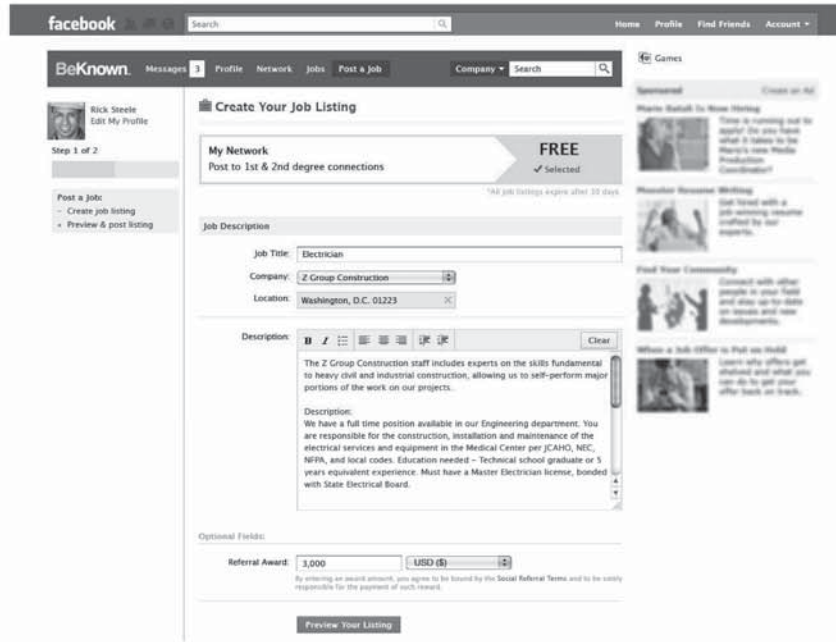


FIGURE 119

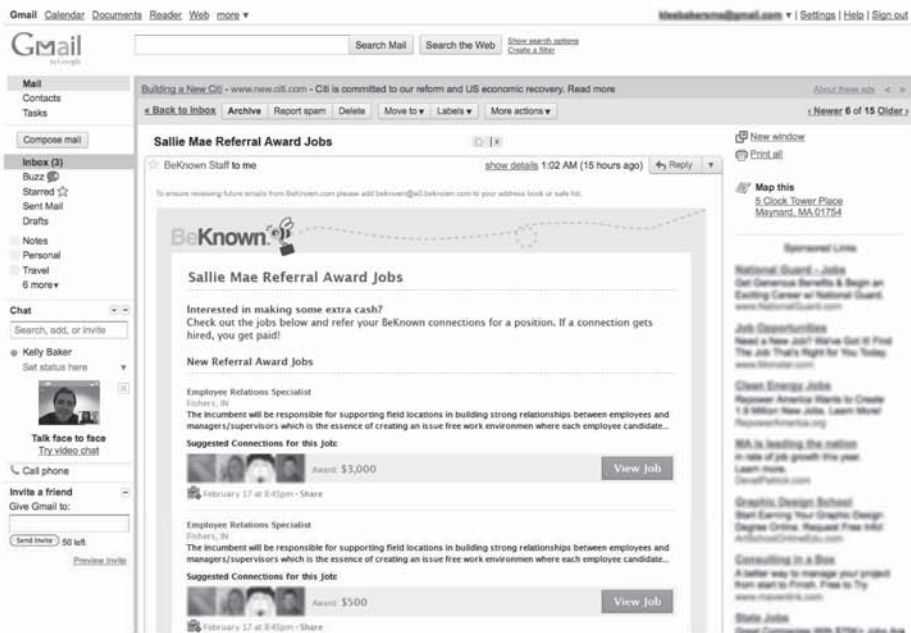


FIGURE 120

The screenshot shows a Facebook job posting for an Electrician at Z Group Construction. The job is posted by Rick Steele on May 12 at 3:15pm and expires in 21 days. The job summary includes a referral award of \$3,000. The job description states that the Z Group Construction staff includes experts on heavy civil and industrial construction. The description also mentions that the position is available in the Engineering department and requires a Master Electrician license and a 5-year equivalent experience. The skills/qualifications listed include Blueprint Interpretation, Communication Systems, Electrical Wiring, Electrical Engineering, and Equipment Testing.

FIGURE 121

The screenshot shows a Facebook message conversation. The message is from Denise Silber, Marketing Manager at Z Group Construction, dated May 13 at 4:17pm. Denise Silber recommends a job on BeKnown for an Electrician at Z Group Construction. The message text reads: "Denise Silber thinks you'll like this job: Electrician at Z Group Construction". Below the message, there is a "View Job Details" button and a reply box with a "Reply" button. The reply box contains the text: "Not the job for you? No prob. Maybe you'll like one of these jobs."

FIGURE 122

The screenshot shows a Facebook job posting for an Electrician position. The job is posted by Rick Steele on May 12 at 3:15pm and expires in 21 days. The job summary includes a referral award of \$3,000. The job description states that the Z Group Construction staff includes experts on the skills fundamental to heavy civil and industrial construction, allowing us to self-perform major portions of the work on our projects. The description also mentions that we have a full time position available in our Engineering department, responsible for the construction, installation and maintenance of the electrical services and equipment in the Medical Center per JCAHO, NEC, NFPA, and local codes. Education needed - Technical school graduate or 5 years equivalent experience. Must have a Master Electrician license, bonded with State Electrical Board. Skills/Qualifications: Blueprint Interpretation, Communication Systems, Electrical Wiring, Electrical Engineering, Equipment Testing. The job is posted by Rick Steele, General Contractor at Z Group Construction.

FIGURE 123

This screenshot is identical to Figure 122, but it shows the job posting after a user has interacted with it. A 'Thanks for Your Interest' button is now visible at the top of the job summary section. The rest of the job details, including the referral award, description, and qualifications, remain the same.

FIGURE 124

The screenshot shows a Facebook message interface within the BeKnown application. At the top, there is a navigation bar with 'facebook' and search fields. Below this, the BeKnown header includes 'Messages', 'Profile', 'Network', 'Jobs', and 'Post a Job'. The left sidebar displays the profile of Rick Steele, a General Contractor at Z Group Construction, with 1 connection, 15 2nd degree connections, 0 followed companies, and 67 companies. The main message area shows a conversation with David Kelly, dated April 29 at 2:37pm. The message text reads: 'Hi Rick, Denise Silber recommended that I contact you about this position. I am writing to express my interest in the Electrician position that is posted on BeKnown, View David Kelly's Profile. Thank you for your consideration. I look forward to hearing from you. Sincerely, - David Kelly'. Below the message is a 'Reply:' field and a 'Reply' button. The right sidebar features a 'Games' section with several recommendations, including 'Mario Kart' and 'Puzzle Resources Writing'.