



(86) Date de dépôt PCT/PCT Filing Date: 2007/08/15  
(87) Date publication PCT/PCT Publication Date: 2008/02/21  
(45) Date de délivrance/Issue Date: 2013/06/11  
(85) Entrée phase nationale/National Entry: 2009/02/13  
(86) N° demande PCT/PCT Application No.: US 2007/018058  
(87) N° publication PCT/PCT Publication No.: 2008/021390  
(30) Priorité/Priority: 2006/08/15 (US60/838,074)

(51) Cl.Int./Int.Cl. *H04M 3/523* (2006.01),  
*H04M 3/51* (2006.01)

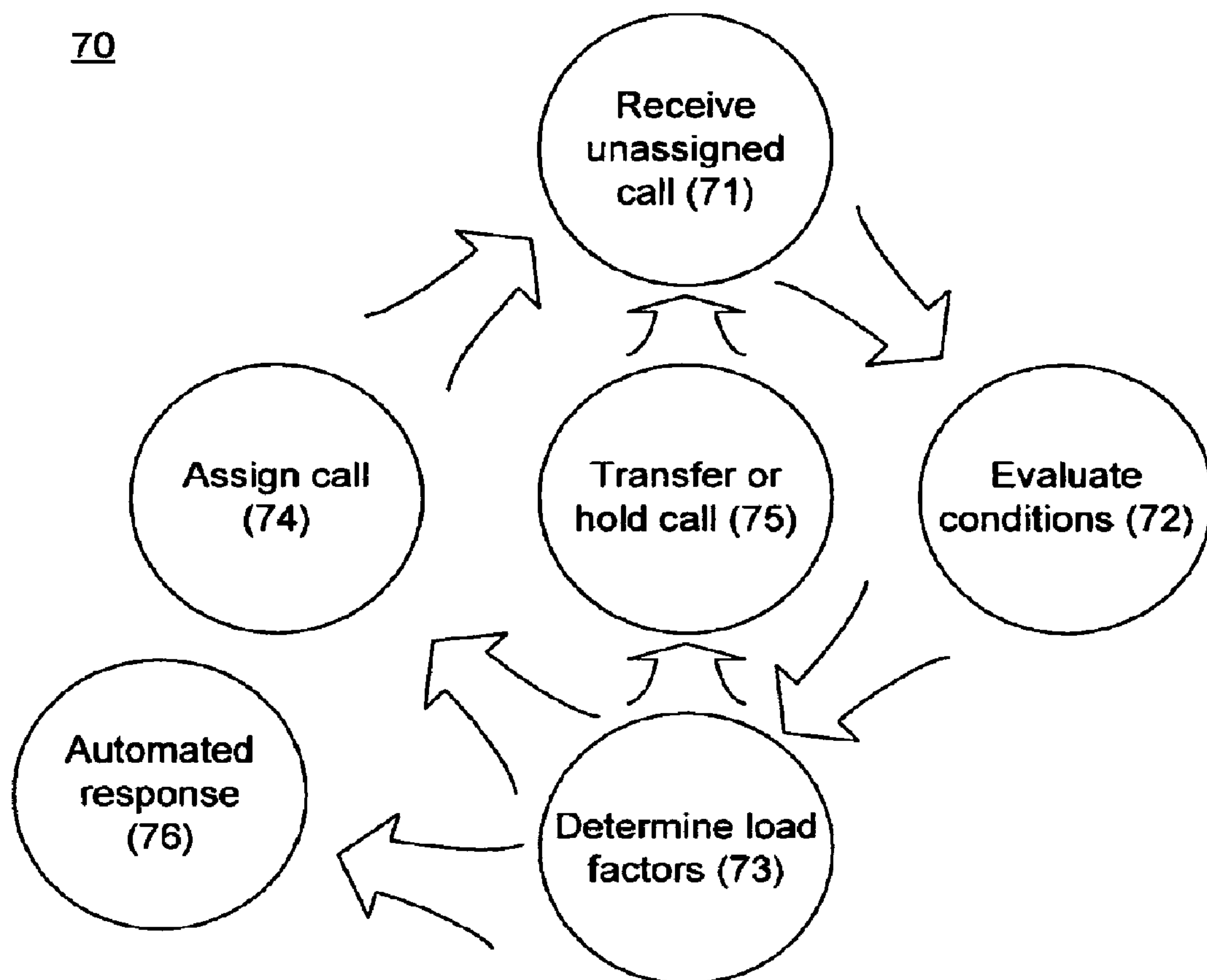
(72) Inventeurs/Inventors:  
WAALKES, ADAM, US;  
ODINAK, GILAD, US;  
JIANG, HAODONG (HOWARD), US;  
THORN, MATTHEW, US;  
MCCANN, TOM, US

(73) Propriétaire/Owner:  
INTELLISIST, INC., US

(74) Agent: INTEGRAL IP

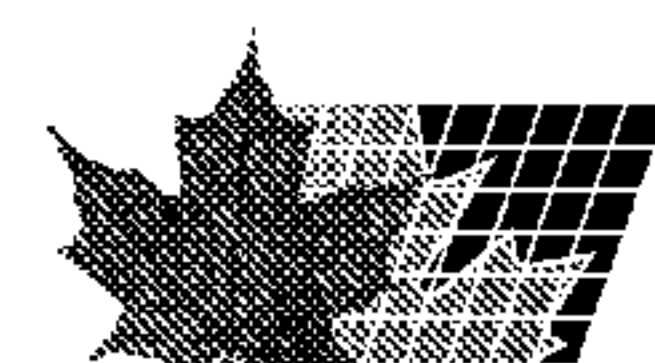
(54) Titre : EQUILIBRAGE DE CHARGE DE CONSOLE D'AGENT PENDANT UN TRAITEMENT D'APPEL AUTOMATISE  
(54) Title: BALANCING AGENT CONSOLE LOAD DURING AUTOMATED CALL PROCESSING

70



(57) Abrégé/Abstract:

A system and method for balancing agent console load during automated call processing (40) is provided. A plurality of agent consoles (101) are operated in a call center (30). Progress of a call session (40) of a caller that is calling into the call center (30) at



(57) **Abrégé(suite)/Abstract(continued):**

each agent console (101) is monitored (71). An absolute upper limit on a number of the call sessions (40) allowed at each agent console (101) at any given time is assigned (92). A load of call session (40) assignments on each agent console (101) is balanced. Objective and subjective conditions (81) pertinent to each assigned call session (40) are continually collected (72) from each agent console (101). The objective and subjective conditions (81) that affect call session (40) availability are updated (72) for each agent console (101). The objective and subjective conditions (81) are evaluated (72) against the absolute upper limit to determine a load factor (109). Each call session (40) is assigned (74) to one such agent console (101) within bounds applicable to the load factor (109).

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

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
21 February 2008 (21.02.2008)

PCT

(10) International Publication Number  
**WO 2008/021390 A1**

(51) International Patent Classification:  
*H04M 3/523* (2006.01) *H04M 3/51* (2006.01)

(74) Agents: WITTMAN, Krista, Anne et al.; 500 Union Street, Suite 1005, Seattle, Washington 98101 (US).

(21) International Application Number:  
PCT/US2007/018058

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

(22) International Filing Date: 15 August 2007 (15.08.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/838,074 15 August 2006 (15.08.2006) US

(71) Applicant (*for all designated States except US*): INTEL-LISIST, INC. [US/US]; 11061 NE 2nd Street, Suite 200, Bellevue, Washington 98004 (US).

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

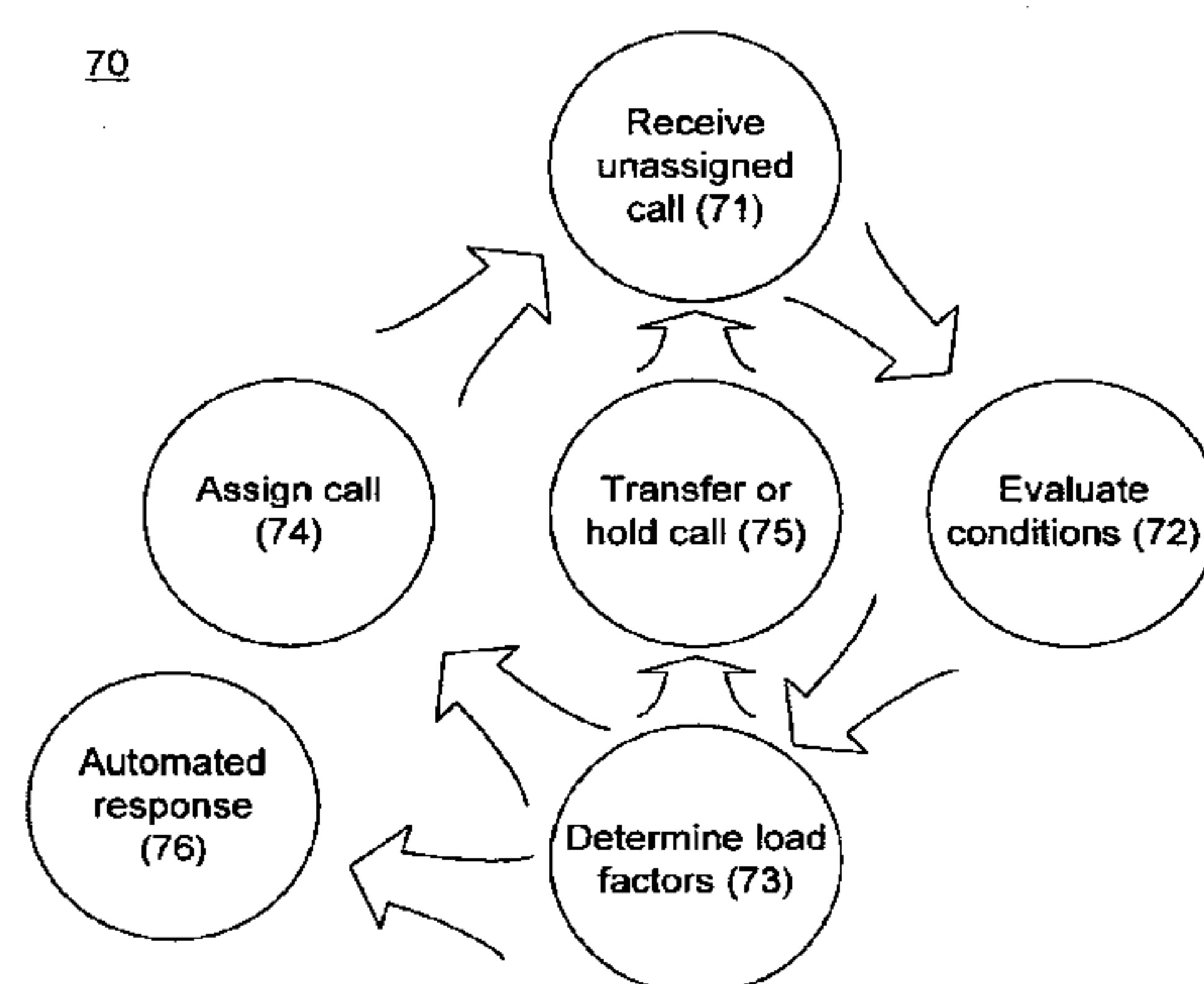
(72) Inventors; and

(75) Inventors/Applicants (*for US only*): WAALKES, Adam [US/US]; 847 Windsor Drive SE, Sammamish, Washington 98074 (US). ODINAK, Gilad [US/US]; 268 W Lake Sammamish Pkwy NE, Bellevue, Washington 98008 (US). JIANG, Haodong (Howard) [CN/US]; 25235 SE Klahanie Blvd. K101, Issaquah, Washington 98029 (US). THORN, Matthew [US/US]; 620 Highland Drive, Apt. A, Seattle, Washington 98109 (US). MCCANN, Tom [US/US]; 1525 SW 15th Place, North Bend, Washington 98045 (US).

## Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

(54) Title: BALANCING AGENT CONSOLE LOAD DURING AUTOMATED CALL PROCESSING



(57) Abstract: A system and method for balancing agent console load during automated call processing (40) is provided. A plurality of agent consoles (101) are operated in a call center (30). Progress of a call session (40) of a caller that is calling into the call center (30) at each agent console (101) is monitored (71). An absolute upper limit on a number of the call sessions (40) allowed at each agent console (101) at any given time is assigned (92). A load of call session (40) assignments on each agent console (101) is balanced. Objective and subjective conditions (81) pertinent to each assigned call session (40) are continually collected (72) from each agent console (101). The objective and subjective conditions (81) that affect call session (40) availability are updated (72) for each agent console (101). The objective and subjective conditions (81) are evaluated (72) against the absolute upper limit to determine a load factor (109). Each call session (40) is assigned (74) to one such agent console (101) within bounds applicable to the load factor (109).

WO 2008/021390 A1

CSCD007-1CA

**BALANCING AGENT CONSOLE LOAD  
DURING AUTOMATED CALL PROCESSING**

**TECHNICAL FIELD**

The present invention relates in general to automated call center operation and, in particular, to a system and method for balancing agent console load during automated call processing.

**BACKGROUND ART**

Customer call centers, or simply, “call centers,” are often the first point of contact for customers seeking direct assistance from manufacturers and service vendors. Call centers are reachable by telephone, including data network-based telephone services, such as Voice-Over-Internet (VoIP), and provide customer support and problem resolution. Although Worldwide Web- and email-based customer support are becoming increasingly available, call centers still offer a convenient and universally-accessible forum for remote customer assistance.

The timeliness and quality of service provided by call centers is critical to ensuring customer satisfaction, particularly where caller responses are generated through automation. Generally, the expectation level of callers is lower when they are aware that an automated system, rather than a live human agent, is providing assistance. However, customers become less tolerant of delays, particularly when the delays occur before every automated system-generated response. Minimizing delays is crucial, even when caller volume is high.

Caller delay times can decreased by having more agents available to handle customer calls. Each new call can be assigned to an agent preferably in an ordered fashion to allow cooperative and consistent call processing, even where the call volume is high. Each additional call that is assigned to an agent, though, can affect the agent’s workload, which can, in turn, affect the delay times experienced by callers. As a result, simply adding more agents to handle calls can be a naïve and ineffective solution.

Determining true agent work load is contingent upon several independent and wide ranging factors, including agent experience, time of day, caller need, and so forth. Moreover, agent load cannot always be generated as an automated index, as the amount of time spent by an agent on each call is both an objective and subjective inquiry. Objective factors, such as the number of calls currently assigned, are readily determinable, but subjective timing considerations based on individual agent control, such as determining how soon an agent may be able to take additional callers, can complicate caller assignment.

CSCD007-1CA

Therefore, there is a need for an approach to efficiently assigning calls to agent consoles based on continually variable independent and agent discretionary conditions.

#### DISCLOSURE OF THE INVENTION

Fixed limits can be applied to each agent that provides service provisioning to individual callers to a call center through an agent console. The balancing of caller session assignments to agent consoles is continually evaluated by a centralized message server based upon an analysis of objective and subjective customer calling conditions against the fixed limits, if applicable. Those conditions that are objectively determinable are generally known to each agent console and the message server throughout the duration of the caller sessions. However, subjective conditions, based on the discretionary assessment of each responsible agent, are only available when provided by the agent. The subjective conditions can be weighed along with other available objective conditions into a normalized determination of agent session workload.

A further embodiment is a system and method for balancing agent console load during automated call processing. A plurality of agent consoles are operated and a call session of a caller calling at each agent console is processed. A fixed limit of call sessions is assigned to each agent console. A call session load on each agent console is balanced. Conditions from each agent console are continually collected. The conditions are evaluated against the fixed limit for each agent console. A load factor is determined as a function of the conditions. The call sessions are assigned to the agent consoles commensurate to the corresponding load factor.

A still further embodiment is a system and method for balancing agent console load during automated call processing. A plurality of agent consoles are operated in a call center. Progress of a call session of a caller that is calling into the call center at each agent console is monitored. An absolute upper limit on a number of the call sessions allowed at each agent console at any given time is assigned. A load of call session assignments on each agent console is balanced. Objective and subjective conditions pertinent to each assigned call session from each agent console are continually collected. The objective and subjective conditions that affect call session availability for each agent console are updated. The objective and subjective conditions are evaluated against the absolute upper limit to determine a load factor. Each call session is assigned to one such agent console within bound applicable to the load factor.

Still other embodiments of the present invention will become readily apparent to those skilled in the art from the following detailed description, wherein is described embodiments of the invention by way of illustrating the best mode contemplated for carrying out the invention. Accordingly, the drawings and detailed description are to be regarded as illustrative in nature and not as restrictive.

CSCD007-1CA

DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a functional block diagram showing an automated call center operational environment.

FIGURE 2 is a functional block diagram showing the automated call center of FIGURE 1.

FIGURE 3 is a process flow diagram showing, by way of example, a typical call sequence, as transacted in the automated call center of FIGURE 2.

FIGURE 4 is a screen diagram showing, by way of example, a view of an agent console for a live call session.

FIGURE 5 is a process flow diagram showing a method for balancing agent console load during automated call processing, in accordance with one embodiment.

FIGURE 6 is a block diagram showing, by way of example, conditions evaluated during agent console load balancing.

FIGURE 7 is a flow diagram showing a routine for load factor determination.

FIGURE 8 is a block diagram showing a system for balancing agent console load during automated call processing, in accordance with one embodiment.

FIGURE 9 is a flow diagram showing a routine for load factor determination with starvation control, in accordance with a further embodiment.

BEST MODE FOR CARRYING OUT THE INVENTION

Automated call centers are a form of voice-response system, which rely upon programmed control to guide callers through a machine-generated dialogue for receiving support and problem resolution assistance. FIGURE 1 is a functional block diagram showing an automated call center operational environment 10. By way of example, a multiplicity of callers call into an automated call center 11, generally through telephonic means, which can include conventional telephone handsets 13a-c through Plain Old Telephone Service (POTS) 12, portable handsets 15a-c through cellular and satellite telephone service 14, VoIP clients 17a-b, and Internet telephony clients 18a-b, as well as other forms and combinations of telephony, message, and voice-based communications. For example, a telephone caller may provide responses to the automated call center 11 through a combination of voice, dual-tone multi-frequency (DTMF), or text messaging. Callers can also “call” or interface into the automated call center 11 using conventional network clients 19 through an internetwork 17, including the Internet. Other types of automated call center access are possible.

Except as otherwise stated, as used herein, the terms “caller,” “user,” and “customer” are used interchangeably to refer to a caller to the automated call center 11. Similarly, the terms “agent,” “guide,” and “operator” are used interchangeably to refer to an agent that provides

CSCD007-1CA

service provisioning to the caller for the automated call center 11. Additionally, although the automated call center 11 is shown as a single point within the automated call center operation environment 10, the automated call center 11 could include two or more logically interconnected but physically separate, including geographically removed, call center operations, which collectively provide a unified automated call center. Other automated call center arrangements and configurations are possible.

Generally, an automated call center provides a single dial-in source for support and problem resolution assistance for customers seeking direct help from manufacturers and service vendors. However, automated call centers 11 can also be applied to many other diverse areas of commerce, such as mail order sales and roadside assistance dispatch. FIGURE 2 is a functional block diagram showing an automated call center 30 for use in the automated call center operational environment 10 of FIGURE 1. The automated call center 30 includes one or more servers 31, 34 and agent consoles 32a-c operatively interconnected over an internal network 33. The agent consoles 32a-c could also be interconnected to the message server 31 over an external network infrastructure 33, such as the Internet or a non-public enterprise data network. The network infrastructure 33 can be either wired or wireless and, in one embodiment, is implemented based on the Transmission Control Protocol/Internet Protocol (TCP/IP) network communications specification, although other types or combinations of networking implementations are possible. Similarly, other network topologies and arrangements are possible.

A message server 31 provides the primary message-based communications infrastructure for the call center operation, such as described in commonly-assigned U.S. Patent Publication No. 2003/0177009, published September 18, 2003 to Odinak et al., and U.S. Patent Publication No. 2005/0177368, published August 11, 2005 to Odinak. During regular operation, the message server 31 executes multiple threads to process multiple simultaneous calls, which are handled by agents executing agent applications on agent consoles 32a-c.

Customer calls are received through a telephony interface 35, which is operatively coupled to the message server 31 to provide access to a telephone voice and data network 36. In one embodiment, the telephony interface connects to the telephone network 36 over a T-1 carrier line, which can provide individual channels of voice or data traffic. Other types of telephone network connections are possible.

The automated call center 30 also includes a recognition server 34 that interfaces directly to the message server 31 as a top-level or root tier of a speech recognition hierarchy. The message server 31 sends streamed audio data for each user call to the recognition server 34, which then performs distributed speech recognition. The message server 31 assigns grammar

CSCD007-1CA

generation and speech recognition to the recognition server 34. Upon startup, the telephony gateway 35 opens a T-1 carrier device channel for each available T-1 time slot. The telephony gateway 35 initiates a new connection to the message server 31, one connection per T-1 device channel, and the message server 31, in turn, initiates a corresponding new connection to the recognition server 34.

The separate telephony gateway-to-message server and message server-to-main recognizer connections form one concurrent session apiece. When a customer call is answered or connected, the telephony gateway 35 sends a call message to the message server 31. The message server 31 then sends a new call message to the recognition server 34.

The message server 31 also assigns each new call to one of the agent consoles 32a-c through dynamic load balancing, such as further described below beginning with reference to FIGURE 5. Each agent console 32a-c provides the primary means for direct customer interaction by providing service provisioning and related assistance to callers, such as further described below with reference to FIGURE 3. The main purpose of each agent console 32a-c is to execute one or more agent applications, which display both user and agent messages and provide menus of actions that can be executed in response to agent commands, including script execution. One or more agent applications execute on each agent console and one or more agent consoles can execute in parallel. Alternatively, multiple instances of agent applications can run on a server machine and can be accessed by agents at agent consoles operating as remote terminals.

Each component, including the message server 31, recognition server 34, and agent consoles 32a-c, is implemented as a computer program, procedure or module, or state transition written as source code or flow specification in a conventional programming language, such as the C++ programming language or modeling tool, and presented for execution by a computer system as object or byte code. Alternatively, the components could be directly implemented in hardware, either as integrated circuitry or burned into read-only memory components. The various implementations of the source code and object and byte codes can be held on a computer-readable storage medium.

Callers to the automated call center interface indirectly with an agent, who multitasks through one or more individual caller sessions via an agent console 32a-c. The callers, however, only perceive an automated calling interface heard, for instance, through their telephone or portable handset. Calls are processed through a sequence of phases. FIGURE 3 is a process flow diagram showing, by way of example, a typical call sequence 40, as transacted by the automated call center 30 of FIGURE 2. Although the form of call processing required may vary

CSCD007-1CA

by subject matter area and other factors, the same overall sequence of caller-to-agent interchanges will loosely apply.

Initially, upon calling into the automated call center 30, each user receives an initial greeting and optionally informational message 41 providing a synopsis of caller options. When provided, the caller options enable the user to navigate through to a specific topic area for assistance or support. However, the user could simply proceed directly into a customer support scenario 42 with either an agent, which is a live person; with a guide, which is an automated prompt that is under the supervision of a live person; or through automated voice response to enable information collection, problem trouble-shooting, and other actions. When present, the guide can listen to and revise caller replies in the background, generally unbeknownst to the caller, such as described in commonly-assigned U.S. Patent Application Publication No. 2003/0177009, entitled "System and Method for Providing a Message-Based Communications Infrastructure for Automated Call Center Operation," published September 18, 2003, pending.

The scenario 42 can be delayed by agent unavailability, caller volume capacity limits, and other factors that can delay providing a response to the caller. Throughout the session, the agent can adjust and customize the flow of the call processing by evaluating caller attributes, such as further described in commonly-assigned U.S. Patent Application Publication No. 2008/0043983, entitled "System and Method for Managing a Dynamic Call Flow During Automated Call Processing," published February 21, 2008, pending. In addition, the order in which information is gathered from the user to assist with problem resolution, such as identifying the caller's needs or troubleshooting a problem, can be dynamically evaluated and controlled, such as further described in commonly-assigned U.S. Patent Application No. 2008/0043982, entitled "System and Method for Processing Out-Of-Order Caller Responses During Automated Call Processing," published February 21, 2008, pending.

As required, service provisioning 43 is provided to the user, either directly during the course of the call or indirectly through a service request dispatch. The type of service provided can include sending information or products, entering a subscription or enrollment, or other forms of commerce or assistance that can be remotely provided over the telephone. Finally, the call ends in a wrap-up 44, which provides closure to the call and a departing salutation. Other forms and variations of customer call sequences are feasible. Importantly, however, from the perspective of the caller, the experience may appear to be an interaction with an intelligent machine and the caller would thus be aware that the agent is automated, not human. Accordingly, the caller would have a more relaxed expectation of agent responsiveness since a machine, and not an actual person, is on the line.

CSCD007-1CA

Following call termination, the automated call center 30 can perform post-call processing 45, which identifies individual speech utterances in each call that can be analyzed or stored into a database 34. The identified speech utterances can be presented to an agent for manipulation, such as ranking or reordering. In addition, the post-call processing 45 can include performing speech recognition on the speech utterances, identifying speaker characteristics, and marking certain speech utterances for later use. In a further embodiment, the automated call center 30 can also perform in-progress call processing 46, which operates on a stream of speech utterances copied from on-going calls. The same types of processing operations can be performed as on completed calls, but allows an agent to perform real time call analysis.

Call sessions are displayed on each agent console through a graphical user interface (GUI). FIGURE 4 is a screen diagram showing, by way of example, a view of an agent console 50 for a live call session. Multiple call sessions can be simultaneously displayed and managed through the GUI controls.

The particulars concerning the live call session are obtained by the message server 31 and provided to the agent. The session particulars can include, for instance, caller profile 51, address 52, contact information 53, reason for call 54, and credit card data 55. Other types of session particulars are possible.

Operationally, the agent can review a recording of the caller's spoken responses through intuitive playback controls 57 and update the display by operating an "Update" control 56. Additionally, the agent can transfer the caller to a live agent by operating a "Transfer to Live Agent" control 58 and can log out by operating a "Logout" control 59. Other GUI controls are possible.

Each call is handled principally by a single agent through an agent console 32a-c. However, prior to undertaking a session with a caller, the message server 31 determines the abilities of a community of agents to handle additional sessions through a continuing evaluation of agent load. FIGURE 5 is a process flow diagram showing a method 70 for balancing agent console load during automated call processing, in accordance with one embodiment. Individual load factors are determined by each agent console 32a-c, while the actual assignment of caller sessions to agent consoles 32a-c is performed by the message server 31. Other delegations of load factor determination and session assignment are possible.

Initially, a new incoming call is received (operation 71) by the message server 31 through the telephony interface 35. The message server 31 continually updates the conditions that can affect the ability of each agent, through their respective agent console 32a-c, to handle additional sessions by evaluating several on-going conditions (operation 72), as further described below with reference to FIGURE 6. A load factor is determined (operation 73) by each agent console

## CSCD007-1CA

32a-c, as further described below with reference to FIGURE 7, and is provided to the message server 31, which will then assign the call (operation 74), typically to the agent console 32a-c with the lowest load factor. However, if no agent is available, that is, every agent console 32a-c has a load factor set to the maximum sessions allowed, the message server 31 will undertake an alternative call disposition by either handling the call without a guide through automated response (operation 76), transferring the call to a live agent, or placing the caller on hold (operation 75). Processing of unassigned calls is repeated for each new caller.

Conditions are evaluated by both the message server 31 and each agent console 32a-c on an on-going basis. FIGURE 6 is a block diagram showing, by way of example, conditions 80 evaluated 81 during agent console load balancing. The particular conditions evaluated can be objective measures that exist independently of agent discretion, and subjective factors, which require individual agent assessment. Objective conditions can be known by the message server 31 and agent console 32a-c alike, whereas subjective conditions are initially known only to the agent until the agent stores or send the subjective conditions to the other systems.

Conditions 81 are evaluated and exchanged continually. The conditions 81 can include the current number of calls 82 being handled by each agent console 32a-c, the state of the call 83, such as "pending resolution" or "awaiting caller reply," limits on maximum number of sessions 84 for a particular agent, priority of caller 85, and group to which the agent is assigned 86. Other conditions 87 include the subjective impression of the difficulty perceived by the agent to resolve a caller's concerns or the difficulty being experienced by the agent in dealing with the personality of a caller. Still other conditions are possible.

A load factor is a numerical metric for approximating the ability of an agent console 32a-c to receive an additional call session assignment. FIGURE 7 is a flow diagram showing a routine 90 for load factor determination. Load factors can be determined by either the message server 31, responsible agent console 32a-c, or other system having access to the conditions 81.

Initially, all of the currently known conditions 81 are assembled by the evaluating system (block 91). Although the conditions 81 are continually collected, those conditions 81, in the process of being tallied, for instance, may be temporarily unavailable at the time of assembly. Each agent console 32a-c is subject to an absolute upper limit on the number of maximum sessions allowed. If the number of sessions for a particular console 32a-c is at the limit (block 92), the load factor is set to an arbitrary maximum value (block 93). Otherwise, if the session limit has not been reached (block 92), a normalized load factor is generated (block 94). Finally, the load factor is returned from the routine (block 95), which includes sending the load factor to the message server 31 if the load factor was determined by the responsible agent console 32a-c.

CSCD007-1CA

A load factor can be determined as a normalized numeric metric on a per console basis. The load factor can be determined whether the maximum number of sessions per console is limited or unspecified. The load factor  $LF_i$  is normalized over the scaling factor  $m$ , such that  $0 \leq LF_i \leq m$ . Where the number of sessions that can be assigned to an agent console is upper bounded, the load factor  $LF_i$  for each agent console  $i$  can be evaluated per equation (1) as:

$$LF_i = \begin{cases} 0, & s_i = 0 \\ \text{ceil} \left( m \frac{\sum_{j=1}^{s_i} t_{ij} s_i}{(wl_i^2)} \right), & s_i < l_i \\ m, & s_i = l_i \end{cases} \quad (1)$$

where  $m$  is a scaling factor;  $s_i$  is the number of sessions currently assigned to an agent console  $i$ ;  $t_{ij}$  is the current state of a session in the agent console  $i$ ;  $l_i$  is the maximum number of sessions allowed for agent the console  $i$ , such that  $l_i \geq 0$ ; and  $w$  is the number of windows that are viewable on the agent console  $i$ , such that  $0 \leq i \leq l_i$ .

In one embodiment,  $m = 400$  and  $w = 4$ , although other values are possible. In addition, the current state  $t_{ij}$  is set to one of three values:

|                     |   |
|---------------------|---|
| waiting (RED):      | 4 |
| attention (YELLOW): | 2 |
| active (GREEN):     | 1 |

Other state values, either in lieu of or in combination with the foregoing state values are possible.

Where the number of sessions that can be assigned to an agent console is not specified, the load factor  $LF_i$  for each agent console  $i$  can be evaluated per equation (2), which is a generalization of equation (1), as:

CSCD007-1CA

$$LF'_i = \begin{cases} 0, & s_i = 0 \\ \text{ceil} \left( m \frac{\sum_{j=1}^{s_i} t_{ij} s_i}{w} \right), & s_i < l_i \\ m, & s_i = l_i \end{cases} \quad (2)$$

Operationally, limiting the maximum number of sessions that can be assigned to an agent console provides load balancing better attuned to current conditions and agent abilities, as the more generalized form of load factoring does not reflect the *capability* of a particular agent console that is implicit in the upper limit. Thus, where equation (2) would indicate that the agent console having the lowest load factor should receive the next caller assignment, equation (1) would qualify the recommendation based upon the relative ability of that agent console to handle an additional caller assignment. Consequently, a first agent console with more assigned callers may still be preferred over another agent console with few assigned callers if the maximum number of sessions for that first console is higher than the other agent console's limit.

Generally, each agent console 32a-c has access to the most up-to-date information regarding the conditions pertaining to load factor determination. FIGURE 8 is a block diagram showing a system 100 for balancing agent console load during automated call processing, in accordance with one embodiment. The message server 31, or other agent console 32a-c, could also determine load factors if given access to the appropriate data. The system 100 operates in accordance with a sequence of process steps, as further described above with reference to FIGURE 5.

Each agent console 101 processes a stream of caller messages 107 and generates a stream of agent responses 108, as well as a load factor 109, which is provided to the message server 31 periodically or upon demand. The console 101 includes a call processing module 102, which processes the message stream 107 to provide service provisioning 43 (shown in FIGURE 3) and provides other caller interactions.

The console 101 also includes a load monitor 103, which continually assembles a set of conditions 105, which are stored in a storage device 104. The load monitor 103 determines the

CSCD007-1CA

load factor 109 by evaluating the conditions 105 against fixed limits 106 that are assigned to the agent and also stored in the storage device 104. Other console operations are possible.

Ordinarily, if no agent is available, the message server 31 will undertake an alternative call disposition, as described above with reference to FIGURE 5. However, a starvation condition can occur when all except one of the agent consoles 32a-c are unavailable due to maximum load factors and the message server 31 is unable to either assign a new call to the last remaining agent console 32a-c, transfer the call, or place the caller on hold. The starvation condition arises where the last remaining agent console 32a-c, which also has the current lowest load factor, has just accepted a call and is temporarily not accepting new calls. The agent console's temporary unavailability results in a deadlock, as the message server 31 still assumes that the agent console can accept calls and will not attempt another call disposition. As a result, starvation occurs.

The starvation condition can be avoided by updating the load factor to reflect agent console availability. FIGURE 9 is a flow diagram showing a routine 110 for load factor determination with starvation control, in accordance with a further embodiment. The load factor is set to the maximum value immediately after each agent console 32a-c has accepted a new call and remains at the maximum value until the agent console is again accepting new calls, after which the load factor is normally determined.

Initially, all of the currently known conditions 81 are assembled by the evaluating system (block 111). If the agent console 32a-c has just accepted a new call (block 112), but has not asked for a new call (block 113), that is, the agent console is temporarily unavailable, the load factor is set to an arbitrary maximum value (block 115). Similarly, if the agent console has not accepted a new call (block 112) but the number of sessions for that console is at the limit (block 114), the load factor is also set to the arbitrary maximum value (block 115). Otherwise, a normalized load factor is generated (block 116) if either the session limit has not been reached (block 114) or the agent console has asked for a new call (block 116). Finally, the load factor is returned from the routine (block 117), which includes sending the load factor to the message server 31 if the load factor was determined by the responsible agent console 32a-c.

CSCD007-1CA

## CLAIMS:

1. A system for balancing agent console load during automated call processing, comprising:  
a console to operate a plurality of agent consoles;  
a call processor to process a plurality of call sessions; and  
a message server to balance a call session load on each agent console, comprising:

a load balance module to continually collect conditions from each agent console, to evaluate the conditions against each agent console, and to determine a load factor as a function of the conditions, wherein the load factor represents a normalized numeric metric  $0 \leq LF_i \leq m$  determined in accordance with the formula:

$$LF'_i = \begin{cases} 0, & s_i = 0 \\ \text{ceil} \left( m \frac{\sum_{j=1}^{s_i} t_{ij} s_i}{w} \right), & s_i < l_i \\ m, & s_i = l_i \end{cases}$$

where  $m$  is a scaling factor,  $s_i$  is a number of sessions currently assigned to one such agent console  $i$ ,  $t_{ij}$  is a current state of a session in that agent console  $i$ , and  $w$  is a number of windows that are viewable on that agent console  $i$ , such that  $0 \leq i \leq l_i$ ; and

a call distributor to assign the call sessions to the agent consoles commensurate to the corresponding load factor.

2. A system according to Claim 1, wherein the conditions comprise at least one of a state of a call previously assigned, a limit on a maximum number of call sessions, a priority of the call previously assigned, and a group to which an agent is assigned.

CSCD007-1CA

3. A system according to Claim 1, wherein the conditions comprise values for at least one of a difficulty resolving the caller's concerns and a difficulty experienced by an agent in dealing with the caller.
4. A system according to Claim 1, wherein  $t_{ij}$  is set for the current state comprising at least one of a waiting, attention, and active state.
5. A system according to Claim 1, wherein the call distributor performs an alternative call disposition when each agent console is unavailable to take the assigned call comprising at least one of handling the assigned call through automated response, transferring the assigned call to a live agent, and placing the assigned call on hold.
6. A system according to Claim 1, wherein the load balance module assigns a maximum load factor to the agent console that receives the assigned call and maintains the maximum load factor until the agent console is ready to accept new calls.
7. A method for balancing agent console load during automated call processing, comprising:
  - operating a plurality of agent consoles;
  - processing a plurality of call sessions; and
  - balancing a call session load on each agent console, comprising:
    - continually collecting conditions from each agent console;
    - evaluating the conditions against each agent console and determining a load factor as a function of the conditions to represent a normalized numeric metric  $0 \leq LF'_i \leq m$  in accordance with the formula:

CSCD007-1CA

$$LF'_i = \begin{cases} 0, & s_i = 0 \\ \text{ceil} \left( m \frac{\sum_{j=1}^{s_i} t_{ij} s_i}{w} \right), & s_i < l_i \\ m, & s_i = l_i \end{cases}$$

where  $m$  is a scaling factor,  $s_i$  is a number of sessions currently assigned to one such agent console  $i$ ,  $t_{ij}$  is a current state of a session in that agent console  $i$ , and  $w$  is a number of windows that are viewable on that agent console  $i$ , such that  $0 \leq i \leq l_i$ ; and

assigning the call sessions to the agent consoles commensurate to the corresponding load factor.

8. A method according to Claim 7, wherein the conditions comprise at least one of a state of a call previously assigned, a limit on a maximum number of call sessions, a priority of the call previously assigned, and a group to which an agent is assigned.
9. A method according to Claim 7, wherein the conditions comprise values for at least one of a difficulty resolving the caller's concerns and a difficulty experienced by an agent in dealing with the caller.
10. A method according to Claim 7, wherein  $t_{ij}$  is set for the current state comprising at least one of a waiting, attention, and active state.
11. A method according to Claim 7, further comprising:  
performing an alternative call disposition when each agent console is unavailable to take the assigned call, comprising at least one of:

handling the assigned call through automated response;

CSCD007-1CA

transferring the assigned call to a live agent; and

placing the assigned call on hold.

12. A method according to Claim 7, further comprising:  
 assigning a maximum load factor to the agent console that receives the assigned call; and  
 maintaining the maximum load factor until the agent console is ready to accept new calls
13. A system for balancing agent console load during automated call processing, comprising:  
 a console to operate a plurality of agent consoles;  
 a call processor to process a plurality of call sessions; and  
 a message server to balance a call session load on each agent console, comprising:

a load balance module to assign a fixed limit to the call sessions for each agent console, to continually collect conditions from each agent console, to evaluate the conditions against each agent console, and to determine a load factor as a function of the conditions, wherein the load factor represents a normalized numeric metric  $0 \leq LF_i \leq m$  determined in accordance with the formula:

$$LF_i = \begin{cases} 0, & s_i = 0 \\ \text{ceil} \left( m \frac{\sum_{j=1}^{s_i} t_{ij} s_i}{(w l_i^2)} \right), & s_i < l_i \\ m, & s_i = l_i \end{cases}$$

where  $m$  is a scaling factor,  $s_i$  is a number of sessions currently assigned to one such agent console  $i$ ,  $t_{ij}$  is a current state of a session in that agent console  $i$ ,  $l_i$  is the fixed

CSCD007-1CA

limit for that agent console  $i$ , such that  $l_i \geq 0$ , and  $w$  is a number of windows that are viewable on that agent console  $i$ , such that  $0 \leq i \leq l_i$ ; and

a call distributor to assign the call sessions to the agent consoles commensurate to the corresponding load factor.

14. A system according to Claim 13, wherein  $t_{ij}$  is set to a value for the current state comprising at least one of a waiting, attention, and active state.
15. A system according to Claim 13, wherein the conditions comprise at least one of a state of a call previously assigned, a limit on a maximum number of call sessions, a priority of the call previously assigned, and a group to which an agent is assigned.
16. A system according to Claim 13, wherein the conditions comprise values for at least one of a difficulty resolving a caller's concerns and a difficulty experienced by an agent in dealing with the caller.
17. A system according to Claim 13, wherein the call distributor performs an alternative call disposition when each agent console is unavailable to take the assigned call sessions comprising at least one of handling one or more of the assigned call sessions through automated response, transferring one or more of the assigned call sessions to a live agent, and placing one or more of the assigned call sessions on hold.
18. A system according to Claim 13, wherein the load balance module assigns a maximum load factor to the agent console that receives one of the assigned call sessions and maintains the maximum load factor until the agent console is ready to accept new calls.
19. A method for balancing agent console load during automated call processing, comprising:
  - operating a plurality of agent consoles;
  - processing a plurality of call sessions; and
  - balancing a call session load on each agent console, comprising:
    - assigning a fixed limit to the call sessions for each agent console;
    - continually collecting conditions from each agent console;

CSCD007-1CA

evaluating the conditions against the fixed limit for each agent console and determining a load factor as a function of the conditions to represent a normalized numeric metric  $0 \leq LF_i \leq m$  in accordance with the formula:

$$LF_i = \begin{cases} 0, & s_i = 0 \\ \text{ceil} \left( m \frac{\sum_{j=1}^{s_i} t_{ij} s_i}{(w l_i^2)} \right), & s_i < l_i \\ m, & s_i = l_i \end{cases}$$

where  $m$  is a scaling factor,  $s_i$  is a number of sessions currently assigned to one such agent console  $i$ ,  $t_{ij}$  is a current state of a session in that agent console  $i$ ,  $l_i$  is the fixed limit for that agent console  $i$ , such that  $l_i \geq 0$ , and  $w$  is a number of windows that are viewable on that agent console  $i$ , such that  $0 \leq i \leq l_i$ ; and

assigning the call sessions to the agent consoles commensurate to the corresponding load factor.

20. A method according to Claim 19, wherein  $t_{ij}$  is set to a value for the current state comprising at least one of a waiting, attention, and active state.
21. A method according to Claim 19, wherein the conditions comprise at least one of a state of a call previously assigned, a limit on a maximum number of call sessions, a priority of the call previously assigned, and a group to which an agent is assigned.
22. A method according to Claim 19, wherein the conditions comprise values for at least one of a difficulty resolving a caller's concerns and a difficulty experienced by an agent in dealing with the caller.

CSCD007-1CA

23. A method according to Claim 19, further comprising:  
performing an alternative call disposition when each agent console is unavailable to take the assigned call sessions, comprising at least one of:
- handling one or more of the assigned call sessions through automated response;
  - transferring one or more of the assigned call sessions to a live agent; and
  - placing one or more of the assigned call sessions on hold.
24. A method according to Claim 19, further comprising:  
assigning a maximum load factor to the agent console that receives one of the assigned call sessions; and  
maintaining the maximum load factor until the agent console is ready to accept new calls.

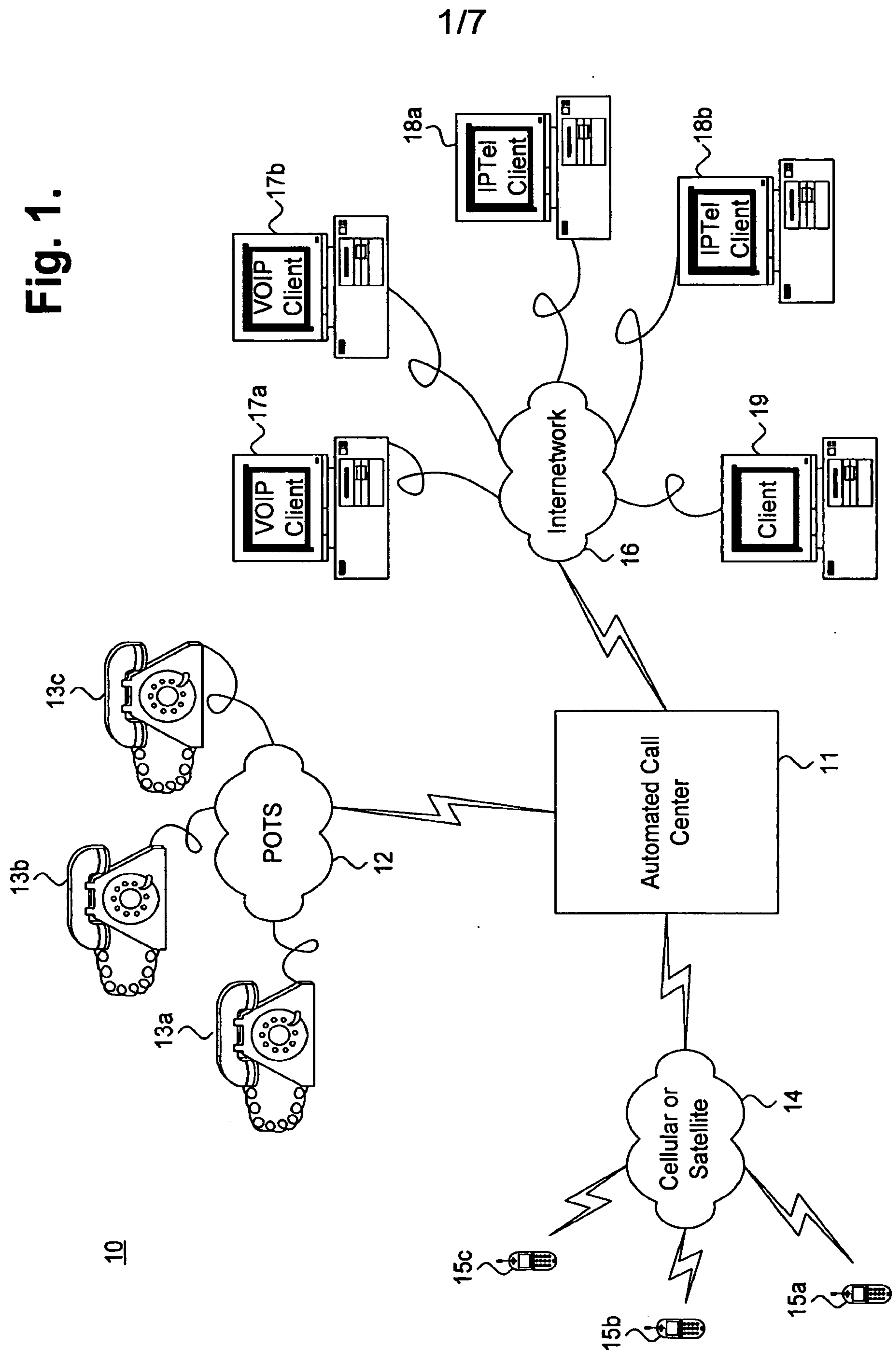
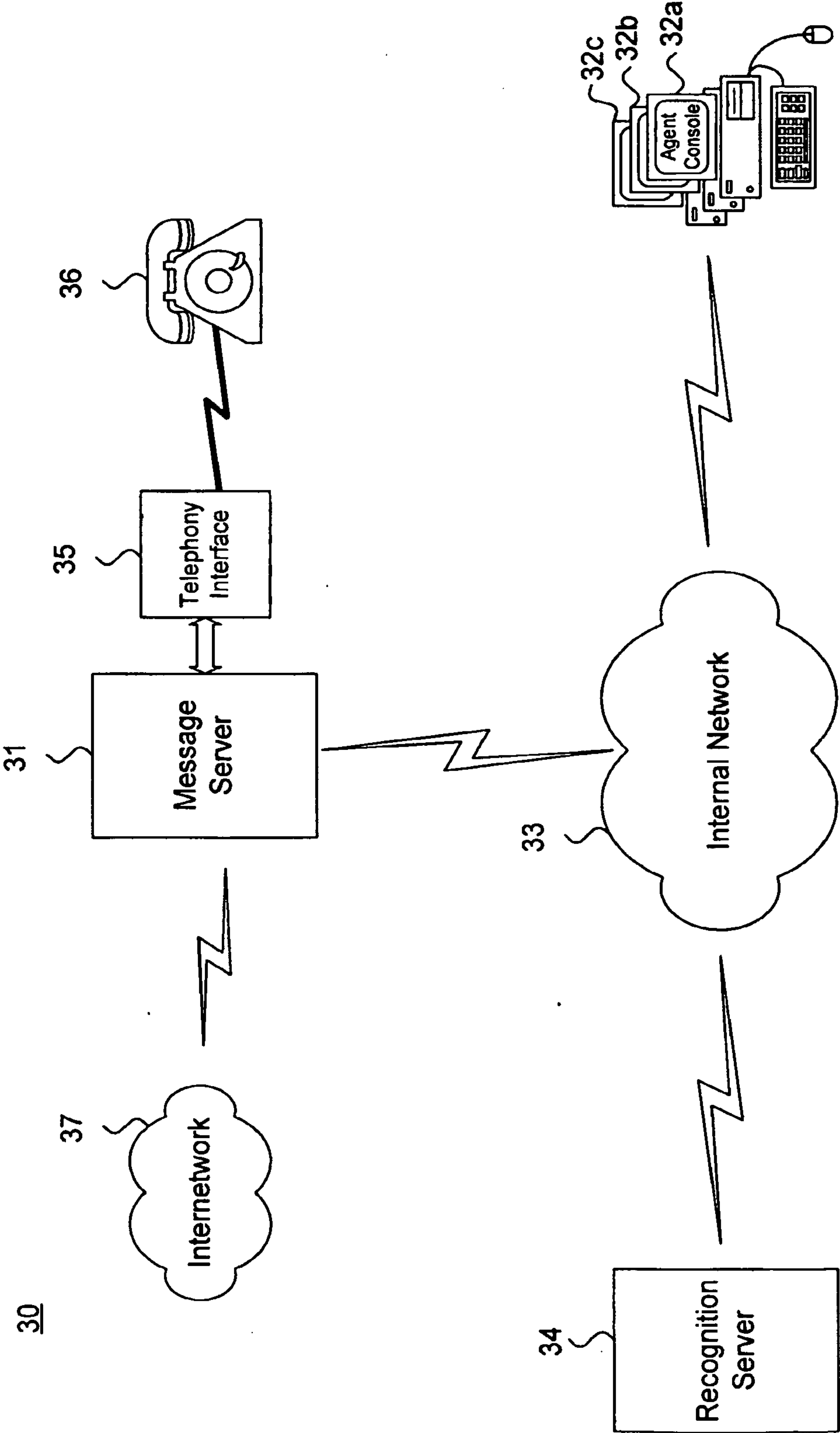


Fig. 2.



3/7

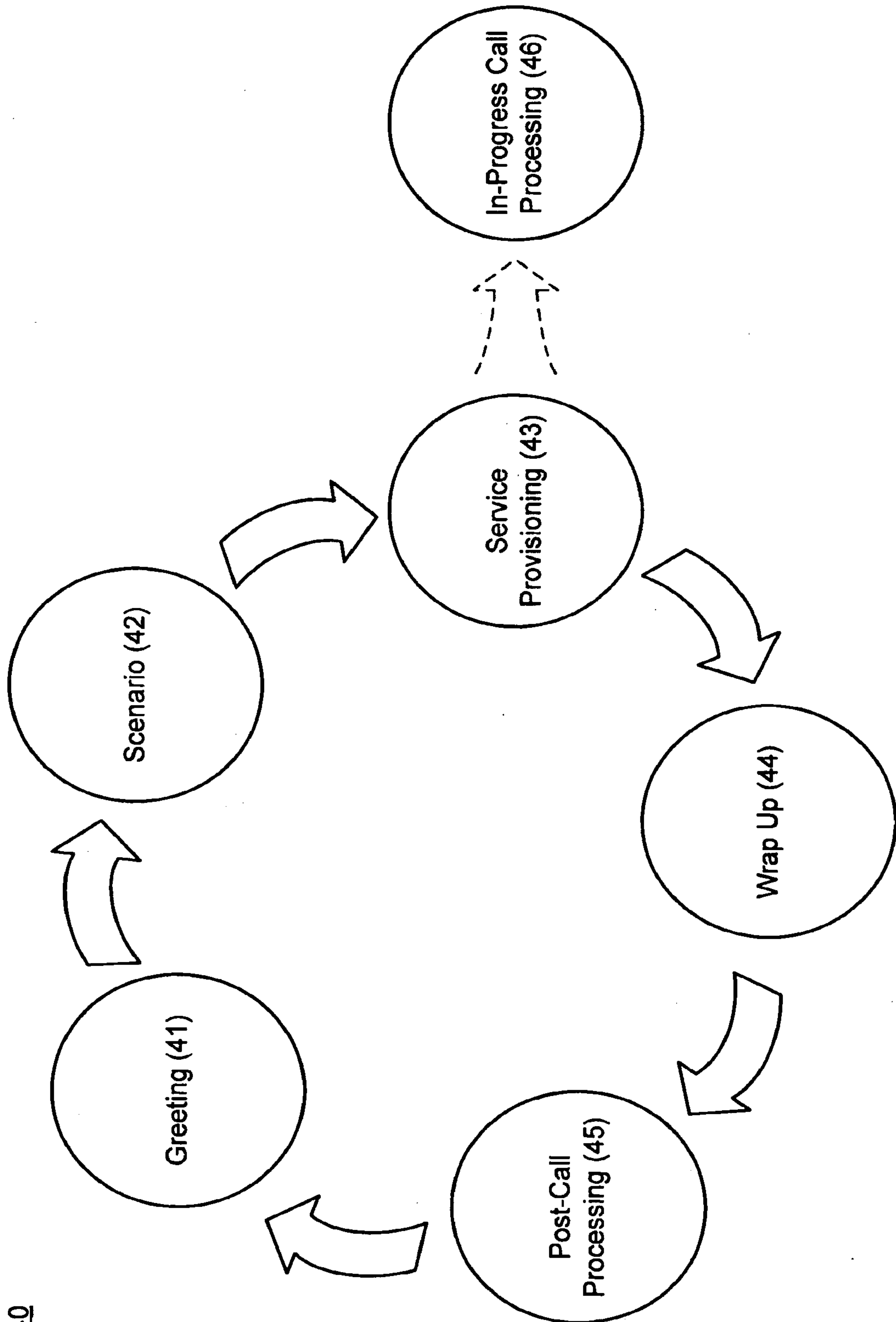
**Fig. 3.**

**Fig. 4.**

4/7

50

50

57 58 59

51 52 53 54 55

BBMFOCOCJAUMGET2CM

CALL TIME 00:01:33

Transfer to Live Agent Logout

**Caller Profile** Listen Update 56

First Name JON

Last Name CONNOR

Zip Code 87654

Existing Customer? no ☐ Yes ☐ No

**Address** Listen Update

House No. 8

Street Name 470<sup>th</sup> PL SE

City NORTH BEND

State WA

**Reason for Call** Listen Update 54

order

☐ Order – something

☐ 1. Order – Superstars of Country

☒ 2. Order – The Best of Luther Vandross

☐ 3. Order – 70s Music Explosion

☐ 4. Order – ELVIS: 30 #1 Hits

☐ 5. Order – Body and Soul

☐ 6. Order – Lifetime of Romance

☐ Cancel Order

☐ Shipping

☐ Return

☐ Bad Product

☐ Others

**Contact Information** 53

Phone Listen Update

Email Listen Update

**Credit Card** Listen Update 55

Type Choose A Type

Number

Exp date

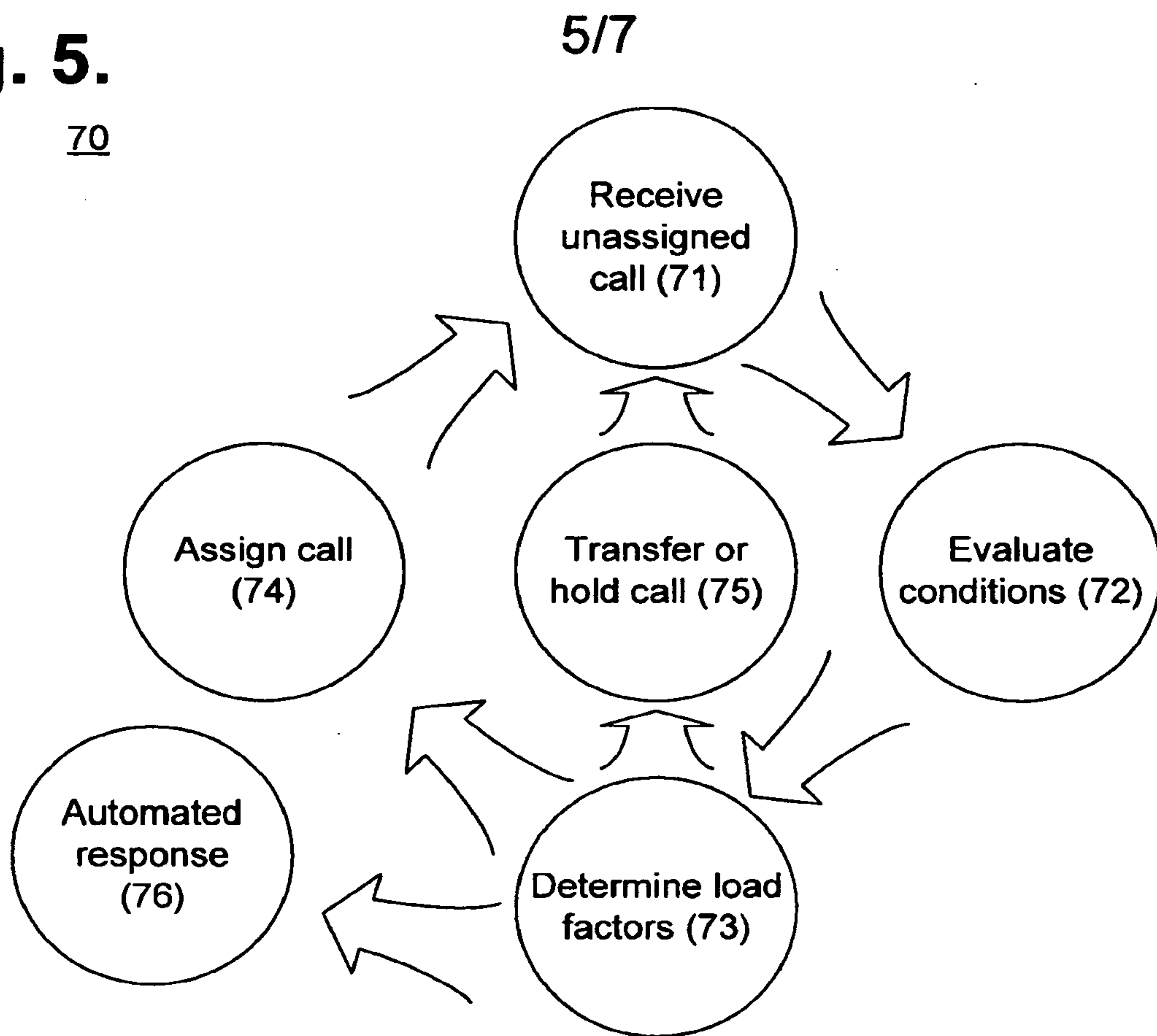
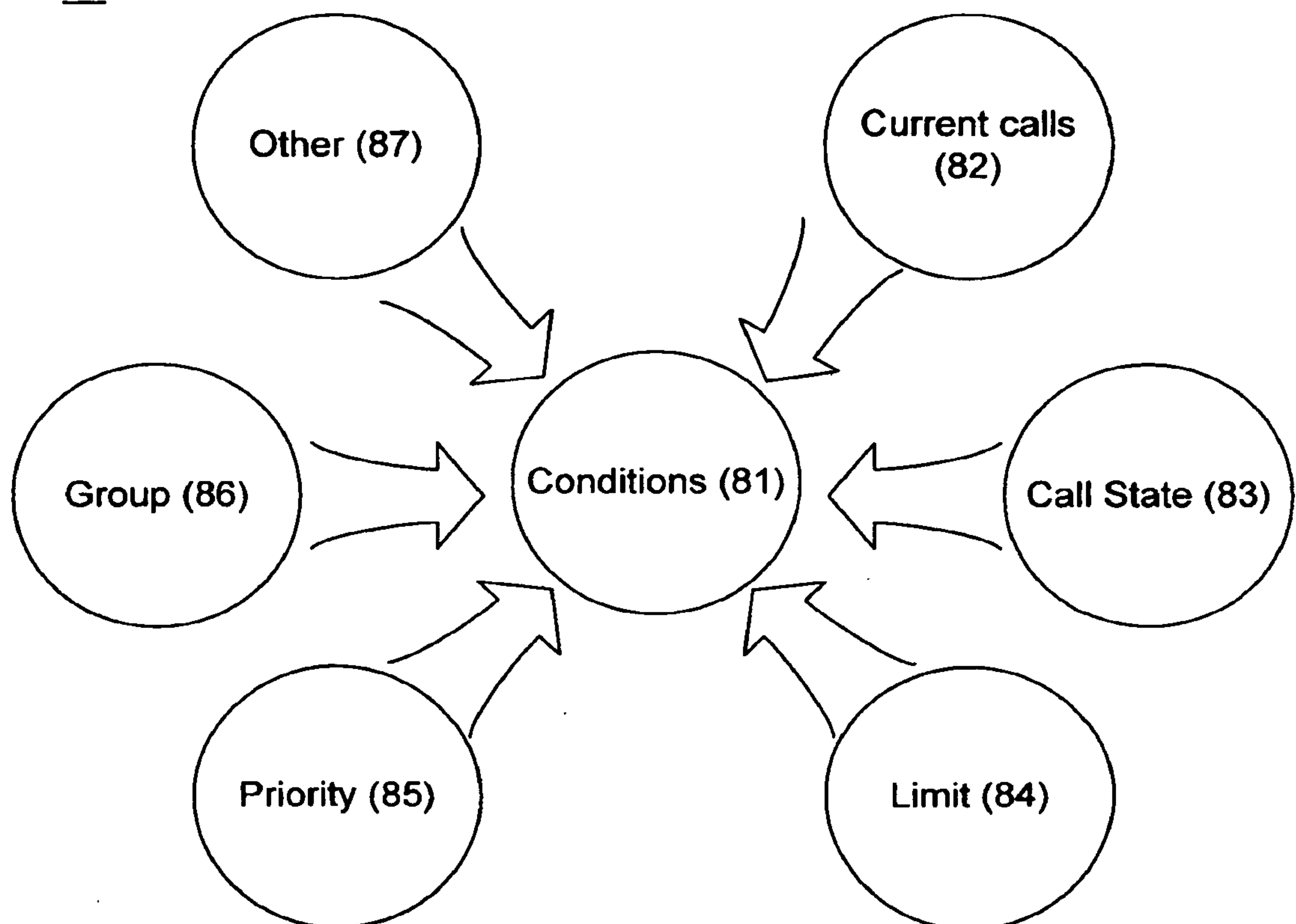
**Fig. 5.**70**Fig. 6.**80

**Fig. 7.**

6/7

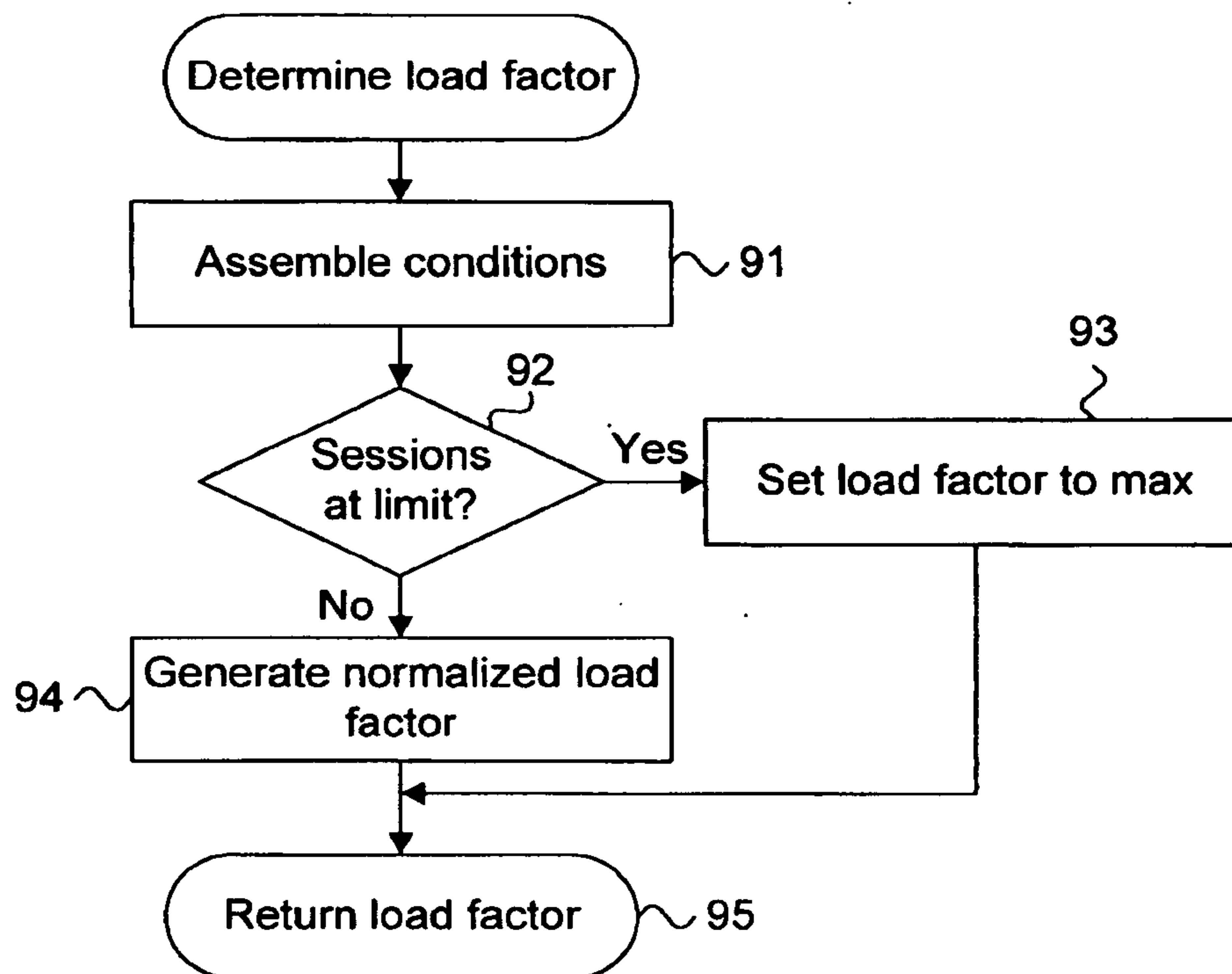
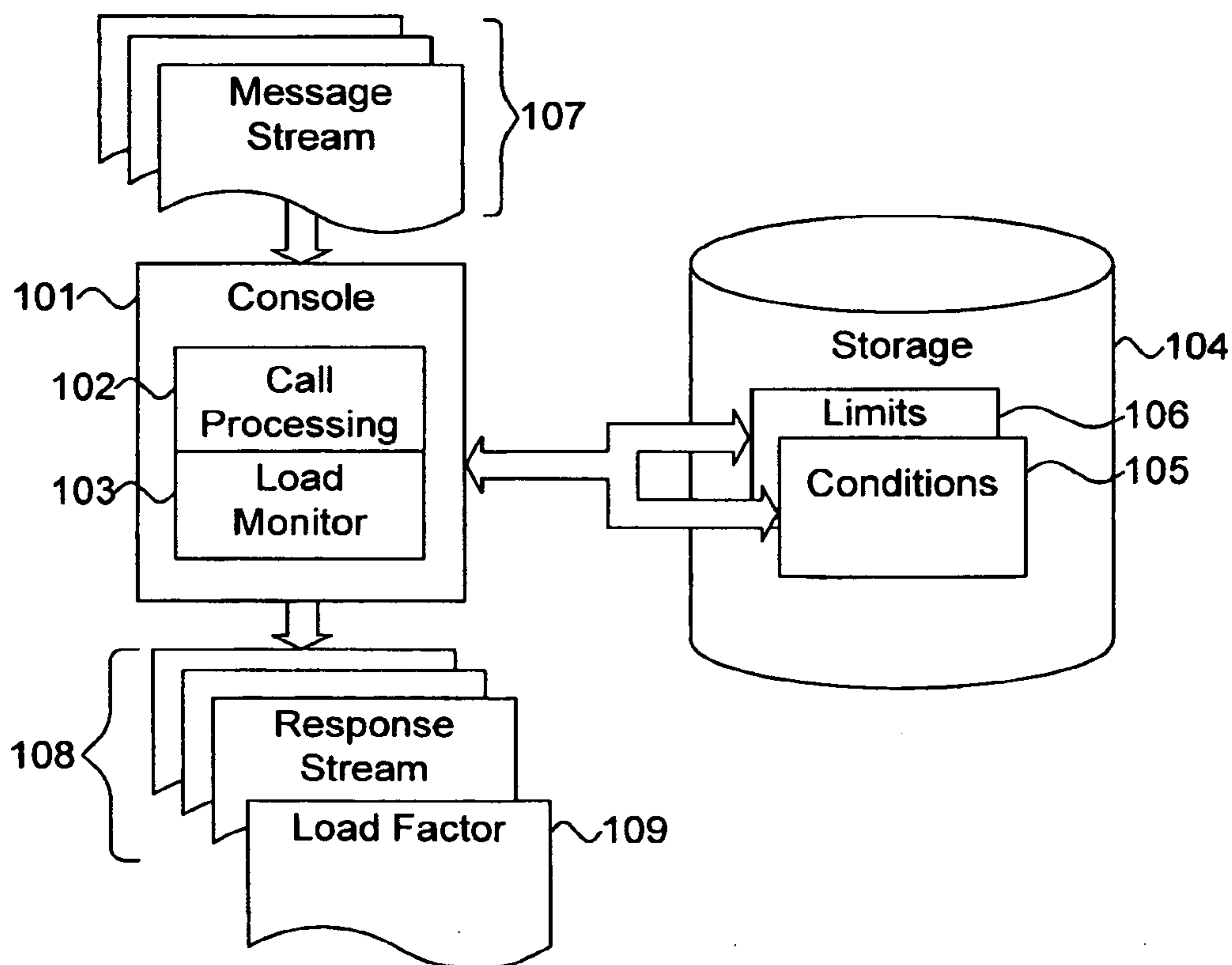
90**Fig. 8.**100

**Fig. 9.**

7/7

110