	assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Therefore, it would be highly desirable to have a system permitting the capability to provide home care and direct information communication to the physician and his or her staff in real time, so as to reduce the recovery period and the risk of complications. Benigno at 6:26-29 The nurse or caregiver then sees the patient almost immediately at home and tracks the patient at home one or more times per day using the system and the information is used to create and update the clinical pathway database records for the patient. Real-time communication systems of the invention allow supervision by the physician, while not requiring the supervision to occur in a hospital setting. Benigno at 9:26-31 The communications subsystems of the invention are important to its capability of providing stable acute care and tracking clinical pathways. Point of service communication at home using either a suitable electronic or computerized device is provided by the invention. The computer can be put into communication with a data storage / server computer via any suitable means, including a modem or network adapter. Benigno at 11:14-19 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the

	actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
(e) removing said remote computer from electronic communication with said first computer;	In step 102, the nurse, 5 using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard

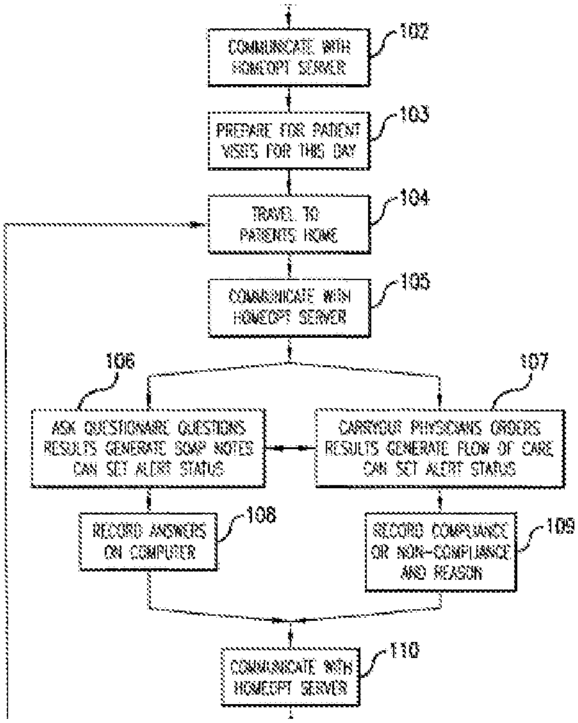
telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc.
Benigno at 47:6-13

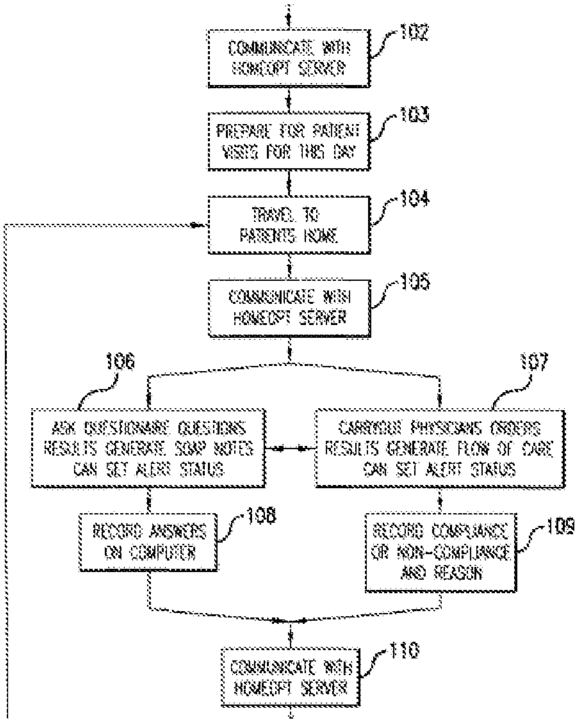
Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402.
Benigno at 47:20-27

In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered.
Benigno at 47:29-30

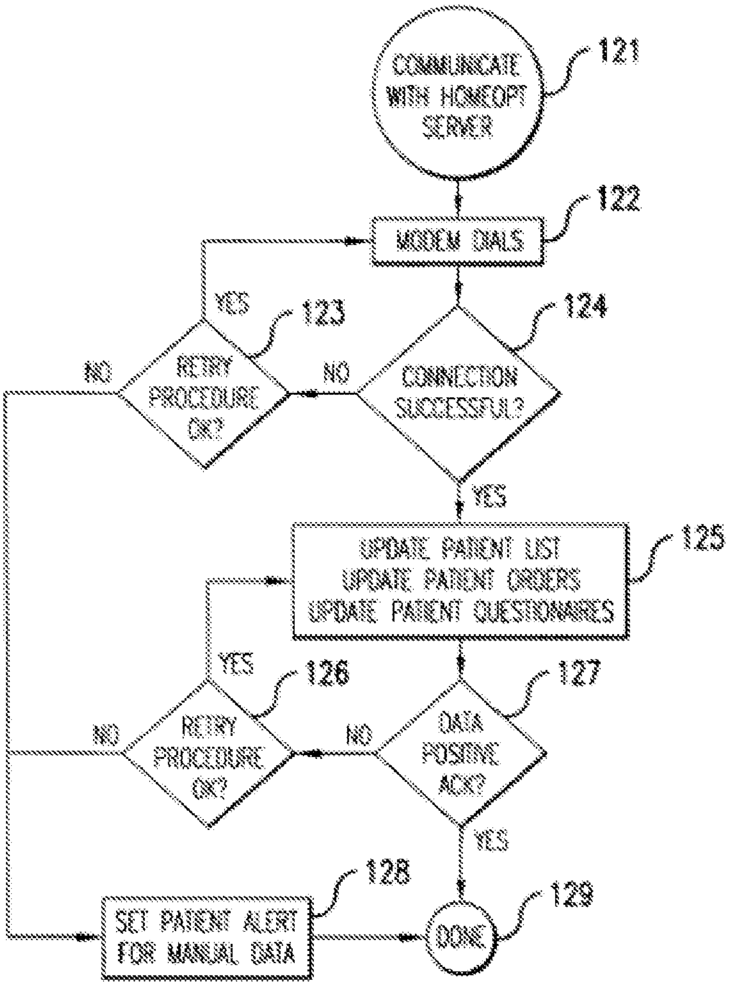
Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than C-Ch active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random.
Rappaport at 7:44-63

In a network that employs an admission control protocol according to the present invention, voice calls, for example (or other time-sensitive stream traffic) may preempt resources of time-insensitive data calls, which result in suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can also be accommodated. Mobile users that

	have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at 2:44-58
(f) within said remote computer, using said transmitted tokenized questionnaire to obtain at least one user response;	In step 106 and 108, the client computer 401, via the questionnaire language previously described, or through any other data collection mechanism, may obtain data from the nurse or other source corresponding to the clinical pathway to be followed, as dictated by the physician. As a result, SOAP notes may be generated, alerts can be generated, etc., for ultimate retransmission to the server 402. Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. The results of such orders may generate a flow of care to be followed by the nurse, and/or may generate alerts, etc. In step 109, the nurse records in the client computer 401 compliance or non-compliance with the orders. If noncompliance, the reasons are also stored. gain, all such stored data may later be transmitted back to the server 402. Benigno at 46:16-28  COMMUNICATE WITH HOMEOPT SERVER 102 PREPARE FOR PATIENT VISITS FOR THIS DAY 103 TRAVEL TO PATIENTS HOME 104 COMMUNICATE WITH HOMEOPT SERVER 105 106 107 ASK QUESTIONAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS RECORD ANSWERS ON COMPUTER 108 RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON 109 COMMUNICATE WITH HOMEOPT SERVER 110

(g) storing said at least one user response within said remote computer;	In step 106 and 108, the client computer 401, via the questionnaire language previously described, or through any other data collection mechanism, may obtain data from the nurse or other source corresponding to the clinical pathway to be followed, as dictated by the physician. As a result, SOAP notes may be generated, alerts can be generated, etc., for ultimate retransmission to the server 402. Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. The results of such orders may generate a flow of care to be followed by the nurse, and/or may generate alerts, etc. In step 109, the nurse records in the client computer 401 compliance or non-compliance with the orders. If noncompliance, the reasons are also stored. gain, all such stored data may later be transmitted back to the server 402. Benigno at 46:16-28  COMMUNICATE WITH HOMEOPT SERVER 102 PREPARE FOR PATIENT VISITS FOR THIS DAY 103 TRAVEL TO PATIENTS HOME 104 COMMUNICATE WITH HOMEOPT SERVER 105 106 107 ASK QUESTIONAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS RECORD ANSWERS ON COMPUTER 108 RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON 109 COMMUNICATE WITH HOMEOPT SERVER 110
(h) modifying said questionnaire with incremental changes at a second computer located at a second site;	Daily communication includes SOAP notes, notification of whether the patient received appropriate IV medications and intravenous fluids, as well as the ability to communicate with nurses, and nurse communications with physicians for order changes. Benigno at 11:22-26 The system is constantly evaluating itself. As the system finds new correlating factors, they are put in place to aid in determining changes to be made to the current or default clinical pathway. In addition, as

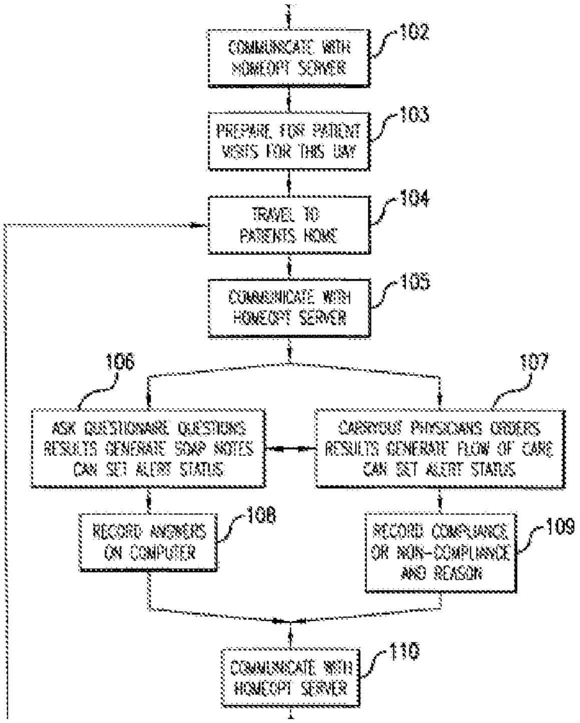
	correlations are determined between clinical pathway decisions and significant outcomes *(i.e.,* outcomes of interest), changes can be made to the default pathway to optimize systematically the clinical pathway toward the desired results. These changes can be automatically made or can be presented to the physician, system administrator, or other user for approval. Benigno at 21:22-29 In addition, the present invention provides a client / server system for manipulation and analysis of data related to clinical pathways, comprising a communication network, a client workstation in communication with the communication network, wherein the client workstation comprises means for generating at least one signal corresponding to a clinical pathway decision and transmitting the at least one decision signal over the communication network, and means for receiving at least one signal corresponding to a clinical pathway modification from the communication network, and means for outputting the at least one modification signal to a signal processing means, a server on the communication network, wherein the server comprises a clinical pathway database for storing an initial procedure decision data element, corresponding to a decision point within the clinical pathway, and at least one subsequent decision data element corresponding to at least one available subsequent decision point within the clinical pathway, and a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element, and processing means, in communication with the communication network, the client workstation, and the server, for performing the steps of receiving the at least one decision signal from the communication network, based on the received decision signal, selecting one of the at least one subsequent decision data elements, comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database, and based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database, then generating at least one signal corresponding to a clinical pathway modification of the subsequent decision data elements in the clinical pathway database, and transmitting the at least one clinical pathway modification signal over the communication network to the receiving means of the client workstation. Benigno at 14:7-15:2

	 COMMUNICATE WITH HOMEOPT SERVER 121 MODEM DIALS 122 YES 123 124 NO RETRY PROCEDURE OK? NO CONNECTION SUCCESSFUL? YES UPDATE PATIENT LIST UPDATE PATIENT ORDERS UPDATE PATIENT QUESTIONAIRES 125 YES 126 127 NO RETRY PROCEDURE OK? NO DATA POSITIVE ACK? YES 128 129 SET PATIENT ALERT FOR MANUAL DATA DONE Benigno at FIG. 1B
(i) placing said remote computer into electrical communication with said second computer;	In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out

orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401.
Benigno at 46:22-24

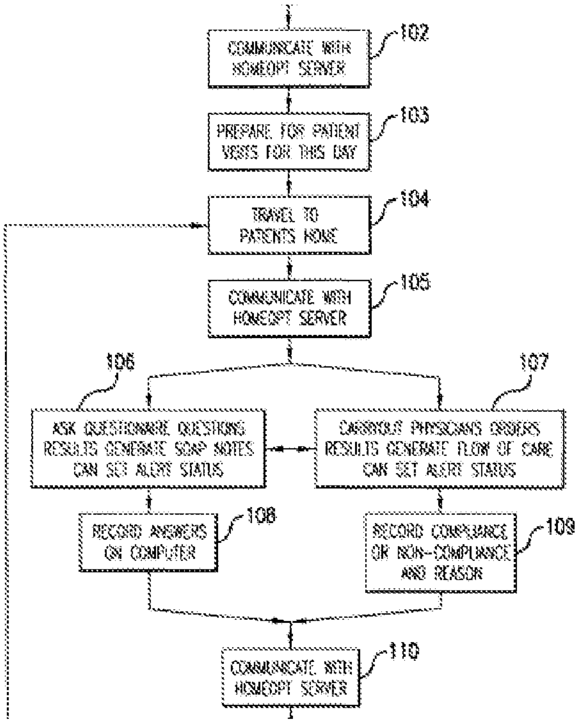
In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached.
Benigno at 46:30-47:4

Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc.
Benigno at 47:6-13


COMMUNICATE WITH HOMEOPT SERVER
102
PREPARE FOR PATIENT VISITS FOR THIS DAY
103
TRAVEL TO PATIENTS HOME
104
COMMUNICATE WITH HOMEOPT SERVER
105
106
107
ASK QUESTIONAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS
CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS
RECORD ANSWERS ON COMPUTER
108
RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON
109
COMMUNICATE WITH HOMEOPT SERVER
110

	Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
(j) transmitting said incremental changes from said second computer to said remote computer;	Therefore, it would be highly desirable to have a system permitting the capability to provide home care and direct information communication to the physician and his or her staff in real time, so as to reduce the recovery period and the risk of complications. Benigno at 6:26-29 The nurse or caregiver then sees the patient almost immediately at home and tracks the patient at home one or more times per day using the system and the information is used to create and update the clinical pathway database records for the patient. Real-time communication systems of the invention allow supervision by the physician, while not requiring the supervision to occur in a hospital setting. Benigno at 9:26-31 The communications subsystems of the invention are important to its capability of providing stable acute care and tracking clinical pathways. Point of service communication at home using either a suitable electronic or computerized device is provided by the invention. The computer can be put into communication with a data storage / server computer via any suitable means, including a modem or network adapter. Benigno at 11:14-19 In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link.

	Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is

	encountered. Benigno at 47:29-30  COMMUNICATE WITH HOMEOPT SERVER 102 PREPARE FOR PATIENT VISITS FOR THIS DAY 103 TRAVEL TO PATIENTS HOME 104 COMMUNICATE WITH HOMEOPT SERVER 105 106 107 ASK QUESTIONNAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS RECORD ANSWERS ON COMPUTER 108 RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON 109 COMMUNICATE WITH HOMEOPT SERVER 110 Benigno at FIG 1A
(k) modifying said transmitted tokenized questionnaire in said remote computer with said incremental changes, thereby creating a modified tokenized questionnaire;	Daily communication includes SOAP notes, notification of whether the patient received appropriate IV medications and intravenous fluids, as well as the ability to communicate with nurses, and nurse communications with physicians for order changes. Benigno at 11:22-26 The system is constantly evaluating itself. As the system finds new correlating factors, they are put in place to aid in determining changes to be made to the current or default clinical pathway. In addition, as correlations are determined between clinical pathway decisions and significant outcomes *(i.e.,* outcomes of interest), changes can be made to the default pathway to optimize systematically the clinical pathway toward the desired results. These changes can be automatically made or can be presented to the physician, system administrator, or other user for approval. Benigno at 21:22-29 In addition, the present invention provides a client / server system for manipulation and analysis of data related to clinical pathways, comprising a communication network, a client workstation in communication with the communication network, wherein the client workstation comprises means for generating at least one signal

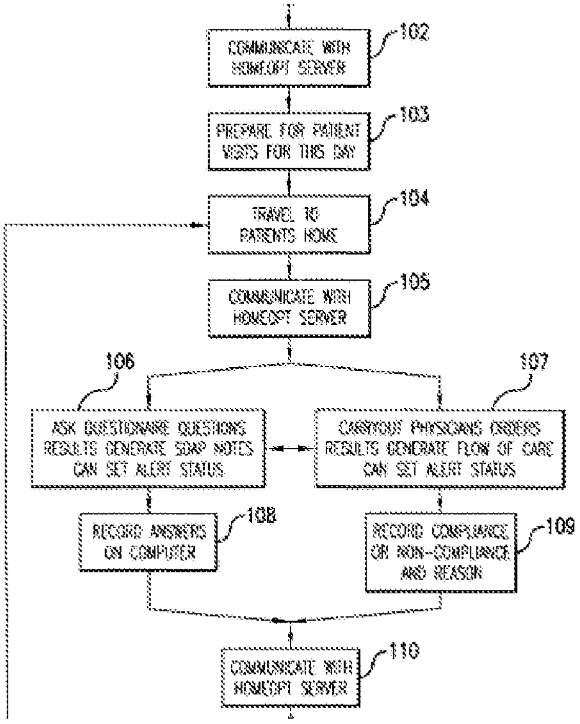
	corresponding to a clinical pathway decision and transmitting the at least one decision signal over the communication network, and means for receiving at least one signal corresponding to a clinical pathway modification from the communication network, and means for outputting the at least one modification signal to a signal processing means, a server on the communication network, wherein the server comprises a clinical pathway database for storing an initial procedure decision data element, corresponding to a decision point within the clinical pathway, and at least one subsequent decision data element corresponding to at least one available subsequent decision point within the clinical pathway, and a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element, and processing means, in communication with the communication network, the client workstation, and the server, for performing the steps of receiving the at least one decision signal from the communication network, based on the received decision signal, selecting one of the at least one subsequent decision data elements, comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database, and based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database, then generating at least one signal corresponding to a clinical pathway modification of the subsequent decision data elements in the clinical pathway database, and transmitting the at least one clinical pathway modification signal over the communication network to the receiving means of the client workstation. Benigno at 14:7-15:2 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401

	by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
(l) removing said remote computer from electronic communication with said second computer;	In step 102, the nurse, 5 using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may

	be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
(m) within said remote computer, using said modified tokenized questionnaire to obtain at least one additional user response;	In step 106 and 108, the client computer 401, via the questionnaire language previously described, or through any other data collection mechanism, may obtain data from the nurse or other source corresponding to the clinical pathway to be followed, as dictated by the physician. As a result, SOAP notes may be generated, alerts can be generated, etc., for ultimate retransmission to the server 402. Benigno at 46:16-20
(n) placing said remote computer into electronic communication with a server;	In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. Benigno at 46:22-24

In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached.
Benigno at 46:30-47:4

Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc.
Benigno at 47:6-13



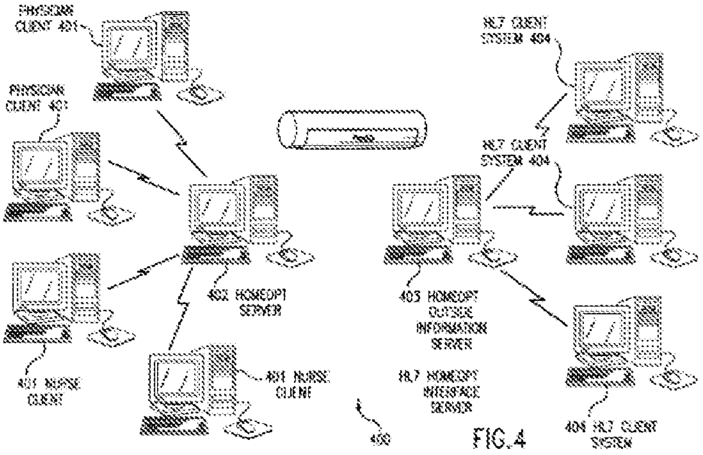
COMMUNICATE WITH HOMEOPT SERVER
102
PREPARE FOR PATIENT VISITS FOR THIS DAY
103
TRAVEL TO PATIENTS HOME
104
COMMUNICATE WITH HOMEOPT SERVER
105
106
107
ASK QUESTIONAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS
CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS
RECORD ANSWERS ON COMPUTER
108
RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON
109
COMMUNICATE WITH HOMEOPT SERVER
110

Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the

	actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
(o) transmitting said at least one user response to said server;	In step 106 and 108, the client computer 401, via the questionnaire language previously described, or through any other data collection mechanism, may obtain data from the nurse or other source corresponding to the clinical pathway to be followed, as dictated by the physician. As a result, SOAP notes may be generated, alerts can be generated, etc., for ultimate retransmission to the server 402. Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. The results of such orders may generate a flow of care to be followed by the nurse, and/or may generate alerts, etc. In step 109, the nurse records in the client computer 401 compliance or non-compliance with the orders. If noncompliance, the reasons are also stored. gain, all such stored data may later be transmitted back to the server 402. Benigno at 46:16-28
(p) transmitting said at least one additional user response to said server;	In step 106 and 108, the client computer 401, via the questionnaire language previously described, or through any other data collection mechanism, may obtain data from the nurse or other source corresponding to the clinical pathway to be followed, as dictated by the physician. As a result, SOAP notes may be generated, alerts can be generated, etc., for ultimate retransmission to the server 402. Benigno at 46:16-20
(q) storing said transmitted at least one user response and said at least one additional user response at said server;	The questionnaire and its answers are all stored. Benigno at 23:10 In step 106 and 108, the client computer 401, via the questionnaire language previously described, or through any other data collection mechanism, may obtain data from the nurse or other source corresponding to the clinical pathway to be followed, as dictated by the physician. As a result, SOAP notes may be generated, alerts can be generated, etc., for ultimate retransmission to the server 402.

	Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. The results of such orders may generate a flow of care to be followed by the nurse, and/or may generate alerts, etc. In step 109, the nurse records in the client computer 401 compliance or non-compliance with the orders. If noncompliance, the reasons are also stored. gain, all such stored data may later be transmitted back to the server 402. Benigno at 46:16-28
(r) preparing a report using any of said at least one user response and said at least one additional user response; and,	The system includes means for … outputting the signal to a signal processing means. Suitable signal processing means include, a communication network, a computer, a storage medium, a display, a printer, or the like. Benigno at 28:26-29:1
(s) displaying at least a portion of said report on a visually perceptible medium;	The system includes means for … outputting the signal to a signal processing means. Suitable signal processing means include, a communication network, a computer, a storage medium, a display, a printer, or the like. Benigno at 28:26-29:1
(t) performing at least steps (d)-(p) using at least two different remote computing device types using the same tokens.	One significant benefit of the invention is that the data gathered about various clinical pathways and their successfulness can be catalogued. The data can be repackaged and manipulated as needed and is believed to be of significant value in and of itself. The gathering of this data as it pertains to the heretofore nonexistent stable acute care patient class is an important advantage of the invention. Benigno at 10:14-18. Statements of the language used to create each questionnaire are saved in the clinical pathway database as opposed to a simple flat file. Entire questionnaires are versioned, and may be easily modified, or recalled from earlier versions. Questions once entered may be reused in many questionnaires. Benigno at 12:27 -31 Each individual question within the questionnaire may be represented by statements in a "questionnaire language". This language is "turing complete" meaning that anything that can be accomplished by any general purpose programming language may be accomplished by the language that represents the questions. This allows the questions to contain data, storage, and logical information about the data within each question, and allows the attachment of significant information to each question within the questionnaire. An example would be information associated with a particular drug. Dosages, appropriate

	application times and such can be encoded within the question that asks if the drug is to be administered. Benigno at 13:1-10 In a further embodiment, the present invention provides a system for assessing utilization of medical resources based upon manipulation and analysis of statistical data related to clinical pathways, comprising a clinical pathway database for storing an initial procedure decision data element, corresponding to a decision point within the clinical pathway, and at least one subsequent decision data element corresponding to available subsequent decision points within the clinical pathway, a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element, and, for each of the previously selected subsequent decision data elements, a utilization value corresponding to the decision data element processing means, including a storage device, for performing the steps of selecting one of the at least one subsequent decision data elements, comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database, and based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database, and statistical processing means, in communication with the clinical pathway database and the historical clinical pathway database, for performing the steps of accessing the historical clinical pathway database, computing pathway utilization value based on the accessed utilization values in the database, generating at least one signal corresponding to the pathway utilization value, and outputting the at least one utilization value signal to a signal processing means. Benigno at 15:18-16:7 In one embodiment of the present invention, clinical pathway database models a decision tree comprising various decision nodes. These nodes are stored as either text or tokenized representations of the Questionnaire Language ("QL") statements *(see infra).* … Question ID is the identifier of the question itself. Benigno at 19:10-24 In step 304, a new patient record is created, using for example the client computer 401 and the server 402. This record may be stored on the server 402, or at any other external location. In step 305, a criteria questionnaire is administered on the client computer 401, in order to determine whether the patient satisfies the criteria to be eligible for, for example, home health care. Examples of such criteria and

	conditions have been previously described elsewhere. Benigno at 49:10-15  PHYSICIAN CLIENT 401 PHYSICIAN CLIENT 401 401 NURSE CLIENT 401 NURSE CLIENT 402 HOMEOPT SERVER 403 HOMEOPT OUTSIDE INFORMATION SERVER HL7 HOMEOPT INTERFACE SERVER HL7 CLIENT SYSTEM 404 HL7 CLIENT SYSTEM 404 404 HL7 CLIENT SYSTEM 400 FIG.4 Benigno at FIG. 4
CLAIM 9	
9. The method for managing data transfers between computers according to claim 8 wherein said first computer and said second computer are a same computer.	In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is

	reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
CLAIM 10	
10. The method for managing data transfers between computers according to claim 9 wherein said server and said first computer are said same computer.	In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out

	orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
CLAIM 11	
11. A method for collecting survey data from a user comprising	A self-analyzing system for suggesting deviation from a current clinical pathway and entry into an alternative clinical pathway based upon historical information about the results of actions. Systems for

the steps of:	tracking clinical pathway outcomes based on data collected post-treatment. …. A questionnaire computer language and subsystem are used in various stages of the systems of the invention. Corresponding methods are also disclosed. Benigno at Abstract
(a) creating a questionnaire comprising a series of questions;	Assessment of the patient's condition is performed using an questionnaire or form generated based upon the current patient's customized and changeable clinical pathway. Benigno at 9:31-10:2 Another aspect of the invention involves a new questionnaire format, which may be used as one way of collecting the data to be analyzed according to the present invention. This questionnaire format allows stable acute care caregivers the ability to closely track and instantly inform a patient's physician of that patient's condition. The format, as it applies to a particular patient, also provides the clinical pathway for the patient, as described *infra.* With the present invention, stable acute care providers receive updated orders about the patient on a visit by visit basis and physicians are able to track the progress of their patients instantly. The questionnaire system in conjunction with the other components of the systems of the invention allows the close communication required between home care givers and physicians in this kind of situation and solves various problems of the prior art. Statements of the language used to create each questionnaire are saved in the clinical pathway database as opposed to a simple flat file. Entire questionnaires are versioned, and may be easily modified, or recalled from earlier versions. Questions once entered may be reused in many questionnaires. Benigno at 12:17-31
(b) tokenizing said questionnaire; thereby producing a plurality of tokens representing said questionnaire;	One significant benefit of the invention is that the data gathered about various clinical pathways and their successfulness can be catalogued. The data can be repackaged and manipulated as needed and is believed to be of significant value in and of itself. The gathering of this data as it pertains to the heretofore nonexistent stable acute care patient class is an important advantage of the invention. Benigno at 10:14-18. Statements of the language used to create each questionnaire are saved in the clinical pathway database as opposed to a simple flat file. Entire questionnaires are versioned, and may be easily modified, or recalled from earlier versions. Questions once entered may be reused in many questionnaires. Benigno at 12:27 -31

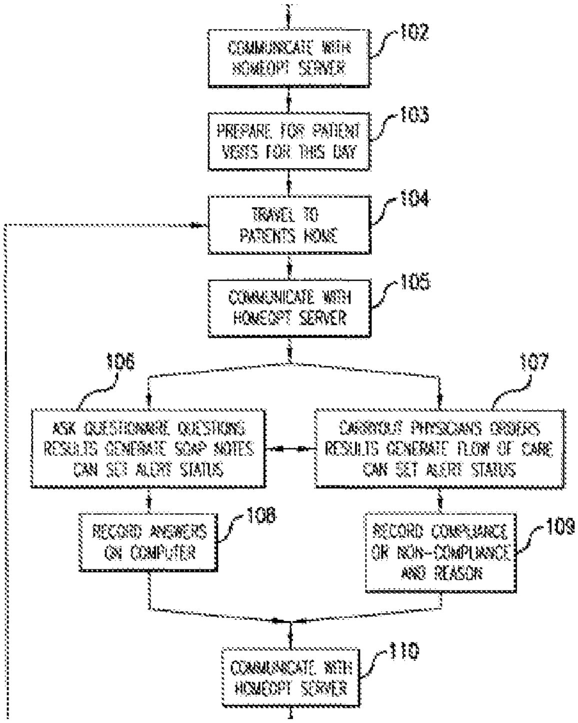
	Each individual question within the questionnaire may be represented by statements in a "questionnaire language". This language is "turing complete" meaning that anything that can be accomplished by any general purpose programming language may be accomplished by the language that represents the questions. This allows the questions to contain data, storage, and logical information about the data within each question, and allows the attachment of significant information to each question within the questionnaire. An example would be information associated with a particular drug. Dosages, appropriate application times and such can be encoded within the question that asks if the drug is to be administered. Benigno at 13:1-10 In one embodiment of the present invention, clinical pathway database models a decision tree comprising various decision nodes. These nodes are stored as either text or tokenized representations of the Questionnaire Language ("QL") statements *(see infra)*. … Question ID is the identifier of the question itself. Benigno at 19:10-24
(c) storing said plurality of tokens on a computer readable medium on a first computer;	Statements of the language used to create each questionnaire are saved in the clinical pathway database as opposed to a simple flat file. Entire questionnaires are versioned, and may be easily modified, or recalled from earlier versions. Questions once entered may be reused in many questionnaires. Benigno at 12:27 -31 Each individual question within the questionnaire may be represented by statements in a "questionnaire language". This language is "turing complete" meaning that anything that can be accomplished by any general purpose programming language may be accomplished by the language that represents the questions. This allows the questions to contain data, storage, and logical information about the data within each question, and allows the attachment of significant information to each question within the questionnaire. An example would be information associated with a particular drug. Dosages, appropriate application times and such can be encoded within the question that asks if the drug is to be administered. Benigno at 13:1-10
(d) placing a handheld remote computing device into electronic communication with said first computer;	The communications subsystems of the invention are important to its capability of providing stable acute care and tracking clinical pathways. Point of service communication at home using either a suitable electronic or computerized device is provided by the invention. The computer can be put into communication with a data

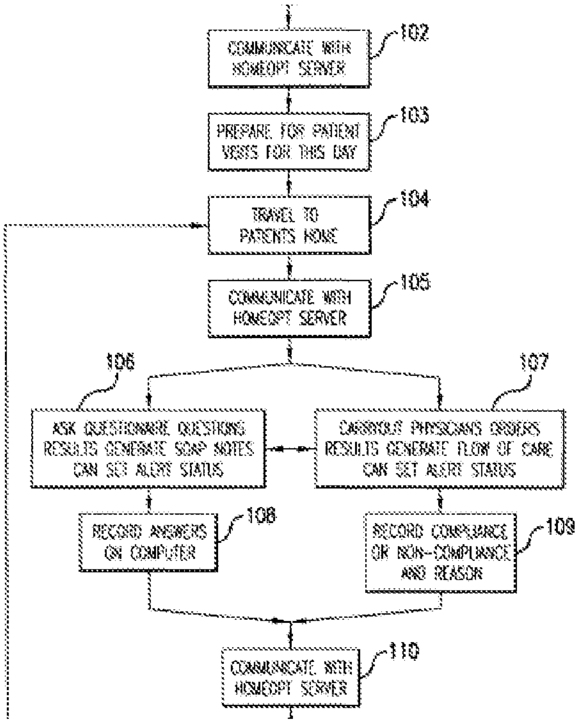
	storage / server computer via any suitable means, including a modem or network adapter. Benigno at 11:14-19 In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13
(e) transmitting said plurality of tokens to said handheld remote computing device;	Therefore, it would be highly desirable to have a system permitting the capability to provide home care and direct information communication to the physician and his or her staff in real time, so as to reduce the recovery period and the risk of complications. Benigno at 6:26-29 The nurse or caregiver then sees the patient almost immediately at home and tracks the patient at home one or more times per day using the system and the information is used to create and update the clinical pathway database records for the patient. Real-time communication systems of the invention allow supervision by the physician, while not requiring the supervision to occur in a hospital setting. Benigno at 9:26-31 The communications subsystems of the invention are important to its capability of providing stable acute care and tracking clinical pathways. Point of service communication at home using either a suitable electronic or computerized device is provided by the invention. The computer can be put into communication with a data storage / server computer via any suitable means, including a modem or network adapter. Benigno at 11:14-19

	In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the

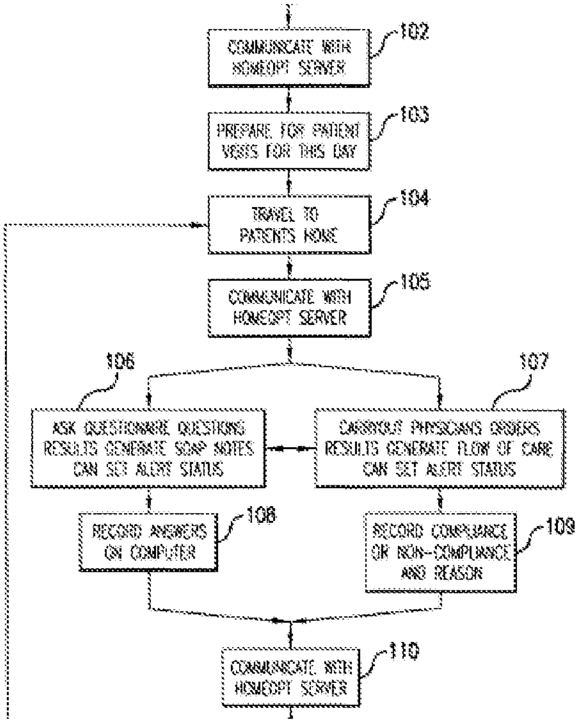
	updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
(f) taking said handheld remote computing device out of electronic communication with said first computer;	In step 102, the nurse, 5 using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines,

	via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30 Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than C-Ch active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random. Rappaport at 7:44-63 In a network that employs an admission control protocol according to the present invention, voice calls, for example (or other time-sensitive stream traffic) may preempt resources of time-insensitive data calls, which result in suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can also be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially

	undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at 2:44-58
(g) after said handheld remote computing device has been taken out of electronic communication with said first computer,	In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14  COMMUNICATE WITH HOMEOPT SERVER 102 PREPARE FOR PATIENT VISITS FOR THIS DAY 103 TRAVEL TO PATIENTS HOME 104 COMMUNICATE WITH HOMEOPT SERVER 105 106 107 ASK QUESTIONNAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS RECORD ANSWERS ON COMPUTER 108 RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON 109 COMMUNICATE WITH HOMEOPT SERVER 110 Benigno at FIG 1A
(g1) executing at least a portion of said plurality of tokens representing said questionnaire on said handheld remote computing device to collect a response from a user, and,	In step 106 and 108, the client computer 401, via the questionnaire language previously described, or through any other data collection mechanism, may obtain data from the nurse or other source corresponding to the clinical pathway to be followed, as dictated by the physician. As a result, SOAP notes may be generated, alerts can be generated, etc., for ultimate retransmission to the server 402. Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. The results of such orders may generate a flow of care to be followed by the nurse, and/or may generate alerts, etc. In step 109, the nurse records in the client computer 401 compliance or non-compliance with the orders.

	If noncompliance, the reasons are also stored. gain, all such stored data may later be transmitted back to the server 402. Benigno at 46:16-28  COMMUNICATE WITH HOMEOPT SERVER 102 PREPARE FOR PATIENT VISITS FOR THIS DAY 103 TRAVEL TO PATIENTS HOME 104 COMMUNICATE WITH HOMEOPT SERVER 105 106 107 ASK QUESTIONNAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS RECORD ANSWERS ON COMPUTER 108 RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON 109 COMMUNICATE WITH HOMEOPT SERVER 110 Benigno at FIG 1A
(g2) storing within said remote computing device said response from the user;	In step 109, the nurse records in the client computer 401 compliance or non-compliance with the orders. If noncompliance, the reasons are also stored. gain, all such stored data may later be transmitted back to the server 402. Benigno at 46:25-28
(h) placing said handheld remote computing device into electronic communication with a second computer;	In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or

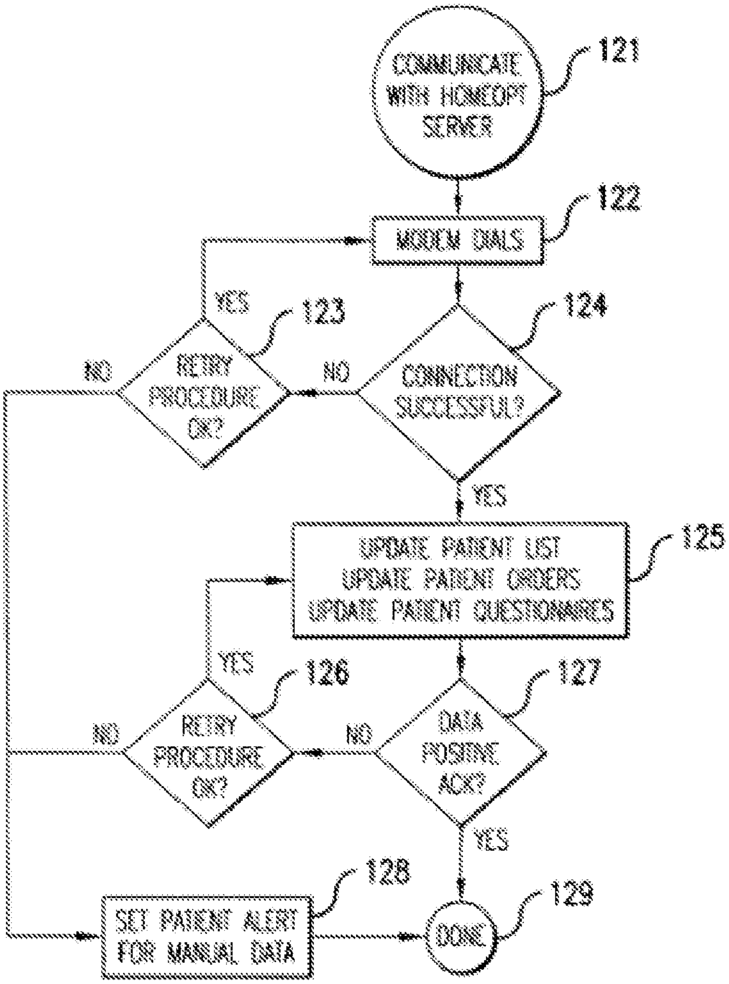
	105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the

	updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
(i) transmitting at least a portion of said response stored within said handheld remote computing device to said second computer; and,	In step 106 and 108, the client computer 401, via the questionnaire language previously described, or through any other data collection mechanism, may obtain data from the nurse or other source corresponding to the clinical pathway to be followed, as dictated by the physician. As a result, SOAP notes may be generated, alerts can be generated, etc., for ultimate retransmission to the server 402. Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401. The results of such orders may generate a flow of care to be followed by the nurse, and/or may generate alerts, etc. In step 109, the nurse records in the client computer 401 compliance or non-compliance with the orders. If noncompliance, the reasons are also stored. gain, all such stored data may later be transmitted back to the server 402. Benigno at 46:16-28  COMMUNICATE WITH HOMEOPT SERVER 102 PREPARE FOR PATIENT VISITS FOR THIS DAY 103 TRAVEL TO PATIENTS HOME 104 COMMUNICATE WITH HOMEOPT SERVER 105 106 107 ASK QUESTIONNAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS RECORD ANSWERS ON COMPUTER 108 RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON 109 COMMUNICATE WITH HOMEOPT SERVER 110

	Benigno at FIG 1A In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4
(j) forming a visually perceptible report from any of said at least a portion of said response so transmitted.	The system includes means for … outputting the signal to a signal processing means. Suitable signal processing means include, a communication network, a computer, a storage medium, a display, a printer, or the like. Benigno at 28:26-29:1
CLAIM 12	
12. A method for collecting survey data from a user according to claim 11, wherein step (j) comprises the step of printing a report from any of said response to transmitted.	The system includes means for … outputting the signal to a signal processing means. Suitable signal processing means include, a communication network, a computer, a storage medium, a display, a printer, or the like. Benigno at 28:26-29:1
CLAIM 13	
13. A method for collecting survey data from a user according to claim 11, wherein said first computer and said second computer are a same computer.	In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link. Benigno at 46:4-9 In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data. Benigno at 46:13-14 Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or

	105 from the server 402 to the client computer 401. Benigno at 46:22-24 In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached. Benigno at 46:30-47:4 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc. Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30
CLAIM 14	
14. A method for modifying a questionnaire used in data management	Daily communication includes SOAP notes, notification of whether the patient received appropriate IV medications and intravenous fluids, as well as the ability to communicate with nurses, and nurse communications with physicians for order changes.

according to the method of claim 11, further comprising the steps of: (k) making at least one incremental change to a portion of said questionnaire;	Benigno at 11:22-26 The system is constantly evaluating itself. As the system finds new correlating factors, they are put in place to aid in determining changes to be made to the current or default clinical pathway. In addition, as correlations are determined between clinical pathway decisions and significant outcomes *(i.e.,* outcomes of interest), changes can be made to the default pathway to optimize systematically the clinical pathway toward the desired results. These changes can be automatically made or can be presented to the physician, system administrator, or other user for approval. Benigno at 21:22-29 In addition, the present invention provides a client / server system for manipulation and analysis of data related to clinical pathways, comprising a communication network, a client workstation in communication with the communication network, wherein the client workstation comprises means for generating at least one signal corresponding to a clinical pathway decision and transmitting the at least one decision signal over the communication network, and means for receiving at least one signal corresponding to a clinical pathway modification from the communication network, and means for outputting the at least one modification signal to a signal processing means, a server on the communication network, wherein the server comprises a clinical pathway database for storing an initial procedure decision data element, corresponding to a decision point within the clinical pathway, and at least one subsequent decision data element corresponding to at least one available subsequent decision point within the clinical pathway, and a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element, and processing means, in communication with the communication network, the client workstation, and the server, for performing the steps of receiving the at least one decision signal from the communication network, based on the received decision signal, selecting one of the at least one subsequent decision data elements, comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database, and based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database, then generating at least one signal corresponding to a clinical pathway modification of the subsequent decision data elements in the clinical pathway database, and transmitting the at least one clinical pathway modification signal over the communication network to the receiving means of the client workstation.

	Benigno at 14:7-15:2  COMMUNICATE WITH HOMEOPT SERVER 121 MODEM DIALS 122 YES 123 124 NO RETRY PROCEDURE OK? NO CONNECTION SUCCESSFUL? YES UPDATE PATIENT LIST UPDATE PATIENT ORDERS UPDATE PATIENT QUESTIONAIRES 125 YES 126 127 NO RETRY PROCEDURE OK? NO DATA POSITIVE ACK? YES 128 129 SET PATIENT ALERT FOR MANUAL DATA DONE Benigno at FIG. 1B
(l) tokenizing said at least one incremental change to said questionnaire;	One significant benefit of the invention is that the data gathered about various clinical pathways and their successfulness can be catalogued. The data can be repackaged and manipulated as needed and is believed to be of significant value in and of itself. The gathering of this data as it pertains to the heretofore nonexistent stable acute care patient class is an important advantage of the invention. Benigno at 10:14-18. Statements of the language used to create each questionnaire are saved in the clinical pathway database as opposed to a simple flat file. Entire questionnaires are versioned, and may be easily modified, or recalled from earlier versions. Questions once entered may be reused in many questionnaires.

	Benigno at 12:27 -31 Each individual question within the questionnaire may be represented by statements in a "questionnaire language". This language is "turing complete" meaning that anything that can be accomplished by any general purpose programming language may be accomplished by the language that represents the questions. This allows the questions to contain data, storage, and logical information about the data within each question, and allows the attachment of significant information to each question within the questionnaire. An example would be information associated with a particular drug. Dosages, appropriate application times and such can be encoded within the question that asks if the drug is to be administered. Benigno at 13:1-10 In one embodiment of the present invention, clinical pathway database models a decision tree comprising various decision nodes. These nodes are stored as either text or tokenized representations of the Questionnaire Language ("QL") statements *(see infra).* … Question ID is the identifier of the question itself. Benigno at 19:10-24
(m) transmitting at least a portion of said tokens resulting from step (k) to said remote handheld computing device, said transmitted tokens comprising less than the entire tokenized questionnaire; and,	Therefore, it would be highly desirable to have a system permitting the capability to provide home care and direct information communication to the physician and his or her staff in real time, so as to reduce the recovery period and the risk of complications. Benigno at 6:26-29 The nurse or caregiver then sees the patient almost immediately at home and tracks the patient at home one or more times per day using the system and the information is used to create and update the clinical pathway database records for the patient. Real-time communication systems of the invention allow supervision by the physician, while not requiring the supervision to occur in a hospital setting. Benigno at 9:26-31 The communications subsystems of the invention are important to its capability of providing stable acute care and tracking clinical pathways. Point of service communication at home using either a suitable electronic or computerized device is provided by the invention. The computer can be put into communication with a data storage / server computer via any suitable means, including a modem or network adapter. Benigno at 11:14-19

In step 102, the nurse, using the client computer 401 (Figure 4) communicates with the server 402, in order to obtain updated pathway instructions, etc., regarding what steps to perform during visit(s) for one or more patient(s). The communication can take place via modern and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link.
Benigno at 46:4-9

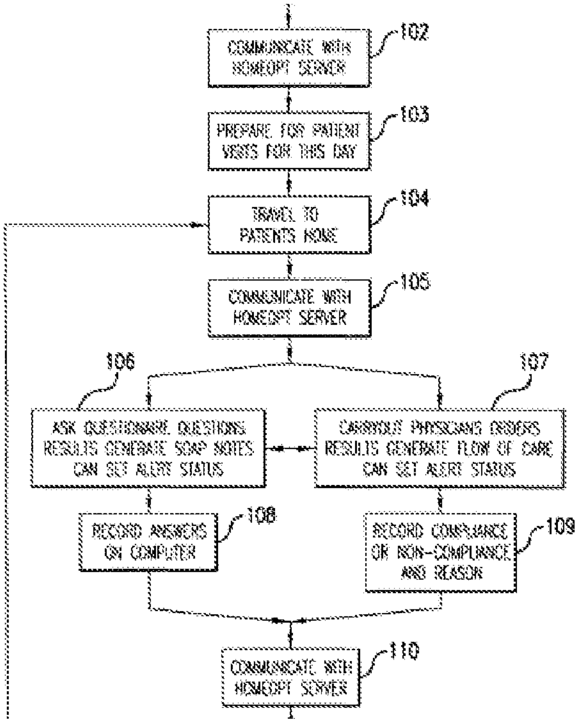
In step 105, the nurse, through client computer 401, may again communicate with server 402, in order to obtain the most current instructions and data.
Benigno at 46:13-14

Alternatively, or in addition, in step 107 the nurse may carry out orders created by the physician and transmitted in steps 102 and/or 105 from the server 402 to the client computer 401.
Benigno at 46:22-24

In step 110, the client computer 401 communicates with the server 402, in order to update both the computer 401 and server 402 as in steps 102 and 105. In step 111, if there are additional patients assigned to the nurse, as would be indicated on a list maintained on the computer 401 (as communicated from the server 402), then steps 104-110 may be repeated for each of the remaining patients. After all patients have been processed by the nurse, the final step 112 is reached.
Benigno at 46:30-47:4

Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc.
Benigno at 47:6-13

Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the

<table>
<tr>
<td></td>
<td>
<p>updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402.
Benigno at 47:20-27</p>
<p>In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered.
Benigno at 47:29-30</p>

COMMUNICATE WITH HOMEOPT SERVER
102
PREPARE FOR PATIENT VISITS FOR THIS DAY
103
TRAVEL TO PATIENTS HOME
104
COMMUNICATE WITH HOMEOPT SERVER
105
106
107
ASK QUESTIONNAIRE QUESTIONS RESULTS GENERATE SOAP NOTES CAN SET ALERT STATUS
CARRYOUT PHYSICIANS ORDERS RESULTS GENERATE FLOW OF CARE CAN SET ALERT STATUS
RECORD ANSWERS ON COMPUTER
108
RECORD COMPLIANCE OR NON-COMPLIANCE AND REASON
109
COMMUNICATE WITH HOMEOPT SERVER
110
<p>Benigno at FIG 1A</p>
</td>
</tr>
<tr>
<td>(n) incorporating said transmitted tokens into said questionnaire at said remote computing device, thereby incrementally changing said questionnaire.</td>
<td>
<p>Daily communication includes SOAP notes, notification of whether the patient received appropriate IV medications and intravenous fluids, as well as the ability to communicate with nurses, and nurse communications with physicians for order changes.
Benigno at 11:22-26</p>
<p>The system is constantly evaluating itself. As the system finds new correlating factors, they are put in place to aid in determining changes to be made to the current or default clinical pathway. In addition, as correlations are determined between clinical pathway decisions and significant outcomes <i>(i.e.,</i> outcomes of interest), changes can be made to the default pathway to optimize systematically the clinical pathway toward the desired results. These changes can be automatically made or can be presented to the physician, system administrator, or other user for approval.</p>
</td>
</tr>
</table>

	Benigno at 21:22-29 In addition, the present invention provides a client / server system for manipulation and analysis of data related to clinical pathways, comprising a communication network, a client workstation in communication with the communication network, wherein the client workstation comprises means for generating at least one signal corresponding to a clinical pathway decision and transmitting the at least one decision signal over the communication network, and means for receiving at least one signal corresponding to a clinical pathway modification from the communication network, and means for outputting the at least one modification signal to a signal processing means, a server on the communication network, wherein the server comprises a clinical pathway database for storing an initial procedure decision data element, corresponding to a decision point within the clinical pathway, and at least one subsequent decision data element corresponding to at least one available subsequent decision point within the clinical pathway, and a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element, and processing means, in communication with the communication network, the client workstation, and the server, for performing the steps of receiving the at least one decision signal from the communication network, based on the received decision signal, selecting one of the at least one subsequent decision data elements, comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database, and based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database, then generating at least one signal corresponding to a clinical pathway modification of the subsequent decision data elements in the clinical pathway database, and transmitting the at least one clinical pathway modification signal over the communication network to the receiving means of the client workstation. Benigno at 14:7-15:2 Steps 102, 105 and 110, wherein the client computer 401 communicates with the server 402, are each described in further detail in steps 121-129, depicted in Figure lB. In step 121, the processes commences. In step 122, the modem on the client computer 401 dials into the server 402. Again, this assumes that the computer 401 and server 402 are to be connected via modem and standard telephone lines. Again, it will be understood that this connection may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc.

	Benigno at 47:6-13 Assuming the connection between the client computer 401 and server 402 is successful in step 124, then in step 125 the patient list, patient orders and patient questionnaire is updated. Specifically, the client computer 401 sends information to the server 402 regarding the actions that the nurse has taken (as input into the client computer 401 by the nurse), and the server 402 sends to the client computer 401 the updated patient list, patient orders, patient questionnaires, flow of care, etc. Other data as appropriate may also be transmitted back and forth between the client computer 401 and the server 402. Benigno at 47:20-27 In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Benigno at 47:29-30

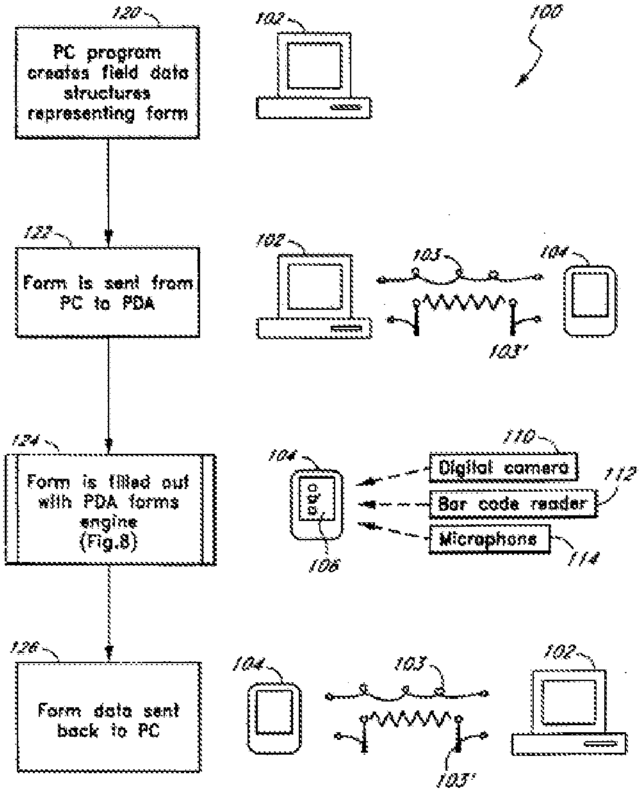
F. CLAIMS 1, 2, 5-7, AND 11-14 OF THE '816 PATENT ARE RENDERED OBVIOUS BY WRIGHT, RAPPAPORT, WARTHEN, BROOKLER, AND ROSSMANN AND THE KNOWLEDGE OF A PERSON OF ORDINARY SKILL IN THE ART

Please see the below claim chart that applies the teachings of Wright in view of Rappaport, Warthen, Brookler, and Rossmann to claims 1, 2, 5-7, and 11-14 of the '816 patent. Requester also would like to inform the Office that a similar rejection to the one provided below is proposed by an Examiner in a non-final Office Action in the '816's child application currently pending as U.S. Application No. 12/910,706.

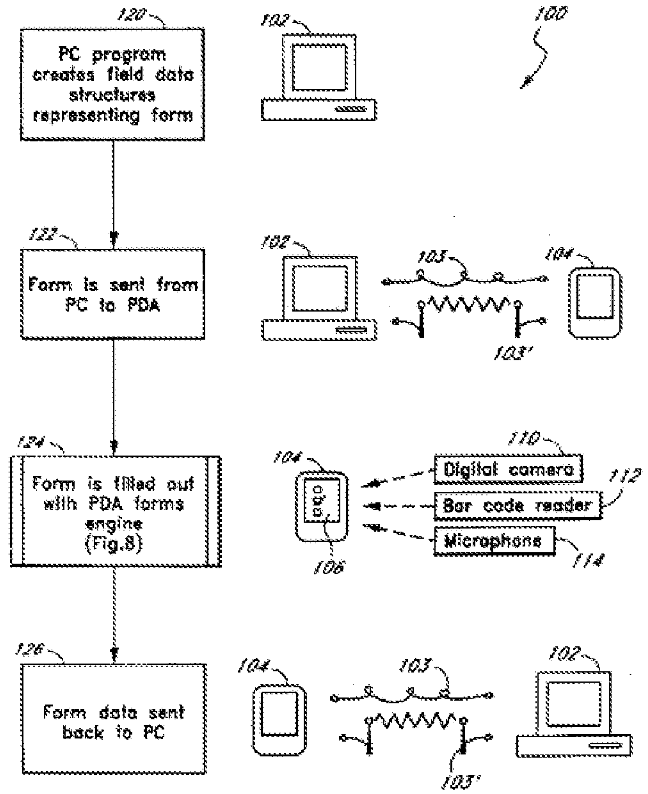
Reasons to Combine:

A person of skill in the art would be motivated to combine the teachings as provided below for the following reasons. Wright teaches the use of surveys and Warthen teaches it was known in the art to tokenize a survey and it would lead to a predictable result providing electronic surveys with feedback. *See* Wright at Abstract; Warthen at Abstract. It would be further obvious to add the teachings of Brookler since Brookler teaches transmitting and storing responses, so as to provide the predictable result of being able to store all answers to survey questions on the server. This would provide a single central location for all results allowing for ease of access by users conducting a survey. *See* Brookler at para. 0002. It also would have been obvious to combine Wright, Warthen and Brookler with Rappaport so that when a connection fails, as will predictably happen, the device can reconnect and send the information upon reconnection. This would motivate a person of skill in the art to make the combination since disconnections are a common occurrence and Rappaport teaches a method of reconnection. *See* Rappaport at Abstract. Finally, it would have been obvious to a person of ordinary skill in the art that users of a system that collects questionnaires would need the ability to print reports based on the completed questionnaires. This would motivate the further combination with Rossmann since it provides a predictable method for printing accumulated results.

CLAIM 1	**Teachings from Wright in view of Warthen, Rappaport, Brookler, and Rossmann**
1. A method for managing data including the steps of:	Once an electronic form is loaded in the PDA 104 (FIG. 1), a data collector or interviewer in a survey or questionnaire type of application, can choose to fill out any of a number of forms which have been previously loaded. The number of forms that can be loaded depends on the amount of random-access memory (RAM) (not shown) in the PDA 104. Current PDA devices can generally store large numbers of reasonably small forms in the standard RAM. Users complete or fill out a form during execution of the forms engine function 124. Once form execution begins, the interviewer has very little to do when compared to the paper based process. Because the scripts, running under the forms engine, control form execution, the user doesn't need to follow any complex skip patterns or even worry about answering a question incorrectly. Aside from answering the current question, he/she can select one of four functions by tapping on buttons (FIG. 3): go to the NEXT field 220, go to PREVIOUS field 222, get HELP 224, or QUIT 226. When the NEXT button 220 is pressed, the field script is executed. At this point, the PDA 104 running the script may check to see that the field was answered correctly and if so, go to the next appropriate field. If not, the script may cause the PDA 104 to beep and display a help message, or perform some other action. The forms engine 124 tracks the progress as the user progresses through the form. At any point while completing the form, the user may tap the PREVIOUS button 222 to go back to change an answer to a previous question. Also, the user may tap the HELP button 224 at any time for instructions for the current field. Wright at 13:38-67

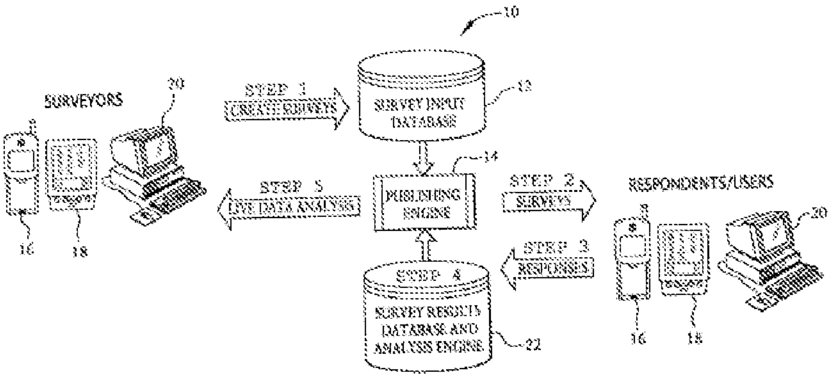
	 120 PC program creates field data structures representing form 102 100 122 Form is sent from PC to PDA 102 103 104 103' 124 Form is filled out with PDA forms engine (Fig.8) 104 110 Digital camera 112 Bar code reader Microphone 114 106 126 Form data sent back to PC 104 103 102 103' *Fig. 1*
(a) creating a questionnaire comprising a series of questions;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting.

	Wright at ABSTRACT Once an electronic form is loaded in the PDA 104 (FIG. 1), a data collector or interviewer in a survey or questionnaire type of application, can choose to fill out any of a number of forms which have been previously loaded. The number of forms that can be loaded depends on the amount of random-access memory (RAM) (not shown) in the PDA 104. Current PDA devices can generally store large numbers of reasonably small forms in the standard RAM. Users complete or fill out a form during execution of the forms engine function 124. Once form execution begins, the interviewer has very little to do when compared to the paper based process. Because the scripts, running under the forms engine, control form execution, the user doesn't need to follow any complex skip patterns or even worry about answering a question incorrectly. Aside from answering the current question, he/she can select one of four functions by tapping on buttons (FIG. 3): go to the NEXT field 220, go to PREVIOUS field 222, get HELP 224, or QUIT 226. When the NEXT button 220 is pressed, the field script is executed. At this point, the PDA 104 running the script may check to see that the field was answered correctly and if so, go to the next appropriate field. If not, the script may cause the PDA 104 to beep and display a help message, or perform some other action. The forms engine 124 tracks the progress as the user progresses through the form. At any point while completing the form, the user may tap the PREVIOUS button 222 to go back to change an answer to a previous question. Also, the user may tap the HELP button 224 at any time for instructions for the current field. Wright at 13:38-67

	 120 PC program creates field data structures representing form 102 100 122 Form is sent from PC to PDA 102 103 104 103' 124 Form is filled out with PDA forms engine (Fig.8) 104 110 Digital camera 112 Bar code reader Microphone 114 106 126 Form data sent back to PC 104 103 102 103' *Fig. 1*
(b) tokenizing said questionnaire; thereby producing a plurality of tokens representing said questionnaire;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting.

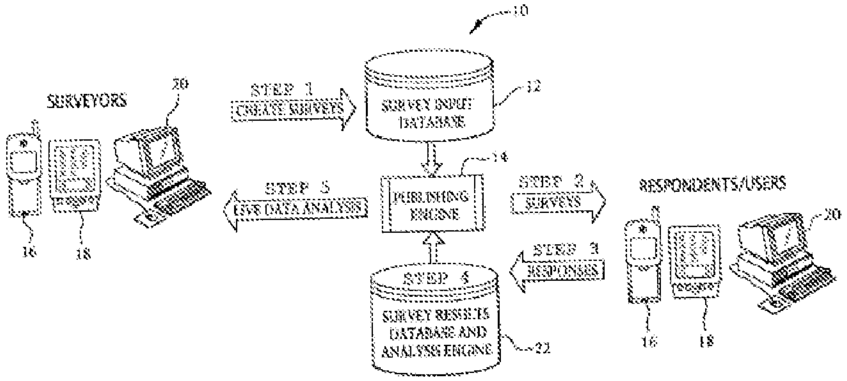
	Wright at ABSTRACT An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at 2:1-11
(c) establishing a first wireless modem or wireless LAN network connection with a remote computing device;	The present invention is directed to a system and method that allows mobile platforms each supporting any of a variety of call types, and each having differing mobility characteristics, to maintain connectivity to a backbone network in spite of unreliable radio links that occasionally fail. It accomplishes this by using automatic and user-transparent reconnection attempts for appropriate call sessions when an interruption of the link occurs. The network may be supporting a variety of different call types simultaneously. Access to network connectivity resources can be provided according to call session priority based on (for example, call session type, platform mobility, hand-off status, and user class (fee-for-service)) criteria. The technology allows support of suspended sessions and uses repeated reconnection attempts with priority access to network resources. It also

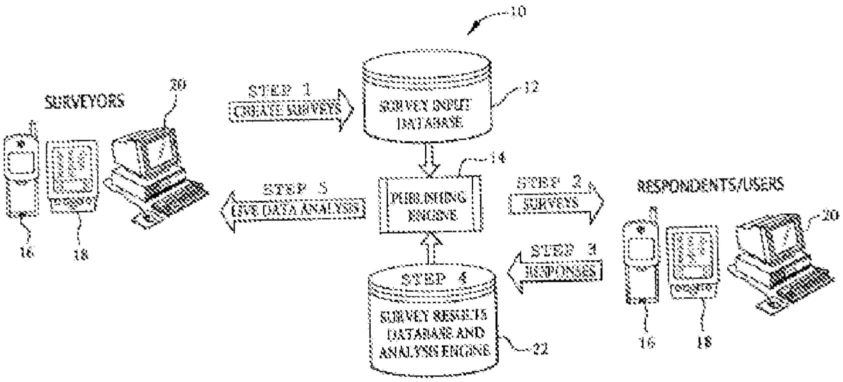
	provides for hand-offs of suspended sessions to neighboring gateways as mobile terminals move throughout the service area. In a network that uses this technology, for example, voice calls (typical of time-sensitive stream traffic) may preempt resources of time-insensitive data calls causing suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at ABSTRACT In a network that employs an admission control protocol according to the present invention, voice calls, for example (or other time-sensitive stream traffic) may preempt resources of time-insensitive data calls, which result in suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can also be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at 2:44-58
(d) transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection;	Step 5: Analyzed Data Publishing and Reporting: Once the data has been analyzed in accordance with parameters prescribed by the surveyor, reports and results are published by publishing engine 14. As specified by the surveyor, publishing engine 14 makes the survey results and reports available for access by the surveyor via one or more interface devices (e.g., cell phone 16, PDA 18 or PC 20). The survey results may be "pushed" by the survey publishing system server to the surveyor in the manner known in the art. Alternatively, and preferably, the survey results and reports may passively reside on the survey publishing system server where they may be "pulled" or retrieved by the surveyor. In the latter case, survey publishing system 10 may be programmed to notify the surveyor via electronic mail message or otherwise that the survey results are available for access. It will be understood that publishing engine 14 will publish different levels and depth of data according to the interface device specified by the surveyor to receive the data, e.g., a cell phone will only get high level reports, whereas a PC will receive detailed analysis. Brookler at [0033]

	 10 SURVEYORS 20 STEP 1 CREATE SURVEYS SURVEY INPUT DATABASE 12 14 STEP 5 LIVE DATA ANALYSIS PUBLISHING ENGINE STEP 2 SURVEYS RESPONDENTS/USERS 20 16 18 STEP 3 RESPONSES STEP 4 SURVEY RESULTS DATABASE AND ANALYSIS ENGINE 22 16 18 FIG. 1 HTTP server 749 uses UDP interface module 748 to send data to and receive data from CDPD network 710. TIL decks 760 are TIL decks that can be accessed by HTTP server 749. Static files containing PIDL decks are converted to TIL decks only once on HTTP server 749. CGI programs 761 are common gateway interface programs that produce PIDL decks that are used by HTTP server 749 to produce TIL decks that in turn are transmitted via UDP interface modules 748 and 714 and cellular telephone network 710 to client module 702. In this embodiment, the services available 35 over airnet network 750 are applications accessible by HTTP server 749 on Internet 140 for which a service developer has written a PIDL deck, or a CGI script that in turn generates a PIDL deck, and is stored on computer 743. Rappaport at 16:30-36.
(e) terminating said first wireless modem or wireless LAN network connection with said remote computing device;	Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than C-C_h active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random.

	Rappaport at 7:44-63 In contrast, mobile users that are engaged in mobile computing (or other forms of data transmission) may have the capability to operate semi-autonomously since data communications with the network are packetized and not necessarily streamed. So with appropriate network design, a temporary disconnection from the network may be transparent to the user. Thus, by implementing the techniques described herein, short term radio link disconnections, which are frequent in mobile communications, need not result in failed sessions, discarded information and wasted use of resources. Rappaport at 5:2-15 In a network that employs an admission control protocol according to the present invention, voice calls, for example (or other time-sensitive stream traffic) may preempt resources of time-insensitive data calls, which result in suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can also be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at 2:44-58
(f) after said first wireless modem or wireless LAN network connection is terminated, executing at least a portion of said plurality of tokens representing said questionnaire at said remote computing device to collect a response from a user;	Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than $C\text{-}C_h$ active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random. Rappaport at 7:44-63 In contrast, mobile users that are engaged in mobile computing (or other

	forms of data transmission) may have the capability to operate semi-autonomously since data communications with the network are packetized and not necessarily streamed. So with appropriate network design, a temporary disconnection from the network may be transparent to the user. Thus, by implementing the techniques described herein, short term radio link disconnections, which are frequent in mobile communications, need not result in failed sessions, discarded information and wasted use of resources. Rappaport at 5:2-15
(g) establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server;	The present invention is directed to a system and method that allows mobile platforms each supporting any of a variety of call types, and each having differing mobility characteristics, to maintain connectivity to a backbone network in spite of unreliable radio links that occasionally fail. It accomplishes this by using automatic and user-transparent reconnection attempts for appropriate call sessions when an interruption of the link occurs. The network may be supporting a variety of different call types simultaneously. Access to network connectivity resources can be provided according to call session priority based on (for example, call session type, platform mobility, hand-off status, and user class (fee-for-service)) criteria. The technology allows support of suspended sessions and uses repeated reconnection attempts with priority access to network resources. It also provides for hand-offs of suspended sessions to neighboring gateways as mobile terminals move throughout the service area. In a network that uses this technology, for example, voice calls (typical of time-sensitive stream traffic) may preempt resources of time-insensitive data calls causing suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at ABSTRACT Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than C-C_h active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data

	session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random. Rappaport at 7:44-63
(h) after said second wireless modem or wireless LAN network connection is established, transmitting at least a portion of said response from the user to said server via said second wireless modem or wireless LAN network connection; and	Step 5: Analyzed Data Publishing and Reporting: Once the data has been analyzed in accordance with parameters prescribed by the surveyor, reports and results are published by publishing engine 14. As specified by the surveyor, publishing engine 14 makes the survey results and reports available for access by the surveyor via one or more interface devices (e.g., cell phone 16, PDA 18 or PC 20). The survey results may be "pushed" by the survey publishing system server to the surveyor in the manner known in the art. Alternatively, and preferably, the survey results and reports may passively reside on the survey publishing system server where they may be "pulled" or retrieved by the surveyor. In the latter case, survey publishing system 10 may be programmed to notify the surveyor via electronic mail message or otherwise that the survey results are available for access. It will be understood that publishing engine 14 will publish different levels and depth of data according to the interface device specified by the surveyor to receive the data, e.g., a cell phone will only get high level reports, whereas a PC will receive detailed analysis. Brookler at [0033]  10 SURVEYORS 20 STEP 1 CREATE SURVEYS SURVEY INPUT DATABASE 12 14 STEP 5 LIVE DATA ANALYSIS PUBLISHING ENGINE STEP 2 SURVEYS RESPONDENTS/USERS 20 16 18 STEP 3 RESPONSES STEP 4 SURVEY RESULTS DATABASE AND ANALYSIS ENGINE 22 16 18 FIG. 1
(i) storing said transmitted response at said server.	Step 5: Analyzed Data Publishing and Reporting: Once the data has been analyzed in accordance with parameters prescribed by the surveyor, reports and results are published by publishing engine 14. As specified by the surveyor, publishing engine 14 makes the survey results and reports

	available for access by the surveyor via one or more interface devices (e.g., cell phone 16, PDA 18 or PC 20). The survey results may be "pushed" by the survey publishing system server to the surveyor in the manner known in the art. Alternatively, and preferably, the survey results and reports may passively reside on the survey publishing system server where they may be "pulled" or retrieved by the surveyor. In the latter case, survey publishing system 10 may be programmed to notify the surveyor via electronic mail message or otherwise that the survey results are available for access. It will be understood that publishing engine 14 will publish different levels and depth of data according to the interface device specified by the surveyor to receive the data, e.g., a cell phone will only get high level reports, whereas a PC will receive detailed analysis. Brookler at [0033]  10 SURVEYORS 20 STEP 1 CREATE SURVEYS SURVEY INPUT DATABASE 12 14 PUBLISHING ENGINE STEP 5 LIVE DATA ANALYSIS STEP 2 SURVEYS RESPONDENTS/USERS 20 STEP 3 RESPONSES STEP 4 SURVEY RESULTS DATABASE AND ANALYSIS ENGINE 22 16 18 16 18 FIG. 1
CLAIM 2	
2. The method for managing data of claim 1 further comprising the step of: (j) translating said response to a format recognizable by a particular computer program; and	According to a presently preferred embodiment, data collected during the survey building or creation process is stored in the survey input database 12 and translated to XML for optimal portability vis-a-vis presently available interface devices. Using publishing engine 14, the surveyor may opt to publish the survey immediately after creation and simultaneously to all types of devices. In the alternative, the surveyor may choose to delay the launch of the surveyor stagger the times at which the survey information is published to the various interface devices selected for participation in the survey. By way of example, publishing engine 14 may at present be configured to publish to cell phones using the wireless application protocol (WAP) (which incorporates WML), short messaging service (SMS) using the global system for mobile communication (GSM) or VoiceXML. Likewise, publishing engine may also publish to computer browsers via HTML and to Palm® devices or other PDAs using PalmOS or other suitable PDA operating systems (or SMS) depending on where the surveyor wishes the poll to be published.

	Brookler at [0051] A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT
(k) accessing the translated response from a computer executing said particular computer program.	According to a presently preferred embodiment, data collected during the survey building or creation process is stored in the survey input database 12 and translated to XML for optimal portability vis-a-vis presently available interface devices. Using publishing engine 14, the surveyor may opt to publish the survey immediately after creation and simultaneously to all types of devices. In the alternative, the surveyor may choose to delay the launch of the surveyor stagger the times at which the survey information is published to the various interface devices selected for participation in the survey. By way of example, publishing engine 14 may at present be configured to publish to cell phones using the wireless application protocol (WAP) (which incorporates WML), short messaging service (SMS) using the global system for mobile communication (GSM) or VoiceXML. Likewise, publishing engine may also publish to computer browsers via HTML and to Palm® devices or other PDAs using PalmOS or other suitable PDA operating systems (or SMS) depending on where the surveyor wishes the poll to be published. Brookler at [0051] A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting

	feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT
CLAIM 5	
5. A method for modifying a questionnaire used in data management according to the method of claim 1 including the steps of: (a) making at least one incremental change to a portion of the questionnaire;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if

	the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10. Wright at 16:41-54 *See also* Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT
(b) tokenizing said at least one incremental change to said questionnaire;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a

handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting.
Wright at ABSTRACT

If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10.
Wright at 16:41-54

See also Wright at Figures 1-11

An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.
Warthen at ABSTRACT

(c) transmitting at least a portion of said tokens resulting from step (b) to a remote loosely networked computing device, said transmitted tokens comprising less than the entire tokenized questionnaire; and,	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10. Wright at 16:41-54 *See also* Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an

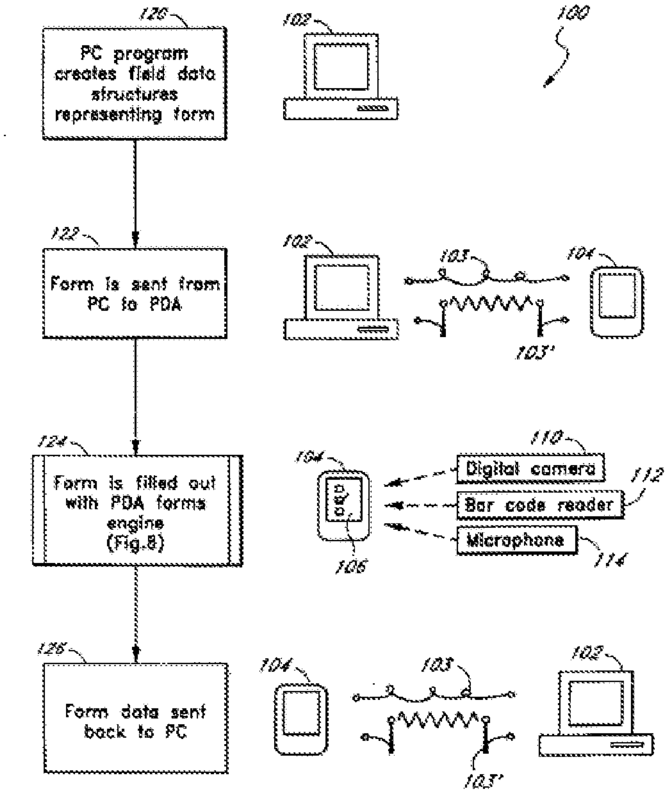
	answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT
(d) incorporating said transmitted tokens into said questionnaire at said loosely networked remote computing device, thereby modifying said questionnaire.	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10.

	Wright at 16:41-54 *See also* Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT
CLAIM 6	
6. A method for managing data according to claim 1, wherein said first wireless modem or wireless LAN network connection and said second wireless modem or wireless LAN network connection are a same wireless modem or wireless LAN network connection.	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is

	completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10. Wright at 16:41-54 *See also* Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT
CLAIM 7	
7. The method of claim 1 further including	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting

performing at least the steps (c)-(k) for at least two different remote computing device types using the same tokens.	feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a

	matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at 2:1-11
CLAIM 11	
11. A method for collecting survey data from a user comprising the steps of:	Once an electronic form is loaded in the PDA 104 (FIG. 1), a data collector or interviewer in a survey or questionnaire type of application, can choose to fill out any of a number of forms which have been previously loaded. The number of forms that can be loaded depends on the amount of random-access memory (RAM) (not shown) in the PDA 104. Current PDA devices can generally store large numbers of reasonably small forms in the standard RAM. Users complete or fill out a form during execution of the forms engine function 124. Once form execution begins, the interviewer has very little to do when compared to the paper based process. Because the scripts, running under the forms engine, control form execution, the user doesn't need to follow any complex skip patterns or even worry about answering a question incorrectly. Aside from answering the current question, he/she can select one of four functions by tapping on buttons (FIG. 3): go to the NEXT field 220, go to PREVIOUS field 222, get HELP 224, or QUIT 226. When the NEXT button 220 is pressed, the field script is executed. At this point, the PDA 104 running the script may check to see that the field was answered correctly and if so, go to the next appropriate field. If not, the script may cause the PDA 104 to beep and display a help message, or perform some other action. The forms engine 124 tracks the progress as the user progresses through the form. At any point while completing the form, the user may tap the PREVIOUS button 222 to go back to change an answer to a previous question. Also, the user may tap the HELP button 224 at any time for instructions for the current field. Wright at 13:38-67

	 120 PC program creates field data structures representing form 102 100 122 Form is sent from PC to PDA 102 103 104 103' 124 Form is filled out with PDA forms engine (Fig.8) 104 110 Digital camera 112 Bar code reader Microphone 114 106 126 Form data sent back to PC 104 103 102 103' *Fig. 1*
(a) creating a questionnaire comprising a series of questions;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer

	for further processing or reporting. Wright at ABSTRACT Once an electronic form is loaded in the PDA 104 (FIG. 1), a data collector or interviewer in a survey or questionnaire type of application, can choose to fill out any of a number of forms which have been previously loaded. The number of forms that can be loaded depends on the amount of random-access memory (RAM) (not shown) in the PDA 104. Current PDA devices can generally store large numbers of reasonably small forms in the standard RAM. Users complete or fill out a form during execution of the forms engine function 124. Once form execution begins, the interviewer has very little to do when compared to the paper based process. Because the scripts, running under the forms engine, control form execution, the user doesn't need to follow any complex skip patterns or even worry about answering a question incorrectly. Aside from answering the current question, he/she can select one of four functions by tapping on buttons (FIG. 3): go to the NEXT field 220, go to PREVIOUS field 222, get HELP 224, or QUIT 226. When the NEXT button 220 is pressed, the field script is executed. At this point, the PDA 104 running the script may check to see that the field was answered correctly and if so, go to the next appropriate field. If not, the script may cause the PDA 104 to beep and display a help message, or perform some other action. The forms engine 124 tracks the progress as the user progresses through the form. At any point while completing the form, the user may tap the PREVIOUS button 222 to go back to change an answer to a previous question. Also, the user may tap the HELP button 224 at any time for instructions for the current field. Wright at 13:38-67
(b) tokenizing said questionnaire; thereby producing a plurality of tokens representing said questionnaire;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field

	in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at 2:1-11
(c) storing said plurality of tokens on a computer readable medium on a first computer;	An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database,

where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.
Warthen at ABSTRACT

In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.
Warthen at 2:1-11

FIG. l(a) is a block diagram of one embodiment of an information server 10 that is Internet-based. Information server 10 is an HTTP server, which responds to requests from HTTP clients such as Web browsers. The actual architecture of an Internet HTTP client-server link and HTTP client is not shown, but it should be understood that information server 10 operates on a computer that is designed and configured for Internet traffic, specifically TCP/IP packets encoding HTTP messages and it should be understood that any manner of client computer can be used to operate the Web browser. Since that technology is well known and readily available, it need not be described further herein.

Information server 10 can either be a dedicated computer, a computing device specifically designed to implement the functions of an information server according to the methods described herein, or information server 10 can be in the form of one or more program code modules designed to run on a general purpose HTTP server as a process of that HTTP server. However implemented, information server 10 usually comprises the components shown in FIG. 1, such as an HTTP daemon 12 and a basic set of active HTML pages, including an opening page 14, a question page 16 and an additional answers page 18.
Warthen at 2:55-3:10

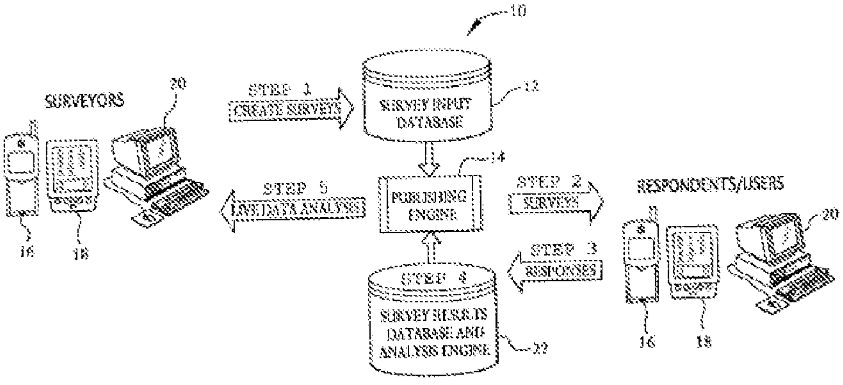
(d) placing a handheld remote computing device into electronic communication with said first computer;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting. alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. **The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA)**. A user of the form. such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Wright at Abstract
(e) transmitting said plurality of tokens to said handheld remote computing device;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting. alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. **The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA)**. A user of the form. such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Wright at Abstract
(f) taking said handheld remote computing device out of electronic communication with said first	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field

computer;	navigation/control (e.g., skip), context sensitive help, data formatting. alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. **The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA)**. A user of the form., such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion. and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at Abstract The present invention is directed to a system and method that allows mobile platforms each supporting any of a variety of call types, and each having differing mobility characteristics, to maintain connectivity to a backbone network in spite of unreliable radio links that occasionally fail. It accomplishes this by using automatic and user-transparent reconnection attempts for appropriate call sessions when an interruption of the link occurs. The network may be supporting a variety of different call types simultaneously. Access to network connectivity resources can be provided according to call session priority based on (for example, call session type, platform mobility, hand-off status, and user class (fee-for-service)) criteria. **The technology allows support of suspended sessions and uses repeated reconnection attempts with priority access to network resources. It also provides for hand-offs of suspended sessions to neighboring gateways as mobile terminals move throughout the service area**. In a network that uses this technology, for example, voice calls (typical of time-sensitive stream traffic) may preempt resources of time-insensitive data calls causing suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at Abstract

	Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than C-C_h active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random. Rappaport at 7:44-63 In a network that employs an admission control protocol according to the present invention, voice calls, for example (or other time-sensitive stream traffic) may preempt resources of time-insensitive data calls, which result in suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can also be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at 2:44-58
(g) after said handheld remote computing device has been taken out of electronic communication with said first computer,	Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than C-C_h active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active

	data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random. Rappaport at 7:44-63 In contrast, mobile users that are engaged in mobile computing (or other forms of data transmission) may have the capability to operate semi-autonomously since data communications with the network are packetized and not necessarily streamed. So with appropriate network design, a temporary disconnection from the network may be transparent to the user. Thus, by implementing the techniques described herein, short term radio link disconnections, which are frequent in mobile communications, need not result in failed sessions, discarded information and wasted use of resources. Rappaport at 5:2-15 In a network that employs an admission control protocol according to the present invention, voice calls, for example (or other time-sensitive stream traffic) may preempt resources of time-insensitive data calls, which result in suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can also be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at 2:44-58
(g1) executing at least a portion of said plurality of tokens representing said questionnaire on said handheld remote computing device to collect a response from a user, and,	Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than $C-C_h$ active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random. Rappaport at 7:44-63

	In contrast, mobile users that are engaged in mobile computing (or other forms of data transmission) may have the capability to operate semi-autonomously since data communications with the network are packetized and not necessarily streamed. So with appropriate network design, a temporary disconnection from the network may be transparent to the user. Thus, by implementing the techniques described herein, short term radio link disconnections, which are frequent in mobile communications, need not result in failed sessions, discarded information and wasted use of resources. Rappaport at 5:2-15 In a network that employs an admission control protocol according to the present invention, voice calls, for example (or other time-sensitive stream traffic) may preempt resources of time-insensitive data calls, which result in suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can also be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at 2:44-58
(g2) storing within said remote computing device said response from the user;	Step 5: Analyzed Data Publishing and Reporting: Once the data has been analyzed in accordance with parameters prescribed by the surveyor, reports and results are published by publishing engine 14. As specified by the surveyor, publishing engine 14 makes the survey results and reports available for access by the surveyor via one or more interface devices (e.g., cell phone 16, PDA 18 or PC 20). The survey results may be "pushed" by the survey publishing system server to the surveyor in the manner known in the art. Alternatively, and preferably, the survey results and reports may passively reside on the survey publishing system server where they may be "pulled" or retrieved by the surveyor. In the latter case, survey publishing system 10 may be programmed to notify the surveyor via electronic mail message or otherwise that the survey results are available for access. It will be understood that publishing engine 14 will publish different levels and depth of data according to the interface device specified by the surveyor to receive the data, e.g., a cell phone will only get high level reports, whereas a PC will receive detailed analysis. Brookler at [0033]

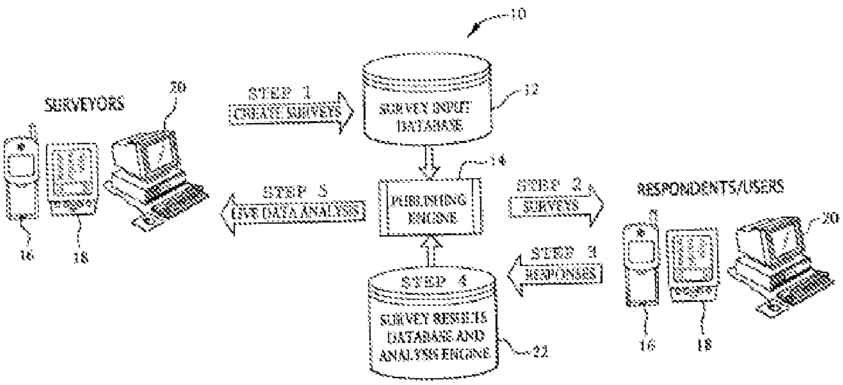
	 10 SURVEYORS 20 STEP 1 CREATE SURVEYS SURVEY INPUT DATABASE 12 14 STEP 5 LIVE DATA ANALYSIS PUBLISHING ENGINE STEP 2 SURVEYS RESPONDENTS/USERS 20 16 18 STEP 3 RESPONSES STEP 4 SURVEY RESULTS DATABASE AND ANALYSIS ENGINE 22 16 18 FIG. 1
(h) placing said handheld remote computing device into electronic communication with a second computer;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled

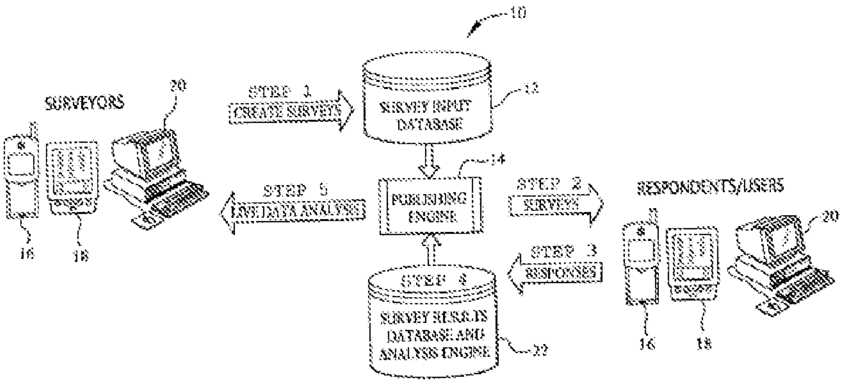
to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.
Warthen at ABSTRACT

In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.
Warthen at 2:1-11

The present invention is directed to a system and method that allows mobile platforms each supporting any of a variety of call types, and each having differing mobility characteristics, to maintain connectivity to a backbone network in spite of unreliable radio links that occasionally fail. It accomplishes this by using automatic and user-transparent reconnection attempts for appropriate call sessions when an interruption of the link occurs. The network may be supporting a variety of different call types simultaneously. Access to network connectivity resources can be provided according to call session priority based on (for example, call session type, platform mobility, hand-off status, and user class (fee-for-service)) criteria. The technology allows support of suspended sessions and uses repeated reconnection attempts with priority access to network resources. It also provides for hand-offs of suspended sessions to neighboring gateways as mobile terminals move throughout the service area. In a network that uses this technology, for example, voice calls (typical of time-sensitive stream traffic) may preempt resources of time-insensitive data calls causing suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that

	is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at ABSTRACT Based on the above, in general, when a request for a new session (voice, data) is received (in step 10), a connection will be established for the new session (affirmative result in step 11) if there are fewer than C-C_h active sessions in the cell. As described above, time-sensitive sessions may be afforded priority. For instance, preferably, voice sessions are given preemptive priority over data sessions for using channel resources. Since voice sessions must be transmitted or received on a real time basis, reconnection attempts for voice sessions are preferably not allowed. When a voice session arrives and finds all channels C occupied, an active data session (if any are present) is preferably suspended (step 12) (or possibly terminated) to accommodate it. More specifically, when a voice session arrives in a cell in which all channels are occupied and fewer than H sessions are in suspension, and, at least, one active session is of data type, an arriving voice session will obtain a connection (step 13) but an active data session will be suspended (step 12). The choice of which data session to be suspended or be terminated is assumed to be random. Rappaport at 7:44-63 In a network that employs an admission control protocol according to the present invention, voice calls, for example (or other time-sensitive stream traffic) may preempt resources of time-insensitive data calls, which result in suspended sessions that do not result in session failures. Priority access for hand-offs of active sessions with respect to new call sessions can also be accommodated. Mobile users that have some autonomy or who are perhaps exchanging time-insensitive data with a remote site can continue to function essentially undisturbed by link failures since the connectivity and reconnection procedures are managed by the network in a manner that is transparent to the end users. Mobile computing sessions and delay-insensitive data communications, for example, will be able to continue, largely unaware of link failures. Rappaport at 2:44-58
(i) transmitting at least a portion of said response stored within said handheld remote computing device to said second computer; and,	Step 5: Analyzed Data Publishing and Reporting: Once the data has been analyzed in accordance with parameters prescribed by the surveyor, reports and results are published by publishing engine 14. As specified by the surveyor, publishing engine 14 makes the survey results and reports available for access by the surveyor via one or more interface devices (e.g., cell phone 16, PDA 18 or PC 20). The survey results may be "pushed" by the survey publishing system server to the surveyor in the manner known in the art. Alternatively, and preferably, the survey results and reports may passively reside on the survey publishing system server

	where they may be "pulled" or retrieved by the surveyor. In the latter case, survey publishing system 10 may be programmed to notify the surveyor via electronic mail message or otherwise that the survey results are available for access. It will be understood that publishing engine 14 will publish different levels and depth of data according to the interface device specified by the surveyor to receive the data, e.g., a cell phone will only get high level reports, whereas a PC will receive detailed analysis. Brookler at [0033]  10 SURVEYORS 20 STEP 1 CREATE SURVEYS SURVEY INPUT DATABASE 12 14 STEP 5 LIVE DATA ANALYSIS PUBLISHING ENGINE STEP 2 SURVEYS RESPONDENTS/USERS 20 16 18 STEP 3 RESPONSES STEP 4 SURVEY RESULTS DATABASE AND ANALYSIS ENGINE 22 16 18 FIG. 1
(j) forming a visually perceptible report from any of said at least a portion of said response so transmitted.	Step 5: Analyzed Data Publishing and Reporting: Once the data has been analyzed in accordance with parameters prescribed by the surveyor, reports and results are published by publishing engine 14. As specified by the surveyor, publishing engine 14 makes the survey results and reports available for access by the surveyor via one or more interface devices (e.g., cell phone 16, PDA 18 or PC 20). The survey results may be "pushed" by the survey publishing system server to the surveyor in the manner known in the art. Alternatively, and preferably, the survey results and reports may passively reside on the survey publishing system server where they may be "pulled" or retrieved by the surveyor. In the latter case, survey publishing system 10 may be programmed to notify the surveyor via electronic mail message or otherwise that the survey results are available for access. It will be understood that publishing engine 14 will publish different levels and depth of data according to the interface device specified by the surveyor to receive the data, e.g., a cell phone will only get high level reports, whereas a PC will receive detailed analysis. Brookler at [0033]

	 10 SURVEYORS 20 STEP 1 CREATE SURVEYS SURVEY INPUT DATABASE 12 14 STEP 5 LIVE DATA ANALYSIS PUBLISHING ENGINE STEP 2 SURVEYS RESPONDENTS/USERS 20 16 18 STEP 3 RESPONSES STEP 4 SURVEY RESULTS DATABASE AND ANALYSIS ENGINE 22 16 18 FIG. 1
CLAIM 12	
12. A method for collecting survey data from a user according to claim 11, wherein step (j) comprises the step of printing a report from any of said response to transmitted.	When server computer 121 receives the information, server computer 121 executes a common gateway interface application (CGI) pointed to by the resource locator. The CGI application grabs the necessary information and transmits the information via e-mail to a fax gateway. The fax gateway, upon receipt of the e-mail, converts the information to a fax and sends the information to the specified telephone number. Thus, cellular telephone 100 requires neither a printer connection nor a print driver, but yet can print using the facsimile machine at ABC Designs. Rossmann p. 11, lines 4-8
CLAIM 13	
13. A method for collecting survey data from a user according to claim 11, wherein said first computer and said second computer are a same computer.	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a

	prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10. Wright at 16:41-54 *See also* Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT
CLAIM 14	
14. A method for	A system and method for providing computerized forms completion and

modifying a questionnaire used in data management according to the method of claim 11, further comprising the steps of: (k) making at least one incremental change to a portion of said questionnaire;	processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10. Wright at 16:41-54 *See also* Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific

	embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT
(l) tokenizing said at least one incremental change to said questionnaire;	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10. Wright at 16:41-54

	See also Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT
(m) transmitting at least a portion of said tokens resulting from step (k) to said remote handheld computing device, said transmitted tokens comprising less than the entire tokenized questionnaire; and,	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT

	If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10. Wright at 16:41-54 *See also* Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT
(n) incorporating said transmitted tokens into said questionnaire at said remote computing device, thereby incrementally	A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed

changing said questionnaire.	accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting. Wright at ABSTRACT If the Next button 220 (FIG. 3) was not selected, as determined at state 350, the forms engine 124 advances to a decision state 370 to determine if the user has made changes to the answer box 164 portion of the display (FIG. 3) response(s). If so, the forms engine 124 moves to a "process answer box input" function 372. Function 372 initiates drawing the selections or input made by the user in the answer box portion 164 of the screen display 162 (FIG. 3) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function 372. Function 372 will be further described in conjunction with FIG. 10. Wright at 16:41-54 *See also* Wright at Figures 1-11 An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to

	obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query. Warthen at ABSTRACT

VI. CONCLUSION

The prior art documents presented in the above Request were either not previously considered by the Office or are now being presented in a new light pursuant to MPEP § 2242(II). Claims 1-14 of the '816 patent are not patentable over the prior art documents cited herein. The prior art documents teach the subject matter of the '816 patent in a manner such that substantial new questions of patentability for all claims are raised by this Request.

In view of the foregoing, it is respectfully submitted that substantial new questions of patentability of claims 1-14 of the '816 patent have been raised by this Request. Accordingly, the Office is requested to grant this Request and to initiate reexamination with special dispatch.

Enclosed is a credit card authorization to cover the Fee for reexamination. If this authorization is missing or defective please charge the Fee to the Novak Druce Deposit Account No. 14-1437.

Respectfully submitted,

/Jay J. Guiliano/
Novak Druce & Quigg, LLP
Jay J Guiliano
Reg. No.
Ryan M. Murphy
Reg. No. 66,285

NOVAK DRUCE + QUIGG LLP
1000 Louisiana St.
53rd Floor
Houston, Texas 77002
P: 713-571-3400
F: 713-456-2836

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	SYSTEM AND METHOD FOR DATA MANAGEMENT
First Named Inventor/Applicant Name:	7,822,816 .
Filer:	Tracy Wesley Druce/Stephanie Dominguez
Attorney Docket Number:	20351.RX816

Filed as Large Entity

ex parte reexam Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
REQUEST FOR EX PARTE REEXAMINATION	1812	1	12000	12000
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
		Total in USD ($)		**12000**

Electronic Acknowledgement Receipt

EFS ID:	15423278
Application Number:	90012829
International Application Number:	
Confirmation Number:	6993
Title of Invention:	SYSTEM AND METHOD FOR DATA MANAGEMENT
First Named Inventor/Applicant Name:	7,822,816 .
Customer Number:	13992
Filer:	Tracy Wesley Druce/Stephanie Dominguez
Filer Authorized By:	Tracy Wesley Druce
Attorney Docket Number:	20351.RX816
Receipt Date:	03-APR-2013
Filing Date:	
Time Stamp:	20:43:17
Application Type:	Reexam (Third Party)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$12000
RAM confirmation Number	6373
Deposit Account	141437
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Non Patent Literature	PA-A_Wright_US5704029.pdf	939633 fcb4e76133e5249eebb98ff6409c0e229710b117	no	34
Warnings:					
Information:					
2	Non Patent Literature	PAT-A_7822816.pdf	21402307 56c83d3fdf764efb80fc54265807da21489852a0	no	17
Warnings:					
Information:					
3	Non Patent Literature	OTH-A_Powerpoint_Showing_Unpatentability.pdf	615059 d957e7b9883def144d37b77745ac97f52f20d5b5	no	28
Warnings:					
Information:					
4	Non Patent Literature	PA-C_Warthen_US6584464_.pdf	24104488 578c41fdea8fd8a2e995f978811c48f30cc28ab6	no	22
Warnings:					
Information:					
5	Non Patent Literature	PA-D_Brookler_US20020007303_.pdf	15241949 55b3d7765ba8ce44f22166ecd706348c341dffa9	no	14
Warnings:					
Information:					
6	Non Patent Literature	PA-H_Bowen_US5442786_.pdf	21787815 f51bb8f27e0b4b5e2e2baec52141a64236228a25	no	21
Warnings:					
Information:					
7	Transmittal of New Application	Transmittal_Form.pdf	197006 625d1e2df2ba699866f159f7b5bb36ef7508d10a	no	2
Warnings:					
Information:					
8	Non Patent Literature	PA-B_Rappaport_US6477373_.pdf	3049364 0c564e8eb58624c81122a4c6d95d92f532ff34b0	no	33

Warnings:					
Information:					
9	Foreign Reference	PA-E_Rossmann_EP0779759A2_.pdf	6706893 8db3f65ddf8c63f108827264a9d6973cdd18ce2e	no	105
Warnings:					
Information:					
10	Foreign Reference	PA-F_Benigno_WO9933390_.pdf	4896131 c38b1c2969178593e99d5609b7067d29c4cde32a	no	79
Warnings:					
Information:					
11	Non Patent Literature	PA-G_Falls_US5991771_.pdf	2885149 db01fca6b1010ac92b104dc45d941e205ee41a81	no	28
Warnings:					
Information:					
12	Information Disclosure Statement (IDS) Form (SB08)	IDS.pdf	612512 449cf4165e66b121cf4a71d515e1af3a02be2fd0	no	4
Warnings:					
Information:					
13	Reexam Certificate of Service	Certificate_Service.pdf	79471 034b944406f67b1eda4ab0f7069a4f1e374c4476	no	1
Warnings:					
Information:					
14	Receipt of Orig. Ex Parte Request by Third Party	ExParte_REEXAMINATION_816_tobefiled.pdf	2427420 cf3b9312e7041e9013517c42af02e07603b46629	no	395
Warnings:					
Information:					
15	Fee Worksheet (SB06)	fee-info.pdf	29866 d32d4f15ddf7333ecd2203bb37be3965b934c80e	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	104975063		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PA-A

US005704029A

United States Patent [19]

Wright, Jr.

[11] **Patent Number: 5,704,029**

[45] **Date of Patent: Dec. 30, 1997**

[54] **SYSTEM AND METHOD FOR COMPLETING AN ELECTRONIC FORM**

[75] Inventor: **Gerald V. Wright, Jr.**, San Diego, Calif.

[73] Assignee: **Wright Strategies, Inc.**, La Jolla, Calif.

[21] Appl. No.: **247,777**

[22] Filed: **May 23, 1994**

[51] **Int. Cl.**^{6} **G06F 17/30**

[52] **U.S. Cl.** **395/149**

[58] **Field of Search** 395/146, 148, 395/149

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,603,232	7/1986	Kurland et al.	379/92
4,651,288	3/1987	Zeising	395/117
4,730,253	3/1988	Gordon	364/413.02
4,863,384	9/1989	Slade	434/107
4,937,439	6/1990	Wanninger et al.	235/456
5,047,960	9/1991	Sloan	395/149
5,100,329	3/1992	Deesen et al.	434/327
5,204,813	4/1993	Samph et al.	364/419.2
5,208,907	5/1993	Shelton et al.	395/149
5,225,996	7/1993	Weber	364/550
5,283,861	2/1994	Dangler et al.	395/149
5,335,164	8/1994	Gough, Jr. et al.	364/149

OTHER PUBLICATIONS

Straley, Straley's Programming with Clipper 5.0, 1991, pp. 166–169, 1015–1020, 1079–1107.

Goodman, Danny, "The Newton Shuffle," MacUser, pp. 183–184, Feb. 1994.

Thornton, J., et al. "Using Computers to survey property," Public Finance Account, Feb. 12, 1993 (abstract only).

Brochure entitled "Introducing the ETE Communicator", Ete, Inc., 1993.

Brochure entitled "Newton Software Titles, Starcore", Newton, 1993.

Brochure entitled "Form Factor, Business and Personal Data Collection for PDAs", Meta Pacific, 1993.

Apple Computer, Inc., The NewtonScript Programming Language, Alpha Draft 1.0, Jul. 1993.

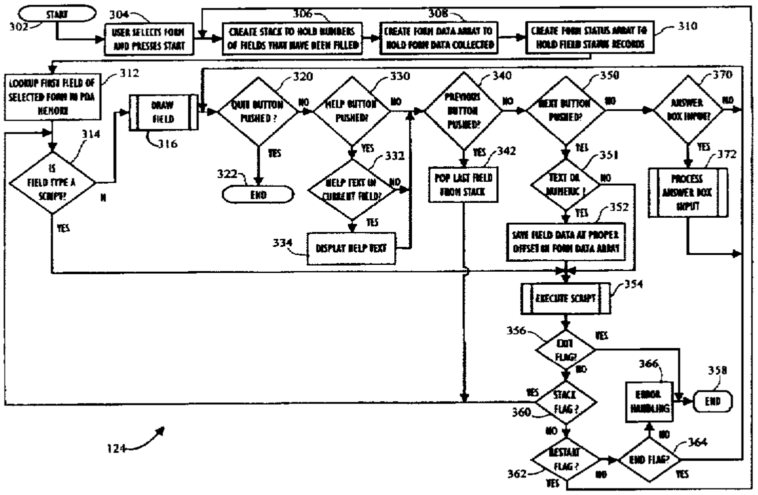
Primary Examiner—Heather R. Herndon
Assistant Examiner—Anton W. Fetting
Attorney, Agent, or Firm—Knobbe, Martens, Olson & Bear

[57] **ABSTRACT**

A system and method for providing computerized forms completion and processing. A forms designer utilizes a forms creation module that includes a scripting feature to create an electronic form. The scripting feature provides flow of control statements and a variety of functions useful in forms, e.g., such as questionnaires. These functions include data validation, field navigation/control (e.g., skip), context sensitive help, data formatting, alert sounds and dialog boxes. The scripting feature ensures that skip patterns are followed correctly and that the form is completed accurately. The forms creation program generates a field description record for each field created by the forms designer. The set of field description records that define the electronic form is then transferred to a handheld computer, such as a personal digital assistant (PDA). A user of the form, such as a respondent to a survey, utilizes the PDA to respond to the statements or questions that are part of the form. A forms engine executing on the PDA interprets one field at a time and displays that field in the sequence designed by the forms designer. Each field includes a prompt portion, an answer box portion, and a control portion that are displayed together on the display screen of the PDA. After the form is completed, the response data is optionally transferred to another computer for further processing or reporting.

23 Claims, 16 Drawing Sheets

Microfiche Appendix Included
(1 Microfiche, 56 Pages)



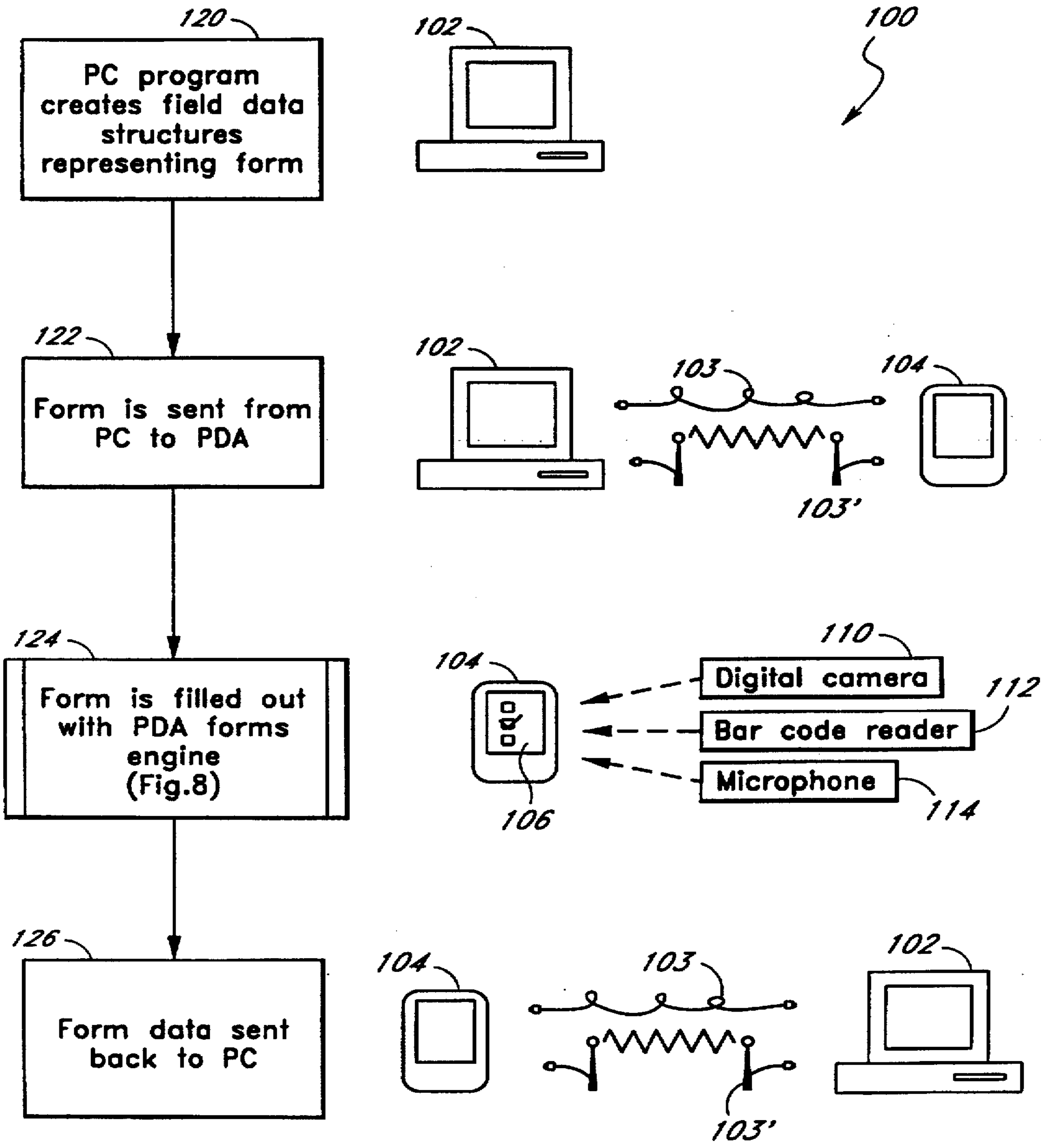

100
120
PC program creates field data structures representing form
102
122
Form is sent from PC to PDA
102
103
104
103'
124
Form is filled out with PDA forms engine (Fig.8)
104
106
110
Digital camera
112
Bar code reader
Microphone
114
126
Form data sent back to PC
104
103
102
103'

Fig. 1

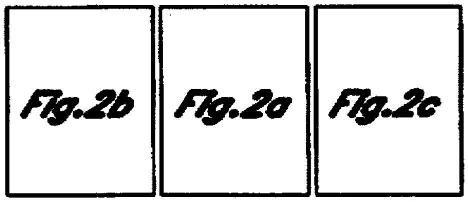
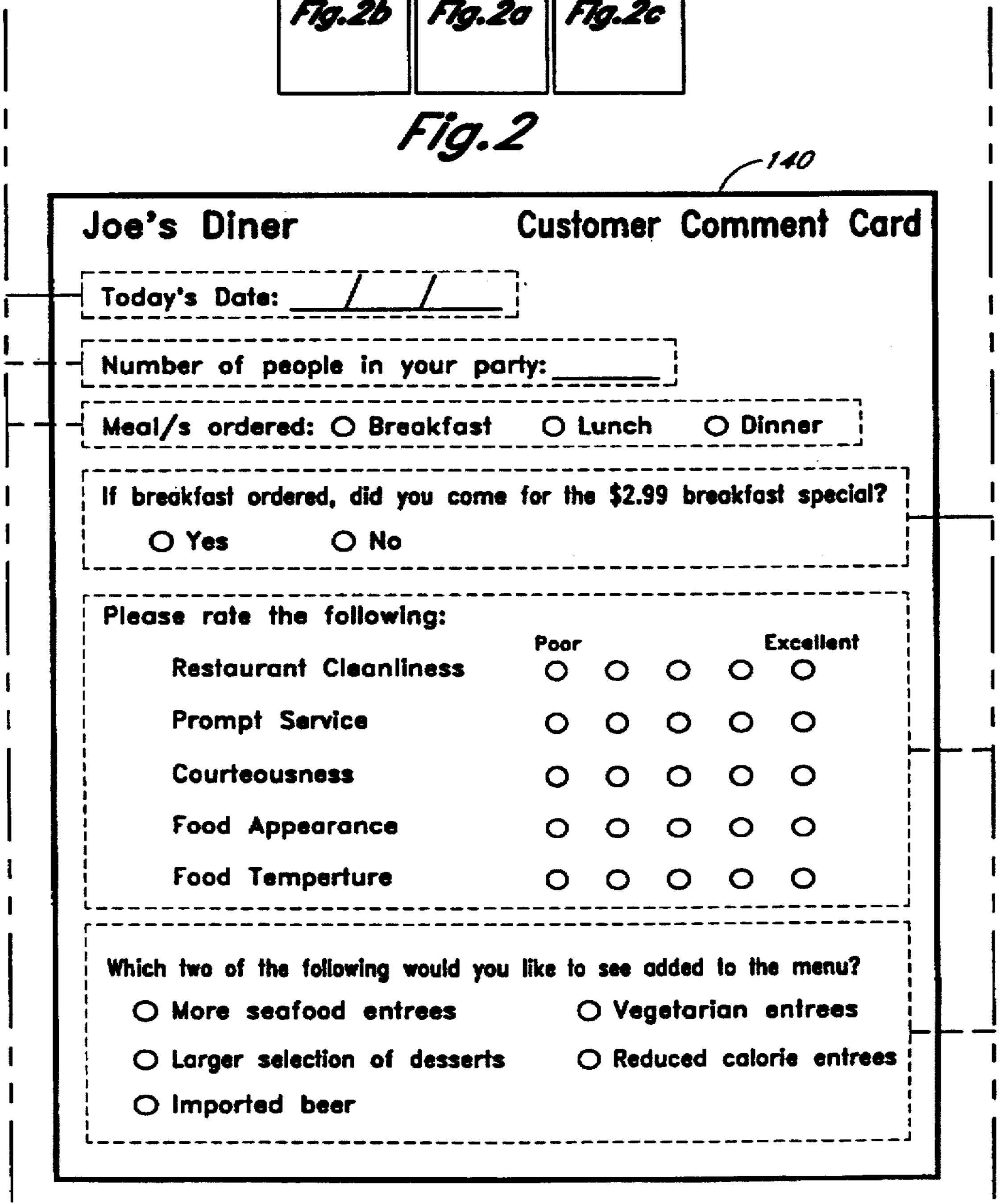


Fig.2



Joe's Diner Customer Comment Card

Today's Date: ___/___/___

Number of people in your party: _______

Meal/s ordered: ○ Breakfast ○ Lunch ○ Dinner

If breakfast ordered, did you come for the $2.99 breakfast special?

○ Yes ○ No

Please rate the following:

	Poor				Excellent
Restaurant Cleanliness	○	○	○	○	○
Prompt Service	○	○	○	○	○
Courteousness	○	○	○	○	○
Food Appearance	○	○	○	○	○
Food Temperture	○	○	○	○	○

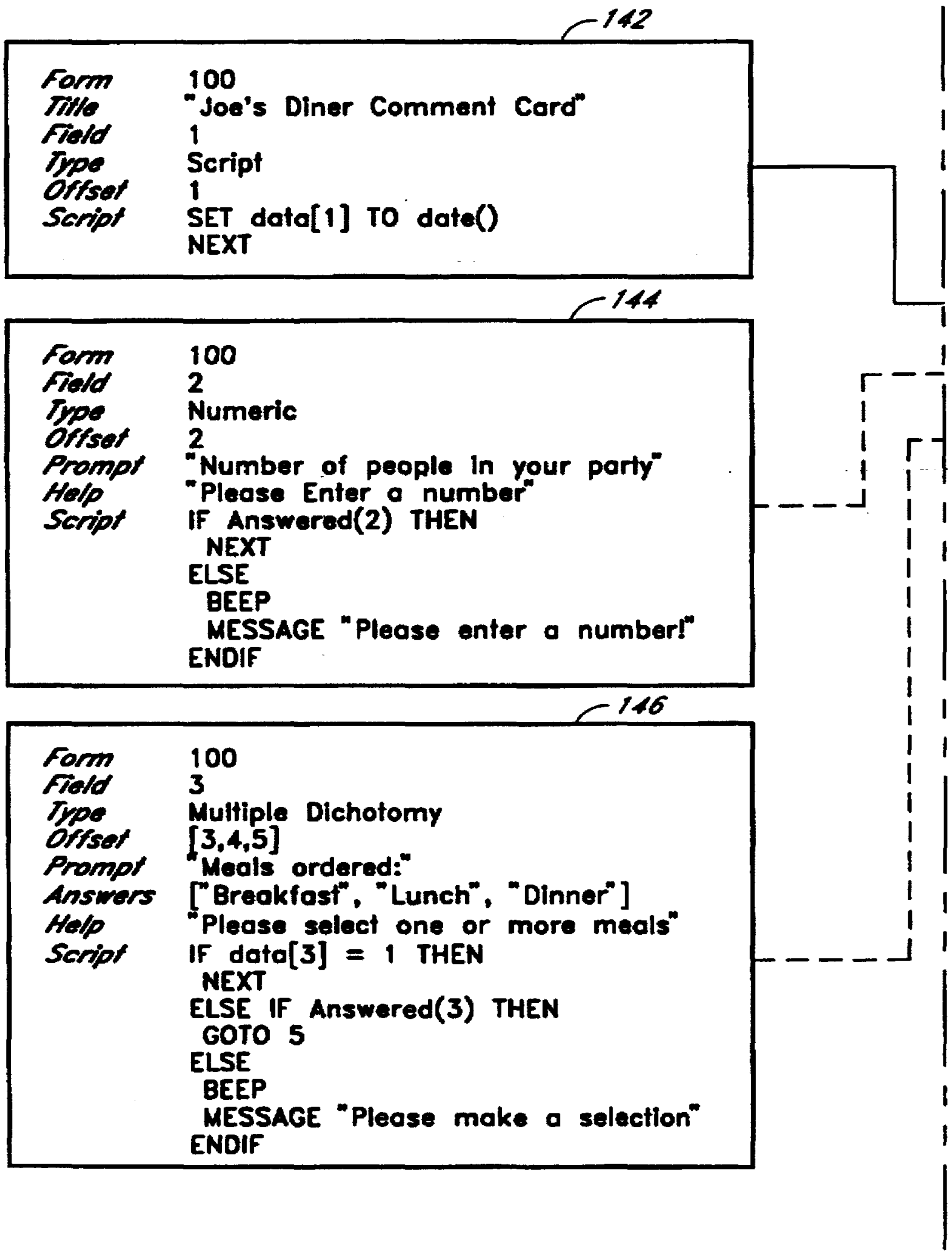
Which two of the following would you like to see added to the menu?

○ More seafood entrees ○ Vegetarian entrees

○ Larger selection of desserts ○ Reduced calorie entrees

○ Imported beer

Fig.2a

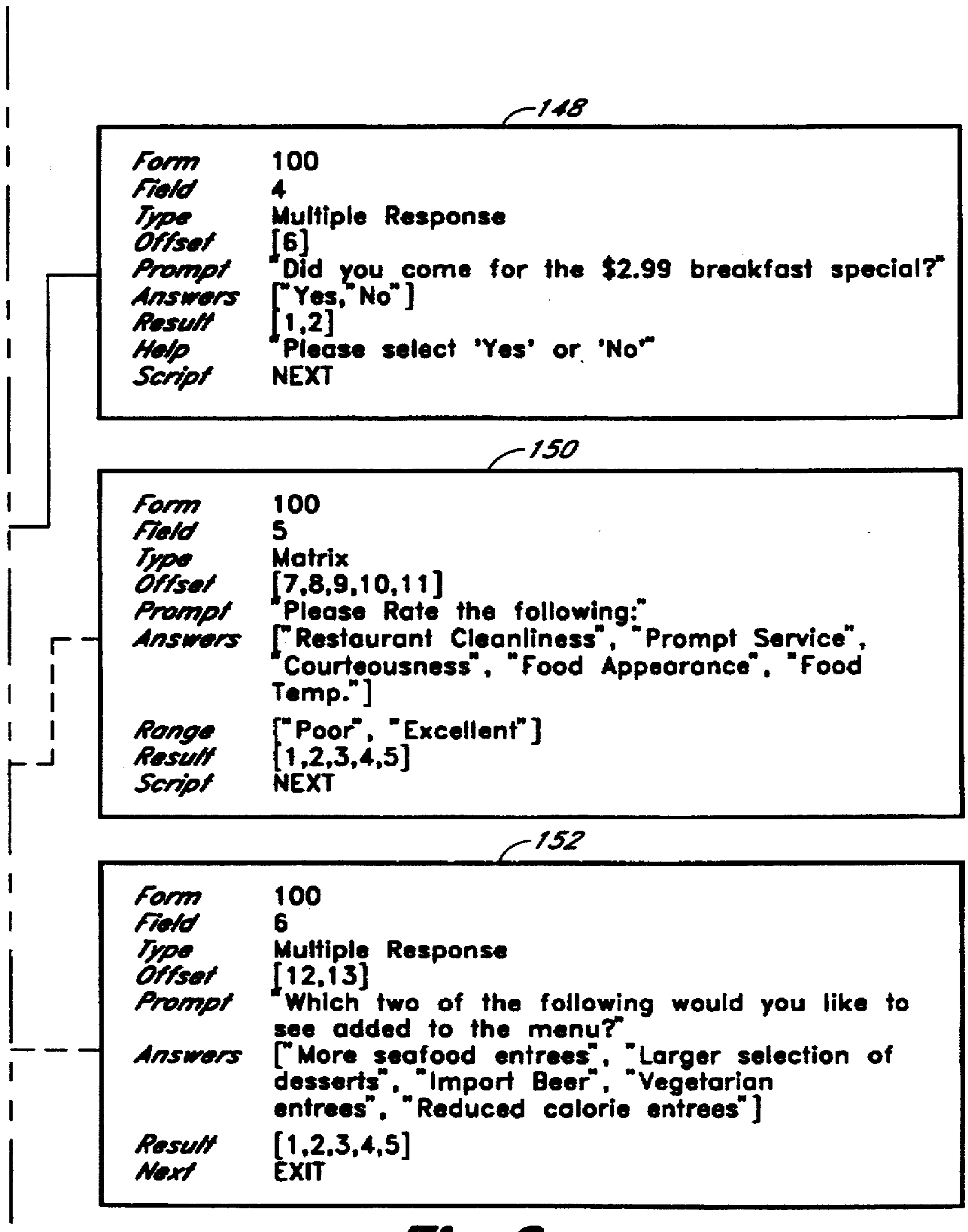


```
Form     100
Title    "Joe's Diner Comment Card"
Field    1
Type     Script
Offset   1
Script   SET data[1] TO date()
         NEXT
```

```
Form     100
Field    2
Type     Numeric
Offset   2
Prompt   "Number of people in your party"
Help     "Please Enter a number"
Script   IF Answered(2) THEN
           NEXT
         ELSE
           BEEP
           MESSAGE "Please enter a number!"
         ENDIF
```

```
Form     100
Field    3
Type     Multiple Dichotomy
Offset   [3,4,5]
Prompt   "Meals ordered:"
Answers  ["Breakfast", "Lunch", "Dinner"]
Help     "Please select one or more meals"
Script   IF data[3] = 1 THEN
          NEXT
         ELSE IF Answered(3) THEN
          GOTO 5
         ELSE
          BEEP
          MESSAGE "Please make a selection"
         ENDIF
```

Fig.2b

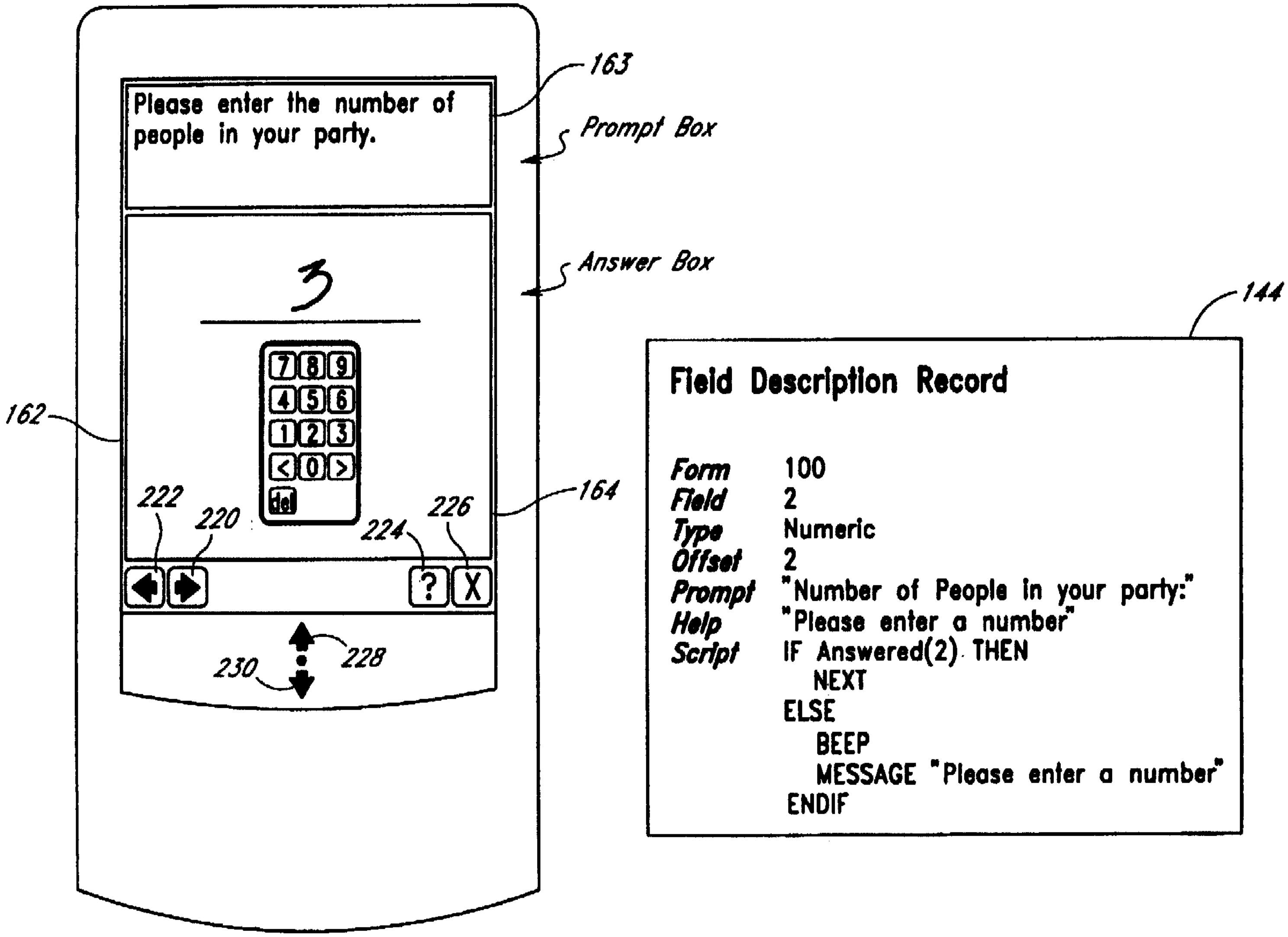


Form	100
Field	4
Type	Multiple Response
Offset	[6]
Prompt	"Did you come for the $2.99 breakfast special?"
Answers	["Yes,"No"]
Result	[1,2]
Help	"Please select 'Yes' or 'No'"
Script	NEXT

Form	100
Field	5
Type	Matrix
Offset	[7,8,9,10,11]
Prompt	"Please Rate the following:"
Answers	["Restaurant Cleanliness", "Prompt Service", "Courteousness", "Food Appearance", "Food Temp."]
Range	["Poor", "Excellent"]
Result	[1,2,3,4,5]
Script	NEXT

Form	100
Field	6
Type	Multiple Response
Offset	[12,13]
Prompt	"Which two of the following would you like to see added to the menu?"
Answers	["More seafood entrees", "Larger selection of desserts", "Import Beer", "Vegetarian entrees", "Reduced calorie entrees"]
Result	[1,2,3,4,5]
Next	EXIT

Fig.2c

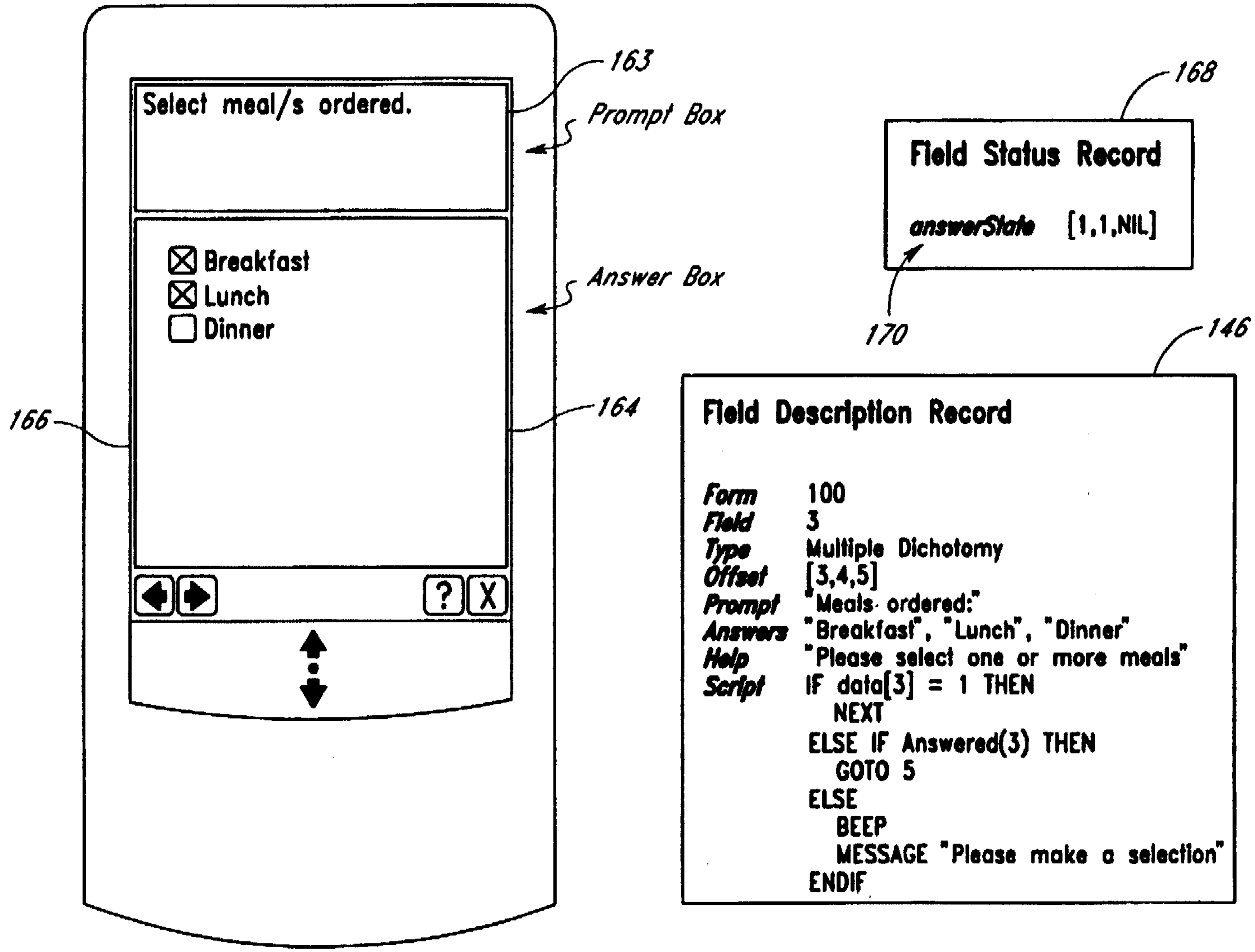


Please enter the number of people in your party.

Field Description Record

```
Form      100
Field     2
Type      Numeric
Offset    2
Prompt    "Number of People in your party:"
Help      "Please enter a number"
Script    IF Answered(2) THEN
             NEXT
          ELSE
             BEEP
             MESSAGE "Please enter a number"
           ENDIF
```

Fig.3



Select meal/s ordered.

☒ Breakfast
☒ Lunch
□ Dinner

Field Status Record

answerState [1,1,NIL]

Field Description Record

```
Form     100
Field    3
Type     Multiple Dichotomy
Offset   [3,4,5]
Prompt   "Meals ordered:"
Answers  "Breakfast", "Lunch", "Dinner"
Help     "Please select one or more meals"
Script   IF data[3] = 1 THEN
           NEXT
         ELSE IF Answered(3) THEN
           GOTO 5
         ELSE
           BEEP
           MESSAGE "Please make a selection"
         ENDIF
```

Fig. 4

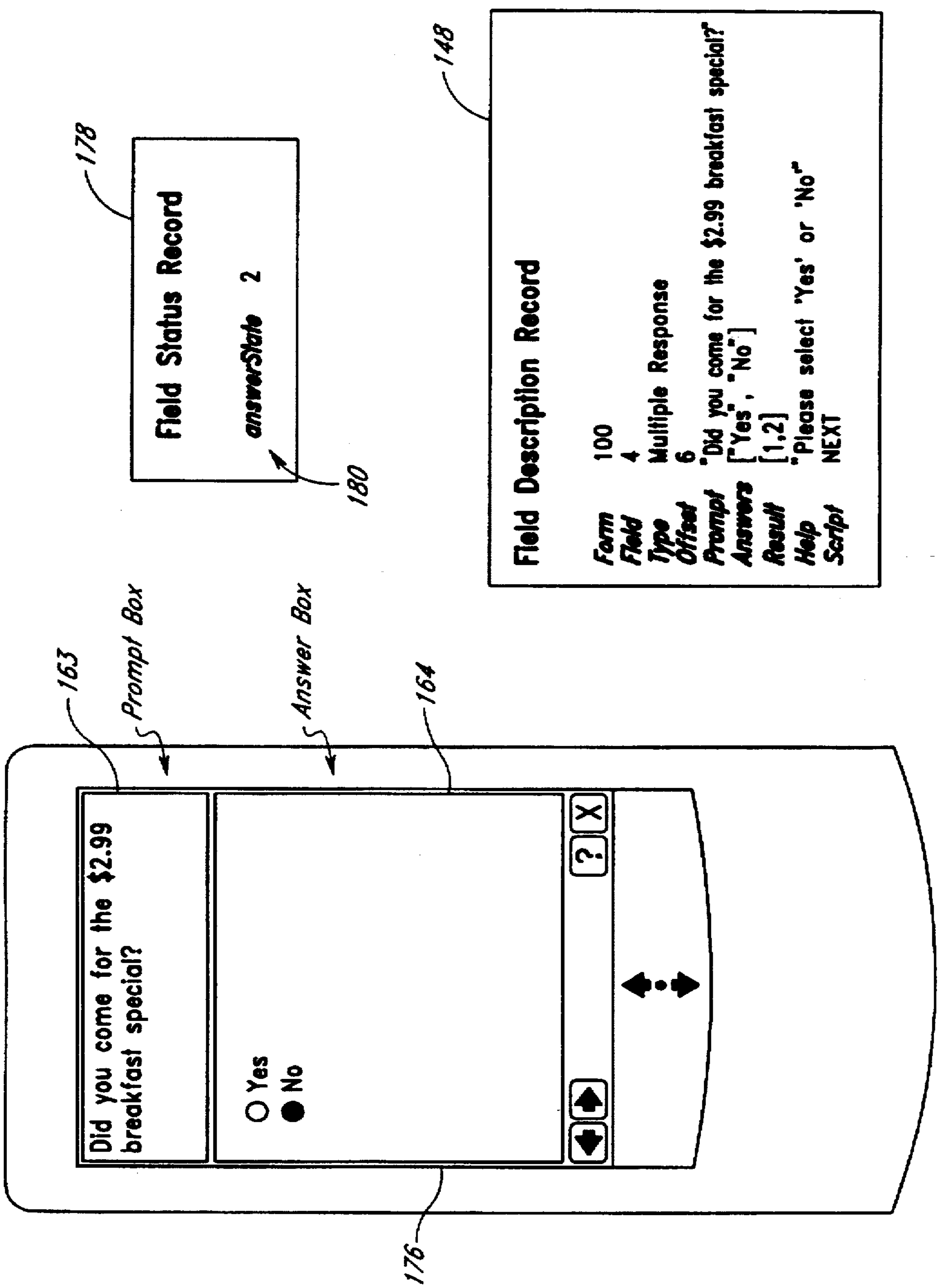
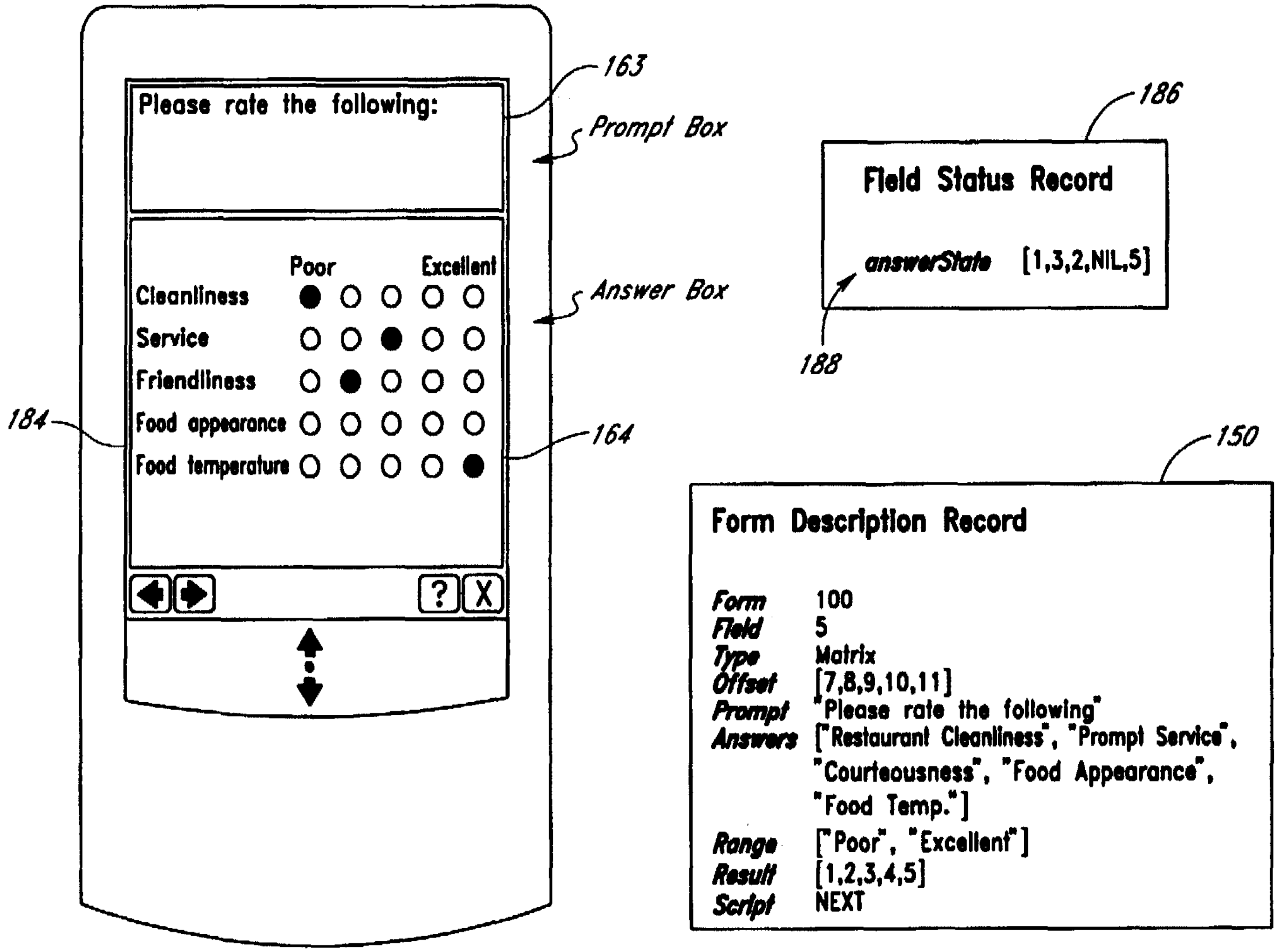

178
Field Status Record
answerState 2
180
148
Field Description Record
Form 100
Field 4
Type Multiple Response
Offset 6
Prompt "Did you come for the $2.99 breakfast special?"
Answers ["Yes", "No"]
Result [1,2]
Help "Please select 'Yes' or 'No'"
Script NEXT
163
Prompt Box
Answer Box
164
Did you come for the $2.99 breakfast special?
○ Yes
● No
?
X
176

Fig.5



Please rate the following:

	Poor				Excellent
Cleanliness	●	○	○	○	○
Service	○	○	●	○	○
Friendliness	○	●	○	○	○
Food appearance	○	○	○	○	○
Food temperature	○	○	○	○	●

Field Status Record

answerState [1,3,2,NIL,5]

Form Description Record

Form	100
Field	5
Type	Matrix
Offset	[7,8,9,10,11]
Prompt	"Please rate the following"
Answers	["Restaurant Cleanliness", "Prompt Service", "Courteousness", "Food Appearance", "Food Temp."]
Range	["Poor", "Excellent"]
Result	[1,2,3,4,5]
Script	NEXT

Fig. 6

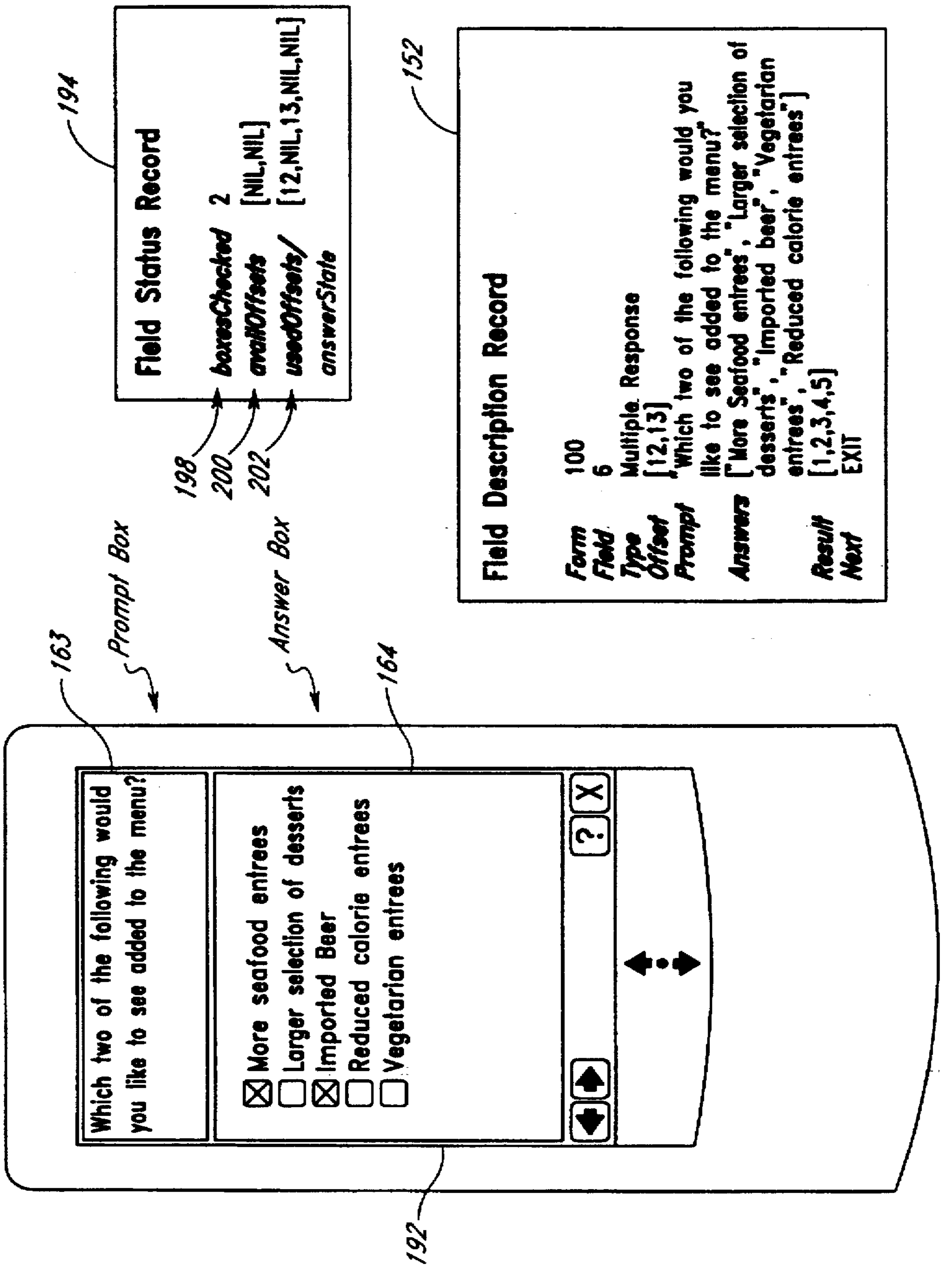

194
Field Status Record
boxesChecked 2
availOffsets [NIL,NIL]
usedOffsets/ [12,NIL,13,NIL,NIL]
answerState
198
200
202
152
Field Description Record
Form 100
Field 6
Type Multiple Response
Offset [12,13]
Prompt "Which two of the following would you like to see added to the menu?"
Answers ["More Seafood entrees", "Larger selection of desserts", "Imported beer", "Vegetarian entrees", "Reduced calorie entrees"]
Result [1,2,3,4,5]
Next EXIT
163
Prompt Box
Answer Box
164
Which two of the following would you like to see added to the menu?
☒ More seafood entrees
□ Larger selection of desserts
☒ Imported Beer
□ Reduced calorie entrees
□ Vegetarian entrees
?
X
192

Fig. 7

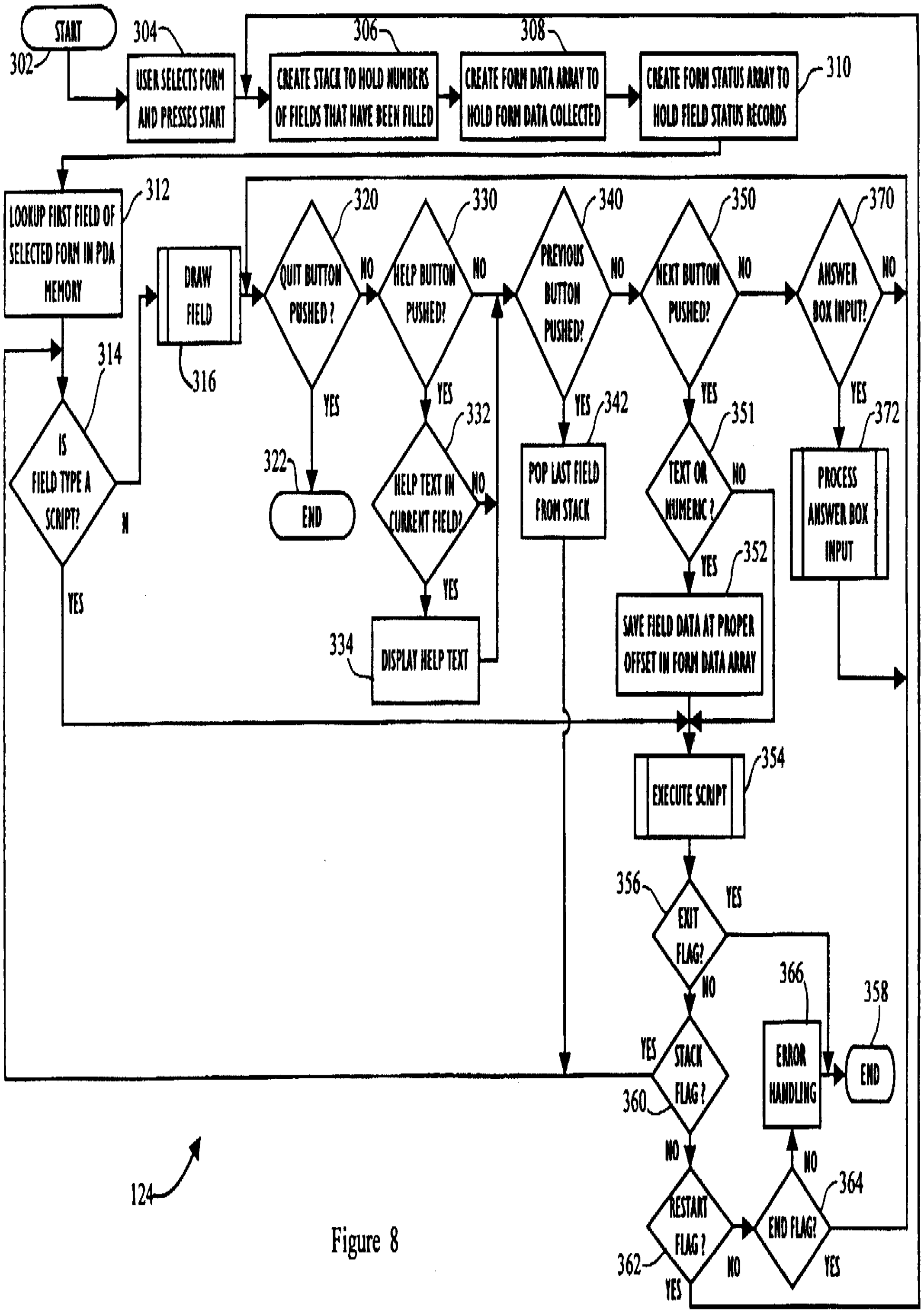

START
302
304
USER SELECTS FORM AND PRESSES START
306
CREATE STACK TO HOLD NUMBERS OF FIELDS THAT HAVE BEEN FILLED
308
CREATE FORM DATA ARRAY TO HOLD FORM DATA COLLECTED
310
CREATE FORM STATUS ARRAY TO HOLD FIELD STATUS RECORDS
312
LOOKUP FIRST FIELD OF SELECTED FORM IN PDA MEMORY
314
IS FIELD TYPE A SCRIPT?
N
YES
DRAW FIELD
316
320
QUIT BUTTON PUSHED ?
NO
YES
322
END
330
HELP BUTTON PUSHED?
NO
YES
332
HELP TEXT IN CURRENT FIELD?
NO
YES
334
DISPLAY HELP TEXT
340
PREVIOUS BUTTON PUSHED?
NO
YES
342
POP LAST FIELD FROM STACK
350
NEXT BUTTON PUSHED?
NO
YES
351
TEXT OR NUMERIC ?
NO
YES
352
SAVE FIELD DATA AT PROPER OFFSET IN FORM DATA ARRAY
370
ANSWER BOX INPUT?
NO
YES
372
PROCESS ANSWER BOX INPUT
354
EXECUTE SCRIPT
356
EXIT FLAG?
YES
NO
366
ERROR HANDLING
358
END
YES
STACK FLAG ?
360
NO
RESTART FLAG ?
362
YES
NO
END FLAG?
NO
364
YES
124

Figure 8

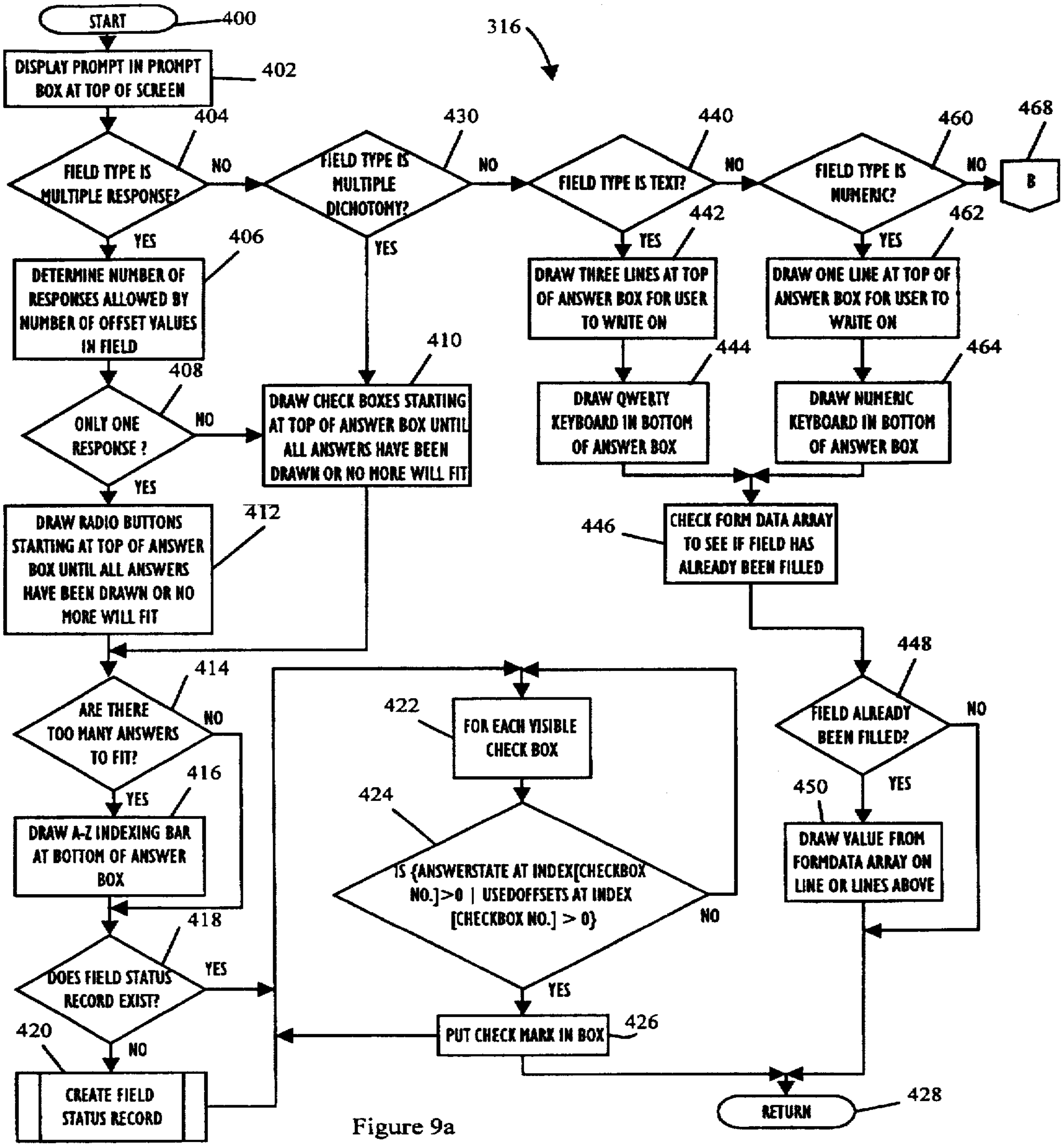

316
START
400
DISPLAY PROMPT IN PROMPT BOX AT TOP OF SCREEN
402
404
FIELD TYPE IS MULTIPLE RESPONSE?
NO
430
FIELD TYPE IS MULTIPLE DICHOTOMY?
NO
440
FIELD TYPE IS TEXT?
NO
460
FIELD TYPE IS NUMERIC?
NO
468
B
YES
406
DETERMINE NUMBER OF RESPONSES ALLOWED BY NUMBER OF OFFSET VALUES IN FIELD
YES
442
YES
DRAW THREE LINES AT TOP OF ANSWER BOX FOR USER TO WRITE ON
462
YES
DRAW ONE LINE AT TOP OF ANSWER BOX FOR USER TO WRITE ON
410
444
DRAW QWERTY KEYBOARD IN BOTTOM OF ANSWER BOX
464
DRAW NUMERIC KEYBOARD IN BOTTOM OF ANSWER BOX
408
ONLY ONE RESPONSE ?
NO
DRAW CHECK BOXES STARTING AT TOP OF ANSWER BOX UNTIL ALL ANSWERS HAVE BEEN DRAWN OR NO MORE WILL FIT
YES
412
DRAW RADIO BUTTONS STARTING AT TOP OF ANSWER BOX UNTIL ALL ANSWERS HAVE BEEN DRAWN OR NO MORE WILL FIT
446
CHECK FORM DATA ARRAY TO SEE IF FIELD HAS ALREADY BEEN FILLED
414
ARE THERE TOO MANY ANSWERS TO FIT?
NO
422
FOR EACH VISIBLE CHECK BOX
448
FIELD ALREADY BEEN FILLED?
NO
416
YES
DRAW A-Z INDEXING BAR AT BOTTOM OF ANSWER BOX
424
IS {ANSWERSTATE AT INDEX[CHECKBOX NO.]>0 | USEDOFFSETS AT INDEX [CHECKBOX NO.] > 0}
NO
450
YES
DRAW VALUE FROM FORMDATA ARRAY ON LINE OR LINES ABOVE
418
DOES FIELD STATUS RECORD EXIST?
YES
YES
420
NO
PUT CHECK MARK IN BOX
426
CREATE FIELD STATUS RECORD
RETURN
428

Figure 9a

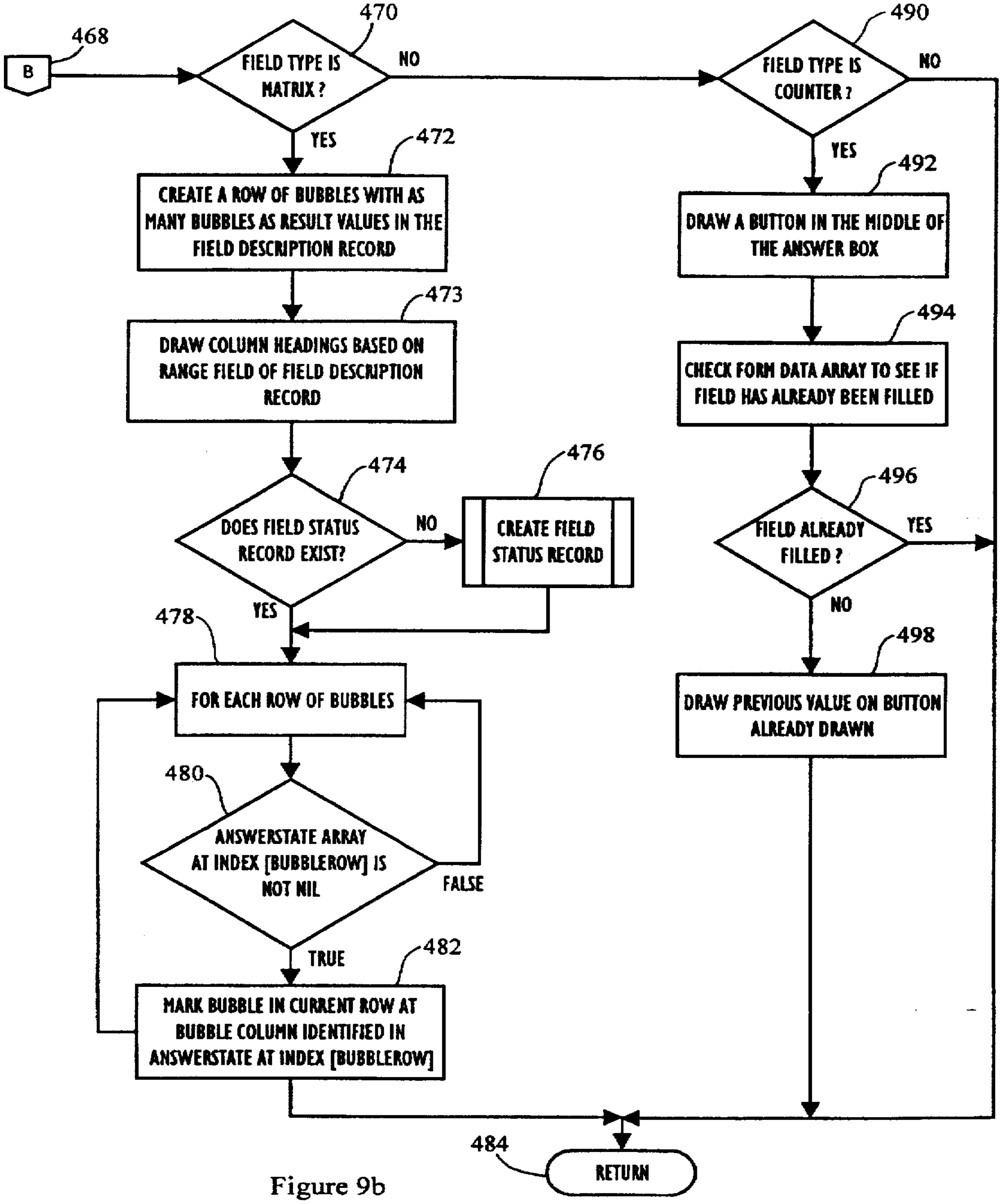

468
B
470
FIELD TYPE IS MATRIX ?
NO
YES
472
CREATE A ROW OF BUBBLES WITH AS MANY BUBBLES AS RESULT VALUES IN THE FIELD DESCRIPTION RECORD
473
DRAW COLUMN HEADINGS BASED ON RANGE FIELD OF FIELD DESCRIPTION RECORD
474
DOES FIELD STATUS RECORD EXIST?
NO
476
CREATE FIELD STATUS RECORD
YES
478
FOR EACH ROW OF BUBBLES
480
ANSWERSTATE ARRAY AT INDEX [BUBBLEROW] IS NOT NIL
FALSE
TRUE
482
MARK BUBBLE IN CURRENT ROW AT BUBBLE COLUMN IDENTIFIED IN ANSWERSTATE AT INDEX [BUBBLEROW]
490
FIELD TYPE IS COUNTER ?
NO
YES
492
DRAW A BUTTON IN THE MIDDLE OF THE ANSWER BOX
494
CHECK FORM DATA ARRAY TO SEE IF FIELD HAS ALREADY BEEN FILLED
496
FIELD ALREADY FILLED ?
YES
NO
498
DRAW PREVIOUS VALUE ON BUTTON ALREADY DRAWN
484
RETURN

Figure 9b

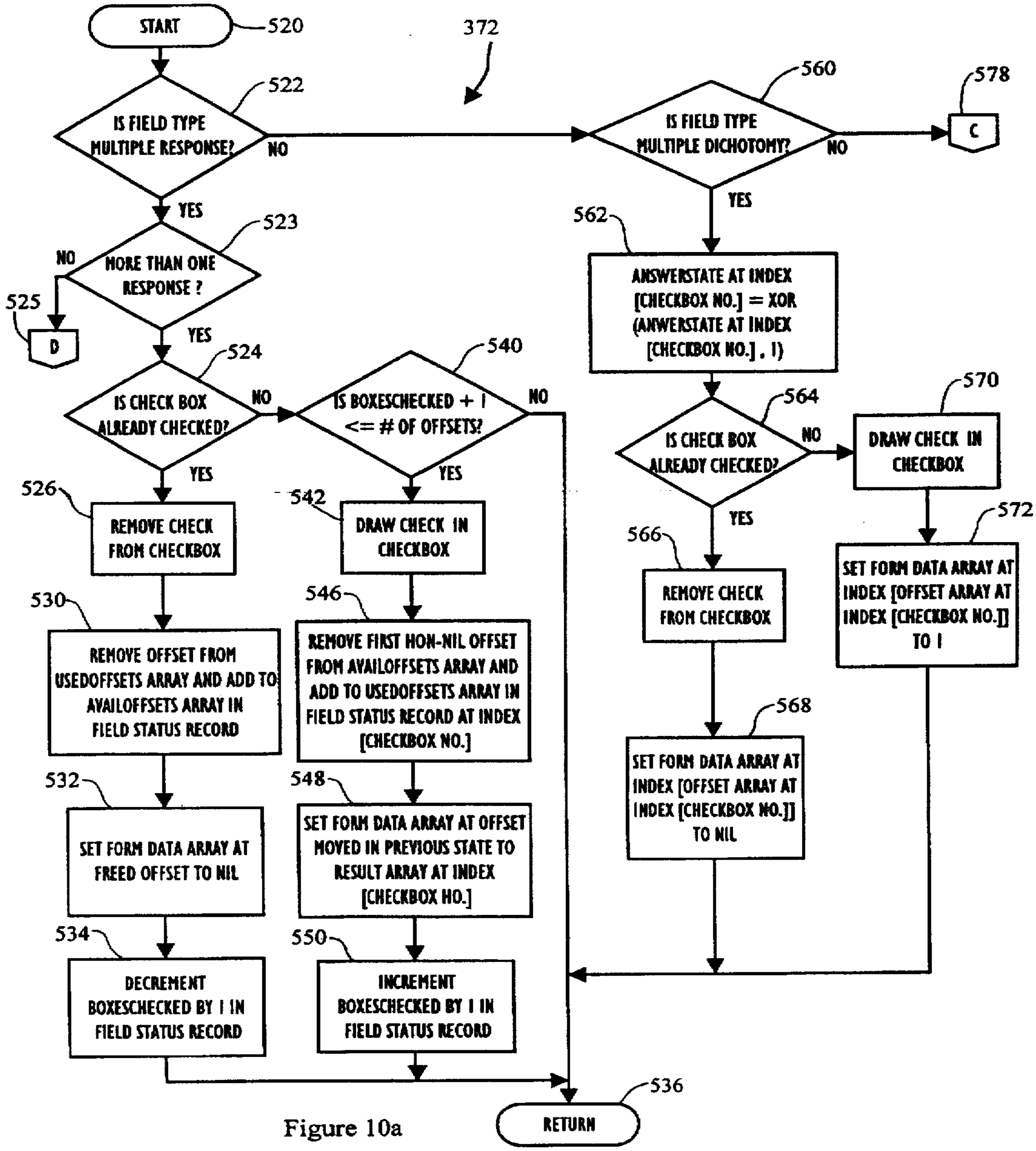

START
520
372
IS FIELD TYPE MULTIPLE RESPONSE?
522
NO
YES
IS FIELD TYPE MULTIPLE DICHOTOMY?
560
NO
C
578
YES
562
ANSWERSTATE AT INDEX [CHECKBOX NO.] = XOR (ANWERSTATE AT INDEX [CHECKBOX NO.] , 1)
MORE THAN ONE RESPONSE ?
523
NO
525
D
YES
IS CHECK BOX ALREADY CHECKED?
524
NO
IS BOXESCHECKED + 1 <= # OF OFFSETS?
540
NO
YES
YES
526
REMOVE CHECK FROM CHECKBOX
542
DRAW CHECK IN CHECKBOX
530
REMOVE OFFSET FROM USEDOFFSETS ARRAY AND ADD TO AVAILOFFSETS ARRAY IN FIELD STATUS RECORD
546
REMOVE FIRST NON-NIL OFFSET FROM AVAILOFFSETS ARRAY AND ADD TO USEDOFFSETS ARRAY IN FIELD STATUS RECORD AT INDEX [CHECKBOX NO.]
532
SET FORM DATA ARRAY AT FREED OFFSET TO NIL
548
SET FORM DATA ARRAY AT OFFSET MOVED IN PREVIOUS STATE TO RESULT ARRAY AT INDEX [CHECKBOX HO.]
534
DECREMENT BOXESCHECKED BY 1 IN FIELD STATUS RECORD
550
INCREMENT BOXESCHECKED BY 1 IN FIELD STATUS RECORD
IS CHECK BOX ALREADY CHECKED?
564
NO
DRAW CHECK IN CHECKBOX
570
YES
566
REMOVE CHECK FROM CHECKBOX
572
SET FORM DATA ARRAY AT INDEX [OFFSET ARRAY AT INDEX [CHECKBOX NO.]] TO 1
568
SET FORM DATA ARRAY AT INDEX [OFFSET ARRAY AT INDEX [CHECKBOX NO.]] TO NIL
RETURN
536

Figure 10a

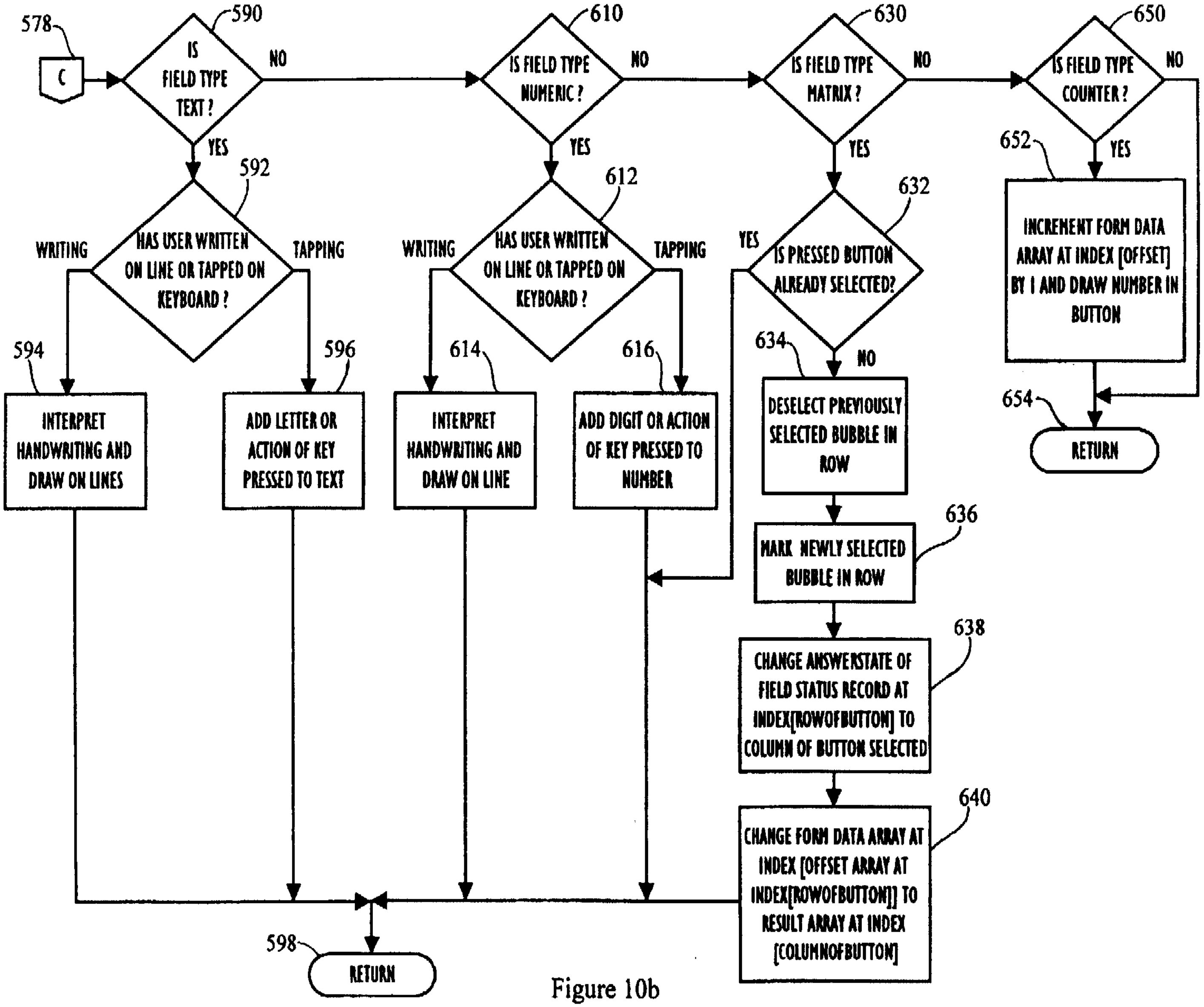

578
C
590
IS FIELD TYPE TEXT ?
NO
YES
592
HAS USER WRITTEN ON LINE OR TAPPED ON KEYBOARD ?
WRITING
TAPPING
594
INTERPRET HANDWRITING AND DRAW ON LINES
596
ADD LETTER OR ACTION OF KEY PRESSED TO TEXT
610
IS FIELD TYPE NUMERIC ?
NO
YES
612
HAS USER WRITTEN ON LINE OR TAPPED ON KEYBOARD ?
WRITING
TAPPING
614
INTERPRET HANDWRITING AND DRAW ON LINE
616
ADD DIGIT OR ACTION OF KEY PRESSED TO NUMBER
630
IS FIELD TYPE MATRIX ?
NO
YES
632
IS PRESSED BUTTON ALREADY SELECTED?
YES
NO
634
DESELECT PREVIOUSLY SELECTED BUBBLE IN ROW
636
MARK NEWLY SELECTED BUBBLE IN ROW
638
CHANGE ANSWERSTATE OF FIELD STATUS RECORD AT INDEX[ROWOFBUTTON] TO COLUMN OF BUTTON SELECTED
640
CHANGE FORM DATA ARRAY AT INDEX [OFFSET ARRAY AT INDEX[ROWOFBUTTON]] TO RESULT ARRAY AT INDEX [COLUMNOFBUTTON]
650
IS FIELD TYPE COUNTER ?
NO
YES
652
INCREMENT FORM DATA ARRAY AT INDEX [OFFSET] BY 1 AND DRAW NUMBER IN BUTTON
654
RETURN
598
RETURN

Figure 10b

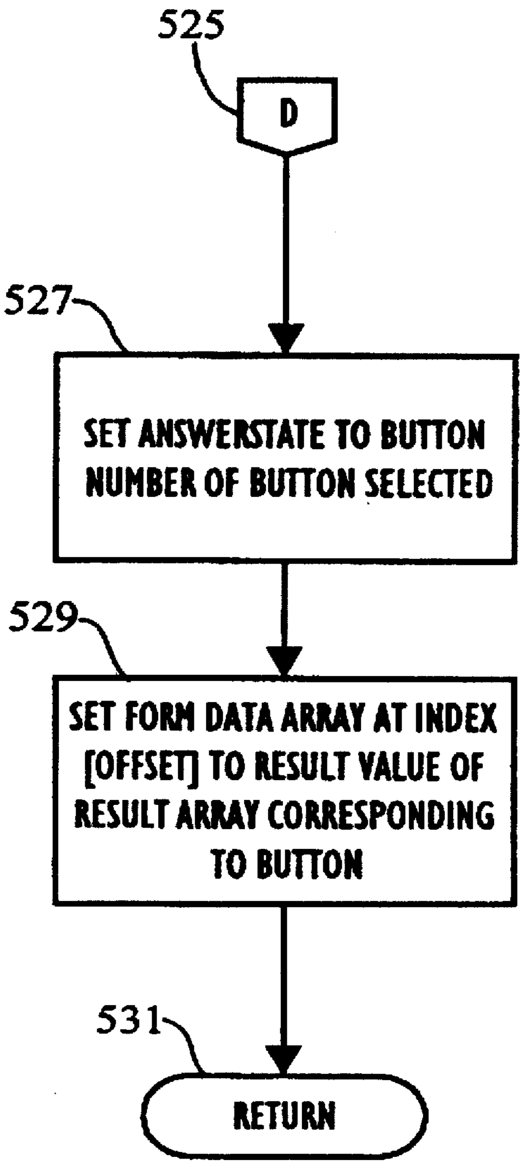

525
D
527
SET ANSWERSTATE TO BUTTON NUMBER OF BUTTON SELECTED
529
SET FORM DATA ARRAY AT INDEX [OFFSET] TO RESULT VALUE OF RESULT ARRAY CORRESPONDING TO BUTTON
531
RETURN

Figure 10c

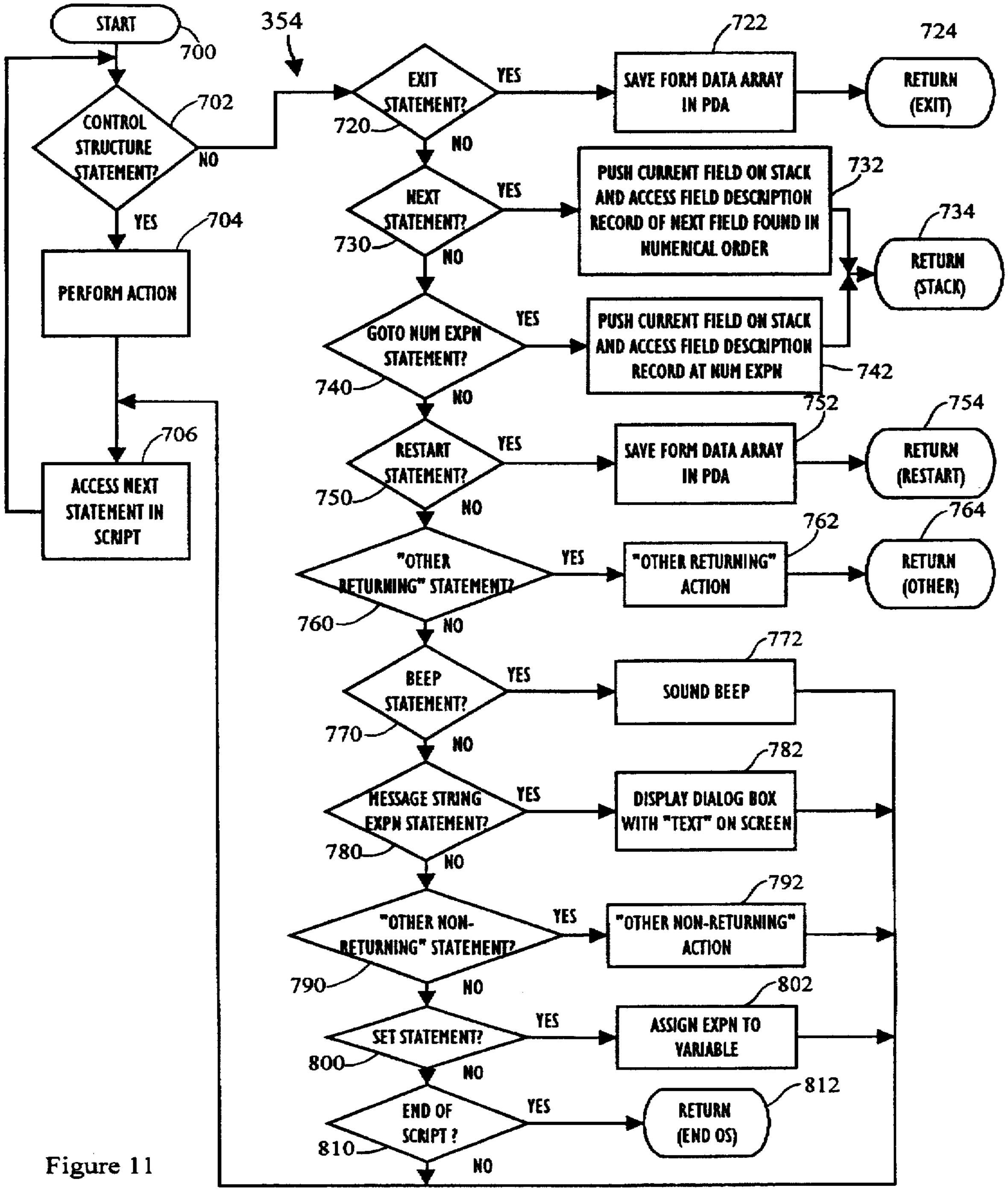

354
START
700
CONTROL STRUCTURE STATEMENT?
702
YES
NO
PERFORM ACTION
704
ACCESS NEXT STATEMENT IN SCRIPT
706
EXIT STATEMENT?
720
YES
NO
SAVE FORM DATA ARRAY IN PDA
722
RETURN (EXIT)
724
NEXT STATEMENT?
730
PUSH CURRENT FIELD ON STACK AND ACCESS FIELD DESCRIPTION RECORD OF NEXT FIELD FOUND IN NUMERICAL ORDER
732
RETURN (STACK)
734
GOTO NUM EXPN STATEMENT?
740
PUSH CURRENT FIELD ON STACK AND ACCESS FIELD DESCRIPTION RECORD AT NUM EXPN
742
RESTART STATEMENT?
750
SAVE FORM DATA ARRAY IN PDA
752
RETURN (RESTART)
754
"OTHER RETURNING" STATEMENT?
760
"OTHER RETURNING" ACTION
762
RETURN (OTHER)
764
BEEP STATEMENT?
770
SOUND BEEP
772
MESSAGE STRING EXPN STATEMENT?
780
DISPLAY DIALOG BOX WITH "TEXT" ON SCREEN
782
"OTHER NON-RETURNING" STATEMENT?
790
"OTHER NON-RETURNING" ACTION
792
SET STATEMENT?
800
ASSIGN EXPN TO VARIABLE
802
END OF SCRIPT ?
810
RETURN (END OS)
812

Figure 11

SYSTEM AND METHOD FOR COMPLETING AN ELECTRONIC FORM

MICROFICHE APPENDIX

A Microfiche Appendix containing computer source code is attached. The Microfiche Appendix comprises one (1) sheet of microfiche having 56 frames, including one title frame.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to business forms and, more particularly, to systems for electronically creating and completing a business form.

2. Description of the Related Technology

For most people, completing a paper form is a bother. Frequently, many questions do not apply to the person or the situation, but the person completing the form must read all the items on the form to determine what is or is not applicable. To avoid the answering of irrelevant questions, the form may have instructions to skip one or more questions under certain conditions, which may cause confusion for the person filling out the form. Then, because there may be unanswered questions due to the skip instructions, the person completing the form may not be sure that all the necessary items or questions were properly answered.

One common type of form, known as a questionnaire, is used to complete a survey. For example, an amusement park operator may want to determine from survey information which rides are popular, why people came on that particular day, and so forth. There are basically two ways survey information is acquired in the field and then processed. The first, and perhaps oldest, method of recorded survey is simply to distribute copies of a paper form which are filled out by hand and, at a later time, entered into a computer by a typist. This is obviously an inefficient approach as data must be entered twice, once during the survey itself and again when the data is entered into the computer. It is also a process which allows two opportunities for error.

A more modern and widely used method utilizes Scantron/National Computer Systems (NCS) technology. This is the approach taken by most field survey data collectors today. In this situation the questionnaire is printed according to stringent technical specifications in a special form which is marked with a #2 pencil during the survey and then fed through a scanner utilizing Optical Mark Recognition (OMR)/Optical Character Recognition (OCR). An example of OMR is a Scantron/NCS scanning machine. The Scantron/NCS machine, hereafter referred to as the 'scanner', then compiles all survey results into a computer file for use in analysis. OCR scanners are also used. OCR forms are also specially printed, but can be written on with block letters.

The scanner approach can provide an order of magnitude increase in survey efficiency and accuracy. Since such forms no longer need to be manually entered into the computer, the significant accuracy loss associated with this process is also eliminated. The high cost of entering the data is also eliminated.

However, the scanner approach also has several serious limitations. First, the questionnaire forms are very expensive. Companies doing even a small number of surveys are incurring costs in the thousands on form duplication alone. Second, although accuracy is generally enhanced, the opportunity for human error still exists during the survey itself. Many forms, especially questionnaires, contain complex sequences of jumps to other parts of the form in the questionnaire depending on information filled in. For instance, if a question asks for gender, the subsequent skip instruction could be "If you answered male to the previous question, go to question 10" and the answer to question **10** may also be the subject of a skip instruction. These instructions are collectively known as skip patterns. Often times survey takers will become lost trying to follow the skip pattern and will answer a question that should not be answered. The only time this mistake can be caught is in the scanning process. The scanner software can perform data validation checks to make sure the survey taker has not answered any questions he shouldn't have, or to check that the answer to a particular questions falls within a certain set of values. However, such post-survey validation takes time.

It takes time for someone to pull out a bad form, try to determine the error from a message on a computer screen, change the form, and re-scan it. Furthermore, it may be impossible to determine what the error was, and the form may thus have to be discarded. As a third limitation, creating new questionnaires is time consuming and expensive. Because the scanner forms must follow stringent technical specifications to be read by the scanner, expensive artwork is involved in creating a new form. To create a new survey, one must create the survey on paper, submit it to the scanner vendor, and pay around $800 for a two-sided 8½×14 form. Once it is created and thousands have been ordered, it is usually no longer economical or feasible to make any changes to the form itself.

Other ways of improving the completion of forms have been proposed. One such system is described in U.S. Pat. No. 4,937,439 to Wanninger, et al. ("Wanninger") wherein a desktop survey system for creating and scanning a survey form to be completed by a survey respondent is described. The survey forms are printed on a scannable form having a preprinted timing track which is scanned by an optical mark scanner. The system also includes a processor for entering and editing customized questions and corresponding response areas and for tabulating and analyzing the scanned results.

Other forms software packages are available for personal computers. The limitations of these forms packages include the following. Either the entire form is displayed on a monitor, or a section of the form is displayed at a time as the user scrolls the display to show the other pieces of the form. The latter situation is more common in a handheld computer environment, such as a personal digital assistant (PDA), because of the smaller screen size as compared to a PC monitor. In either case, the user's attention is not fixed on a single field of the form. To conserve space on a typical form created by a forms package, one or more levels of menus and/or dialog boxes must be selected. Thus, a discrete set of choices or answers are not shown on the form.

Another limitation based on the display of the entire form is that the order of completing the form cannot be controlled. A user may choose to fill in responses in an arbitrary order which may lead to incorrect results. A further limitation is the inability to easily and automatically backtrack through the completed sequence of items to correct a previous answer.

Thus a need exists for a system which (1) inexpensively creates new and modifies existing forms, (2) automatically handles skip patterns, (3) performs error validation as the form is filled out, (4) reduces error by limiting the presentation of information to a survey taker, and (5) is mobile.

SUMMARY OF THE INVENTION

The present solution to the problem of creating and accurately, quickly, and completely filling-in a business form is a computerized forms engine and system designed to automate and simplify the process. The goal of the forms system is to accurately and easily gather information and present it for further processing in a known format, or to even electronically send the gathered information to a desired party.

For example, a company may use an electronic form version of a product registration card. When the end user completes or fills out the registration form, the forms engine verifies the form information and then electronically forwards the information to the company in a format desired and recognized by the computer of the company. Thus, no paper forms are needed, the mail system is not needed, people to sort the mail at the company are not needed, and people to enter the form information into the company computer are not needed. These benefits all lead to savings in resources, overhead and money.

In the presently preferred embodiment of the invention, a personal computer (PC) program (e.g., running on a Apple Macintosh, or IBM compatible) creates a set of field structures representing the electronic form. The data representing the electronic form is sent from the PC to a portable computerized device, such as a Personal Digital Assistant (PDA) that has a graphics display. The electronic form is completed by a user of the PDA, and then the forms engine verifies the input data or information. The PDA sends a form data array containing the responses by the user to the computer. The first item in the form data array is a form identification so that the computer can identify the form that the data is associated with.

The forms engine presents a single item or question to the person, using as much of the display screen as is needed. Then, based on user-defined script contained in the form, the forms engine presents the next item or question to the person or beeps or displays a message. When a person answers a question and desires to move to the next field, the forms engine 124 executes a script written by the form designer. Although the most expected and common result of the script will be to advance to another question, it may also perform other actions such as beep and display a message, quit, or launch another form. This continues until all the pertinent questions have been answered, and only the questions applicable to the person or situation will be asked. Each question corresponds to a certain location on a hard copy version of the electronic form.

One aspect of the present invention includes, in an automated forms system including a portable computer, a method of displaying a response driven sentence, comprising the steps of providing an electronic form comprising a plurality of form descriptors, wherein each form descriptor defines a displayable item; displaying only one of the displayable items on a graphics display of the computer, wherein the displayable item includes a sentence and a plurality of possible responses, and wherein the one displayable item utilizes the entire display; and receiving one or more response entries to the displayable item selected by a user of the computer.

Another aspect of the present invention includes, in an automated forms system including a portable computer, a method of completing an electronic form comprising a plurality of form descriptors that define displayable items, comprising the steps of displaying only one of the displayable items on a graphics display of the computer, wherein the displayable item includes a sentence and a plurality of possible responses; receiving one or more response entries to the displayable item selected by a user of the computer; storing the response in the computer; branching to a subsequent displayable item in response to the selected response; and displaying the subsequent displayable item, wherein the subsequent item includes a plurality of possible response entries, on the display of the computer.

Yet another aspect of the present invention includes, in an automated forms system including a portable computer, a method of completing an electronic form, comprising a plurality of form descriptors that define displayable items, comprising the steps of displaying only one of the displayable items on a graphics display of the computer at any one time, wherein the displayable item includes a sentence and a plurality of possible responses; receiving one or more user selected response entries to the displayable item; storing the response in the computer; and validating the response according to a criteria.

Still yet another aspect of the present invention includes, in an automated forms system including a portable computer, a method of completing an electronic form, comprising a plurality of form descriptors that define displayable items, comprising the steps of displaying only one of the displayable items on a graphics display of the computer, wherein the displayable item includes a sentence and a plurality of possible responses, and wherein the one displayable item utilizes the entire display; receiving one or more response entries to the displayable item selected by a user of the computer; storing the responses in the computer; and automatically sending the responses to a remote computer via a communications device.

Yet another aspect of the present invention includes, in an automated forms system including a portable computer, a method of completing an electronic form, comprising the steps of providing the electronic form comprising a form descriptor that defines a displayable item; and displaying the one displayable item on a graphics display of the computer, wherein the screen displays a button and a number, and wherein the number is changed when the button is selected by a user of the computer.

Yet another aspect of the present invention includes a system for completing an electronic form, comprising a portable unit including a processor, a graphics display, an input device mechanism and a memory; an electronic form stored in the memory of the unit comprising a plurality of form descriptors that define displayable items; a forms engine to display a single displayable item on the graphics display described by one of the form descriptors, wherein the displayable item includes a sentence and a set of possible response entries, and wherein the one displayable item utilizes the entire display; and wherein the input device mechanism accepts one or more of the responses selected by a user of the unit.

Another aspect of the present invention includes a system for completing an electronic form, comprising means for processing and storing data; means connected to the processing means for displaying graphics; means for defining an electronic form including a plurality of displayable items; means for displaying a single one of the displayable items on the graphic means, wherein the displayable item includes a sentence and a set of possible response entries, and wherein the one displayable item utilizes the entire graphic means; and input means connected to the processing means for accepting one or more of the responses selected by a user of the system.

Still yet another aspect of the present invention includes, in a forms system including a computer and a portable computer, a method of automated forms completion, comprising the steps of generating an electronic form using the computer; sending data defining the electronic form from the computer to the portable computer; completing the electronic form, so that response data is generated and stored in the portable computer; and sending the response data from the portable computer to the computer.

Another aspect of the present invention includes, in an automated forms system including a portable computer, a method of completing an electronic form comprising a plurality of form descriptors, each form descriptor defining a displayable item, the system comprising the steps of displaying a selected one of the displayable items on a graphics display of the computer, wherein the displayable item includes a sentence and a plurality of possible responses, and wherein only the one displayable item is visible at any one time; receiving one or more user selected response entries to the selected displayable item; and providing a script stored in the form descriptor defining the selected displayable item, wherein the script defines a set of one or more actions to be executed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a high-level diagram illustrating the basic steps and components of a presently preferred embodiment of the forms system of the present invention;

FIG. **2** (comprising FIG. **2***a*, **2***b* and **2***c*) is exemplary field data structure printed form and a set of exemplary field data structures for the form as used in the system of FIG. **1**;

FIG. **3** is a diagram of an exemplary screen display as presented on the display of the handheld computer for the field **2** data structure and the associated field description record of the exemplary form of FIG. **2**;

FIG. **4** is a diagram of an exemplary screen display as presented on the display of the handheld computer for the field **3** data structure, and the associated field description record and a field status record of the exemplary form of FIG. **2**;

FIG. **5** is a diagram of an exemplary screen display as presented on the display of the handheld computer for the field **4** data structure, and the associated field description record and a field status record of the exemplary form of FIG. **2**;

FIG. **6** is a diagram of an exemplary screen display as presented on the display of the handheld computer for the field **5** data structure, and the associated field description record and a field status record of the exemplary form of FIG. **2**;

FIG. **7** is a diagram of an exemplary screen display as presented on the display of the handheld computer for the field **6** data structure, and the associated field description record and a field status record of the exemplary form of FIG. **2**;

FIG. **8** is a top-level flow diagram of the form engine process performed while the form is filled out by the "Form is filled out with PDA" step of FIG. **1**;

FIGS. **9***a* and **9***b* are a flow diagram of the "draw field" function **316** presented in FIG. **8**;

FIGS. **10***a*, **10***b* and **10***c* are a flow diagram of the "Process answer box input" function **372** presented in FIG. **8**; and

FIG. **11** is a flow diagram of the "execute script" function **354** presented in FIG. **8**.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The following detailed description of the preferred embodiments presents a description of certain specific embodiments to assist in understanding the claims. However, the present invention can be embodied in a multitude of different ways as defined and covered by the claims.

For convenience, the following description is topicalized into the following principal sections: Introduction, System Overview, Forms Creation, Forms Engine, Drawing Fields, Answer Box Input, Execute Script, Benefits of the Forms System, Optional System Configuration, and Summary of Advantages of the Present Invention.

I. INTRODUCTION

The present invention utilizes electronic forms to gather information. Electronic forms are easier to complete and can be validated as they are being completed. Paper forms do not have to be printed and stored. Revisions are easily made to an electronic form, thus eliminating the cost associated with a new printing of a paper form.

The system of the present invention is typically used in either a forms creation and forms completion mode, or in just the forms completion mode, if the electronic form already exists. Referring to FIG. **1** showing a presently preferred embodiment, if a new form is necessary, or if a previous form needs to be revised or customized, a forms creator or forms designer creates the new form (state **120**) by use of a forms creation program that runs on an Apple Macintosh or a Microsoft Windows based Personal Computer (PC). The forms creation program is considered to be a module of the forms system. After the form (not shown) is built, form descriptors or field description records, which define each of the form's fields, are transferred at a state **122** to a portable computer **104**, which is henceforth referred to as a personal digital assistant (PDA), such as the Newton, available from Apple Computer, Inc. Newton is a trademark of Apple Computer, Inc. In the instance of the portable computer being characterized as a handheld computer, such as a Newton, the handheld does not have a keyboard and does not have a hinged display (i.e., a hinged display is not integrated with the processor section of the computer). The Newton does include a non-volatile, semiconductor random-access memory (RAM) subsystem. The RAM subsystem includes 640 Kb of static RAM and a 3 volt lithium backup battery. In the presently preferred embodiment, the battery has a lifetime in average use of approximately one year. Non-volatility refers to the ability of a memory to retain data when the main power for the computer is turned off. Thus, the static RAM in conjunction with the backup battery provide a non-volatile RAM subsystem. In another embodiment, data may be stored in non-volatile flash memory.

In a presently preferred embodiment, the transfer at state **122** is accomplished by connecting a serial cable **103** between the computer **102** and the PDA **104**. Alternatively, other communication devices **103'**, such as modems or wireless technology, may be used for communication between the PDA **104** and computer **102**.

Once one or more electronic forms are created and transferred to the PDA **104**, a particular form to be completed is selected by, for example, a data collector (interviewer). The forms creator designs the form and a user of the form, e.g., the interviewer, are separate entities. A paper representation of the form is not used or needed. A forms engine (discussed below with respect to FIG. **8**) displays one question or statement (and if appropriate, a corresponding set of possible answers or responses) at a time on the display screen **106** of the PDA **104** at a function **124**. Note that the forms system can be operated in a portion of the display, which, for example, is commonly implemented as a window.

When the user (not shown) answers each question, thereby providing response data, the forms engine executes a script program for that question that may perform data validation, sound an alarm, display a message, quit, launch another form, skip to another question, and so forth. The script allows the forms designer to go to another question based on the previous answer(s). Thus, complex skip patterns may be utilized without burdening the user about which question is next, what questions are to be left blank or skipped, and so forth. Selections can be made by use of a stylus, such as a pencil, by a finger of the user, or other ways to identify a choice to the PDA **104**.

Optional peripheral input devices which may be connected to the PDA **104** include a digital camera **110** to capture images, a barcode reader **112**, and a microphone **114** for converting sound into an electrical signal. One or more of these external peripherals may be utilized to provide user input during the course of completing a form.

After the user completes a particular form, the collected data, stored in a form data array, is saved in the PDA **104**. The response data may optionally be sent to a host computer **102** at state **126**, which may be different than that used to create the form, for further processing. Various communication methods can be utilized to send the data to the host computer **102**, e.g., a modem.

Thus, the forms system **100** can be used to create "smart" forms which automatically guide a user through the form. The respondent need only be concerned with answering each question on the screen **106** of the PDA **104** and the rest is taken care of by the forms engine.

II. SYSTEM OVERVIEW

Referring to FIG. **1**, the components of a presently preferred embodiment of the computerized forms system **100** of the present invention are shown. The personal computer (PC) **102** may be an IBM compatible PC running Microsoft Windows version 3.1 or higher. To run Windows 3.1, the PC preferably includes an 80386 class processor with at least 4 Megabytes (Mb) of memory and an 80 Mb hard drive. A keyboard, a pointing device, such as a mouse or trackball, or other such input devices are connected to the computer for input and control. An alternate computer is an Apple Macintosh running Apple Operating System level 7.0 or higher. To run System 7.0, the processor is preferably a 68030 class or better. The memory, hard storage and input devices are similar to those for the PC. Of course, other similar computers with appropriate software and capability may be utilized.

The preferred Newton PDA operates under Newton Intelligence version 1.05 or higher operating system. The forms engine software is written in Newton Script programming language using the Newton beta-level tool kit, and is compiled by the Newton tool kit. The Newton Script Programming Language manual (Alpha Draft 1.0) is hereby incorporated by reference. The forms creation software is written in OMNIS 7 version 1.3 and compiled using OMNIS 7. Source code for the form engine software is included in the attached Microfiche Appendix. Of course, other handheld computers and programming languages may be used.

III. FORMS CREATION

The object the forms system **100** works with, or its document type, is the form. Forms may be created, edited, duplicated, and sent. Visually, forms appear on the screen of the PC **102** as a window made up of a list of fields. Within the current form, users may add new fields, edit fields, or remove fields.

A. Form Creation Operations

The forms creation program used to design a new electronic form includes several menus: a "File" menu, a "Connect" menu, and a "Design" menu. Each menu lists a plurality of options representing operations performed in the creation of a form. The forms designer selects from among the options to initiate a desired operation. The options germane to the creation of the form are now briefly discussed below.

Available from "File" menu unless otherwise designated.

New Form

Dialog box prompts user for form name and description. A form number is automatically assigned. A form window is displayed with an empty field list.

Open Form

Dialog box prompts user for form to open from list of existing forms. After selection is made, a window opens containing a form field list.

Duplicate Form

Dialog box prompts user for form to duplicate from a list of existing forms. After selection is made, a new form is created with name "Duplicate of 'Previous Name'". A new form number is also assigned. This allows the user to create a derivative form without reentering all form information from another form.

Remove Form

Dialog box prompts user for the form to remove. The program removes it after the selection is made and confirmed.

Send Form (Available from "Connect" menu)

Dialog box prompts user for form to send. After the form is selected, program goes into a wait state until the PDA **104** is connected. Once the PDA **104** is connected, form is sent as series of field description records. If the PDA **104** is not connected, a keystroke sequence allows the wait state to be terminated.

The following are Field Operations (Available from "Design" menu).

New Field

The user is prompted with a dialog box from which the desired field type is selected. In the presently preferred embodiment, this list includes multiple response, multiple dichotomy, matrix, text, number. In another embodiment, counter types are utilized. After a selection is made, an appropriate editor window allows the field to be created. There are specific editors for use with each type of field. Each editor is designed to gather appropriate information for the respective field type in a user-friendly manner.

Edit Field

The selected field is displayed with the appropriate field editor.

Remove Field

The selected field or fields are removed if the user accepts a confirmation message.

he following is a Program Function (Available from "Connect" menu).

Receive Data

The receive data function puts the program in a wait state. Multiple PDAs may be connected, and all form information is downloaded by program. When the user has finished with all PDAs, a keystroke sequence is entered that will end the wait state. At this point all form data received is sorted by form number and saved in individual files by form.

B. Steps in Creating Example Form

Forms are created as a series of numbered fields which are navigated by field scripts. Field types include common survey response structures, as well as text and numeric fields for open ended entry. The execution of the form is driven by field scripts written in a scripting language supporting flow of control statements such as IF/THEN/ELSE, and over 100 functions. Scripts can provide common survey functions such as data validation, field navigation, context sensitive help, data formatting, alert sounds, and dialog boxes. Scripts can ensure that skip patterns are always followed correctly, and that forms are always filled out accurately. In another embodiment, skip or sequence control statements, such as Next and Goto may be utilized by the forms engine outside of the script, i.e., sequence control statements do not have to be part of the script. Other embodiments may not even use scripts.

Thus, the scripting feature of the forms system **100** further improves survey accuracy and reliability. Scripts also allow the form designer or forms creator to find bottlenecks in questionnaires by determining how much time the user takes to answer a particular field, or to trigger other PDA applications from inside the form.

Referring now to, FIGS. **2***a*, **2***b* and **2***c* the field description records or form descriptors generated by the steps below are illustrated in a composite diagram. The comment card **140** (FIG. **2***a*) is a diagrammatic paper representation shown as a composite of the individual display screens for each field of the form. At this juncture, it must be emphasized that the form as a whole is not displayed (as illustrated in FIG. **2***a*) on the graphics display of the PDA **104** (FIG. **1**). The steps or instructions for creating an example form, a "Customer Comment Card" for Joe's Diner, are as follows. (FIGS. **3**–**7** show details of fields **2**–**6**. Field **1** is not shown in detail because it is not a displayable field, i.e., a screen display is not generated for this field.)

1. Create and Name the Form

Select New . . . from the File menu and enter a name for the form and a brief description. Click OK to bring up an empty form window.

2. Create Field 1

Select New Field . . . from the Design menu to create a new field. When prompted for field type, select Script, then click OK. Because the first field in the form always holds the current date, the form fills this field in automatically using a script type field. In the field editor that is brought into use by the forms creation program, some attributes have already been filled in: number and offset. The field editor is used by the forms designer to edit an individual field description record.

Referring now to a record **142** (FIG. **2***b*), the number in a field attribute or field uniquely identifies the field description record (FDR) (and the field defined by the FDR), and the offset field indicates a position in the resulting data stream, i.e., the form data array, that will be used by the script. In this case, offset **1** is used to store the date. The text in a text edit box is changed by the forms creator to the following:

```
1)    SET data[1] TO date( )
2)    NEXT
```

The first line of this script uses the set command to put the date in offset **1**. The second line uses the NEXT statement to move on to the next field in the form. The forms creator clicks the OK button to accept the changes made to this field. The result is example field description record (FDR) **142**.

3 . Create Field 2—Number of people in your party

Referring now to FDR **144** (FIG. **2***b*), select New Field . . . from the Design menu to create a new field. When prompted for field type, select Numeric. This field requires more attributes than the script field. The question number and offset have been entered automatically. First enter the desired text in the form of a question or statement. In this example, the text, "Number of people in your party", is entered in the prompt box. Next, the forms creator writes a field script that will be evaluated when the user taps the next field button. In this case, the forms engine continues on to field **3** if the user fills in this field, or beeps and displays a message if the field is empty. The forms creator then changes the default script "NEXT" to the following:

```
1)    IF Answered(2) THEN
2)        NEXT
3)    ELSE
4)        BEEP
5)        MESSAGE "Please enter a number!"
6)    ENDIF
```

This script uses an IF/THEN/ELSE statement to do two different things depending on a certain condition, that is, whether or not field **2** has been answered. If it has been answered, line **2** of the script uses the NEXT statement to advance the user to the next field. However, if field **2** has not been answered, line **4** of the script initiates sounding a beep with the BEEP statement and line **5** uses the MESSAGE statement to display a dialog box with the text "Please enter a number!". Note that the script does not wait for the user to provide an answer as part of the script, but rather, after the script is performed, the forms engine waits and checks for the user to either press a button or provide answer box input during the forms engine function **124** (FIG. **8**).

If the forms designer would like to provide the user with help, i.e., a message activated with the help button, the help message is entered in the entry box marked Help Message. The forms designer clicks OK to accept and save the changes made to this field. The result is example field description record **144**. An example screen display **162** shown in FIG. **3** corresponds with the FDR **144**.

4. Create Field 3—Meal/s Ordered

Referring now to FDR **146** (FIG. **2***b*), select New Field . . . from the Design menu to create a new field. When prompted for field type, select Multiple Dichotomy. This field is a little different, in that its results take up more than one position in the form data array. In fact, this type requires one position, or offset, in the form data array for each of the possible selections. The data will contain a 1 (one) at the appropriate offset if the item is selected, and will remain blank otherwise.

First, the forms creator enters the text, "Meal/s Orderd", in the entry box marked prompt. Next, the forms creator clicks Add . . . below the list box labeled Answer Choices, enters "Breakfast" in the dialog, and clicks Add. "Lunch" and "Dinner" are entered in the same way. When all answer choices have been entered, click Done. As the answers are entered, the offsets 3,4, and 5 are inserted automatically by the forms creation program. The forms creator now writes a script that will control the sequence of the electronic form when the user taps the next button. In this example, if the user has selected "Breakfast", the form asks if he came for the $2.99 special, otherwise the form requests the user to rate some aspects of the restaurant. If the user doesn't select any answer, the forms engine will sound a beep and display a message asking the user to make a selection.

The forms creator enters in the following script to exhibit this behavior:

```
1)    IF data[3] = 1 THEN
2)         NEXT
3)    ELSE IF Answered(3) THEN
4)         GOTO 5
5)    ELSE
6)         BEEP
7)         MESSAGE "Please make a selection"
8)    ENDIF
```

The IF/THEN/ELSE statement is used to take different actions depending on certain conditions. The first line of the script checks to see if "Breakfast" has been selected by checking whether offset 3 contains a 1 (one). If it does, line 2 of the script uses the NEXT statement to continue with the next field in order, i.e., field 4. If offset 3 does not contain a 1 (one), line 3 of the script is evaluated. Line 3 checks to see if the other choices, "Lunch" and "Dinner", were chosen in field 3 by using the Answered( ) function. The argument of the Answered( ) function is a field number and returns a true or false result. If the user has selected at least one response or provided an answer for the field identified in the Answered( ) function argument, the function returns as true. If no answer has been provided by the user, the function returns false. If one of the other choices has been selected, line 4 uses the GOTO statement to skip to field 5 (FDR 150). Otherwise lines 6 and 7 of the script initiate sounding a beep and display the message "Please make a selection".

If the forms designer would like to provide the user with help, the help message is entered in the entry box marked Help Message. The forms designer clicks OK to accept and save the changes made to this field. The result is example field description record 146. An example screen display 166 shown in FIG. 4 corresponds with the FDR 146. An associated field status record 168 provides information used in the display of the selected answer(s).

5. Create Field 4—"If breakfast ordered, did you come for the $2.99 breakfast special?"

Referring now to FDR 148 (FIG. 2*c*), select New Field . . . from the Design menu to create a new field. When prompted for field type, select Multiple Response. This field type is similar to Multiple Dichotomy in that the forms creator can list multiple answers for the user to choose from, but different in the respect that the number of answers that a user can select simultaneously can be limited. The results from this field occupy as many offsets in the form data array as there are simultaneous responses allowed. If the forms creator limits the user to only two responses, only two offsets are used to store the results of the user's selections.

To create the field, the forms creator first enters the question text in the prompt entry box. Next, the number of responses allowed is entered, which in this case is 1 (one). (The form user can select "Yes" or "No", but not both.) The offset 4 is automatically inserted in the offset list by the forms creation program. The forms creator must also enter result values for each of the answer choices available. These will be the values placed at the appropriate offsets for the answer choice selected. For example, the forms creator can choose the value 1 (one) for "Yes" and the value 2 (two) for "No". This will cause a 1 (one) to be placed at offset 4 when the user selects "Yes" and a 2 (two) to be placed at offset 4 when the user selects "No". To enter these answers and result values, the forms creator clicks on the Add button below the list box, and in the dialog box, enters the answers and the respective result values. The forms creator clicks Done when finished.

In this example, because the forms designer doesn't care if the user leaves this field unanswered, the script doesn't need to do anything complicated. It only needs to take the user to the next field. Therefore, no changes need to be made to the default script, "NEXT".

If the forms designer would like to provide the user with help, the help message is entered in the entry box marked Help Message. The forms designer clicks OK to accept and save the changes made to this field. The result is example field description record 148. An example screen display 176 shown in FIG. 5 corresponds with the FDR 148. An associated field status record 178 provides information used in display of the selected answer.

6. Create Field 5—"Please rate the following:"

Referring now to FDR 150 (FIG. 2*c*), select New Field . . . from the Design menu to create a new field. When prompted for field type, select Matrix. The matrix type allows the forms designer to create several rows of buttons from which the user can select only one button per row. The matrix field editor is more complex than some of the others. First, the forms designer fills in the Prompt entry box with the prompt: "Please rate the following." Next, the forms designer fills in the item labels and their corresponding offsets, the column labels, and the result values. To add items to any of these lists, the forms designer clicks on the list, then clicks the Add button. The forms designer enters the matrix row labels in the first list box starting with "Restaurant Cleanliness" and ending with "Food Temperature". As these items are entered, the next available offsets, 7 through 11, are automatically inserted by the forms creation program. Next, the forms designer inserts the matrix column labels. In this example, there are only two: "Poor" and "Excellent". Column labels will always appear equally spaced across the top of the matrix. Next, the forms designer enters the result values. The number of result values entered determines the number of bubbles there are for each row. The result values themselves will be inserted in the form data array at the offset for each item. In this example, the forms designer uses a scale of 1 to 5. The numbers 1 through 5 are added by clicking on the Result Value list and then clicking the Add button.

Again, because the forms designer does not require a response for this field, the default script, "NEXT" is used, as in the previous field (field 4).

If the forms designer would like to provide the user with help, the help message is entered in the entry box marked Help Message. The forms designer clicks OK to accept and save the changes made to this field. The result is example field description record 150. An example screen display 184 shown in FIG. 6 corresponds with the FDR 150. An associated field status record 186 provides information used in display of the selected answer(s).

7. Create Field 6—"Which two of the following would you like to see added to the menu?"

Referring now to FDR 152 (FIG. 2*c*), select New Field . . . from the Design menu to create a new field. When prompted for field type, select Multiple Response. This is the same type of field used in the Yes/No field above (field 4), but now, more than one response is allowed. In the Prompt entry box, the forms designer enters the text: "Which two of the following would you like to see added to the menu?". Next, the forms designer enters "2" for the number of responses allowed. The next two available offsets, 12 and 13, are automatically inserted in the offset list by the forms creation program. These offsets are the locations in the form data array where the two selections will be stored. Next, the forms designer clicks on the Add button below the Answer Choices list box to enter the answer choices ("More Seafood entrees" through "Reduced calorie entrees") and their corresponding result values (1 through 5).

The script for this field is very simple. The forms designer replaces the default "NEXT" script with "EXIT" to cause the form to end after this field has been completed.

If the forms designer would like to provide the user with help, the help message is entered in the entry box marked Help Message. The forms designer clicks OK to accept and save the changes made to this field. The result is example field description record **152**. An example screen display **192** shown in FIG. **7** corresponds with the FDR **152**. An associated field status record **194** provides information used in display of the selected answer.

8. Send Form to the PDA

Select Send Form . . . from the Connect menu. If the form window is still open, a status window will appear waiting for the forms designer to connect the PDA **104**. If the forms designer has closed the form window, he/she will be prompted for a form to send, and then the status window appears. The forms designer connects the PDA **104** with a serial cable, taps on the envelope icon, and then taps Connect on the pop-up item. The form is then sent to the PDA **104** and the status window closes. As an alternative, a modem or other communications device can be used in place of the serial cable to transfer the form. The communications device may use wireless technology.

IV. THE FORMS ENGINE

Once an electronic form is loaded in the PDA **104** (FIG. 1), a data collector or interviewer in a survey or questionnaire type of application, can choose to fill out any of a number of forms which have been previously loaded. The number of forms that can be loaded depends on the amount of random-access memory (RAM) (not shown) in the PDA **104**. Current PDA devices can generally store large numbers of reasonably small forms in the standard RAM.

Users complete or fill out a form during execution of the forms engine function **124**. Once form execution begins, the interviewer has very little to do when compared to the paper based process. Because the scripts, running under the forms engine, control form execution, the user doesn't need to follow any complex skip patterns or even worry about answering a question incorrectly. Aside from answering the current question, he/she can select one of four functions by tapping on buttons (FIG. **3**): go to the NEXT field **220**, go to PREVIOUS field **222**, get HELP **224**, or QUIT **226**. When the NEXT button **220** is pressed, the field script is executed. At this point, the PDA **104** running the script may check to see that the field was answered correctly and if so, go to the next appropriate field. If not, the script may cause the PDA **104** to beep and display a help message, or perform some other action. The forms engine **124** tracks the progress as the user progresses through the form. At any point while completing the form, the user may tap the PREVIOUS button **222** to go back to change an answer to a previous question. Also, the user may tap the HELP button **224** at any time for instructions for the current field.

When an electronic form has been completed, the form data stored in the PDA **104** can be sent back to the PC **102** via serial cable, modem, or other communications devices. Once transferred, the form data can be automatically cleared from the PDA memory. In the presently preferred embodiment, the form data is then sorted by form and saved to disk, preferably in a tab-delimited format, where it is available for use with spreadsheets, databases, tabulation packages, and other programs. In another embodiment, other formats for the form data are utilized.

Referring now to FIG. **8**, the forms engine function **124** located in the PDA **104** (FIG. **1**) will now be described. Beginning at a start state **302**, the forms engine **124** moves to a state **304** wherein the user selects the electronic form to be completed and presses a start button displayed on the screen **106** of the PDA **104**. Moving to state **306**, the forms engine **124** creates a stack to hold the numbers of the fields already answered in the order they were answered thus providing a mechanism to maintain a history of the answered fields. A stack is a well known data structure in the programming technology.

Proceeding to state **308**, the forms engine **124** creates a form data array to hold the form data collected as each field is completed. The data array is indexed by the offset number provided in the field description record (FDR). For example, the field description record **148** (FIG. **2c**) for field number **4** uses offset **6** to hold the answer for that field, while FDR **150** for field number **5** uses offsets **7–11** to hold the multiple answers for that field. At some time after the form is completed, the data array is transmitted from the PDA **104** to the computer **102** as a data stream.

Advancing to state **310**, the forms engine **124** creates a form status array to hold a field status record for each field type that requires it. In the presently preferred embodiment, the field types that require the field status record are the multiple dichotomy, multiple response, and matrix. The field status record is created and used by the forms engine **124** to track which answers or responses are selected by the user and is used to draw the field. Referring to the example of FIG. **4**, field status record **168** includes an answerState **170**. The answerState essentially indicates which box, button, or bubble of a plurality of boxes, buttons or bubbles is selected. The answerState is a one-dimensional array having as many elements as the number of offsets defined in the corresponding FDR. Three offsets are indicated by the FDR **146**, so the answerState will be a three element array. Thus, since only the first and second of three boxes are checked in the example of FIG. **4**, the answerState is [1,1,NIL]. The field status record will be further described in conjunction with the draw field function **316** (FIG. **9**) and answer box input function **372** (FIG. **10**).

Moving to state **312**, the forms engine **124** accesses the first field (field number 1) of the selected electronic form in the PDA memory. In the example form presented in FIG. **2**, field 1 referenced by numeral **142** is accessed. Proceeding to a decision state **314**, forms engine **124** determines if the type field of the current field is "script". The example field 1 (**142**) is of type "script". If the field type is script, the forms engine advances to an "execute script" function **354** that executes the script developed by the forms designer for the current field. In general, if the type field is script, the script initiates a calculation or an action to be performed by the PDA **104**. The execute script function **354** will be further described in conjunction with FIG. **11**.

If the field type is not script, as determined at state **314**, the forms engine **124** moves to a "draw field" function **316** that generates a screen display for the current field being processed. In the example field drawn on the screen display **162** shown in FIG. **3**, the user of the form is required to respond to a sentence in a prompt box portion **163** of the screen display **162** by making a selection or writing an answer in an answer box portion **164**. The draw field function **316** will be further described in conjunction with FIG. **9**.

Moving to decision states **320**, **330**, **340**, **350** and **370**, the forms engine **124** determines if one of buttons **220** to **226** (FIG. **3**) in the control portion of the screen display **162** has been selected, or if the user is making a selection or writing an answer in the answer box portion **164**. The control portion of the screen display **162** comprises functions under forms engine control, i.e., buttons **220** through **226** and scroll arrows **228**, **230**.

At decision state **320**, the forms engine **124** determines if the quit button **226** has been selected. If the quit button has been selected, the forms engine proceeds to an end state **322** to terminate the forms engine process. If the quit button was not selected, as determined at state **320**, the forms engine **124** advances to a decision state **330** to determine if the help button **224** has been selected. If so, the forms engine **124** moves to a decision state **332** to determine if the current field has associated "help text" in the help field of the FDR. If so, the forms engine **124** proceeds to state **334** and displays the help text on the screen **106** of the PDA **104**. If the forms designer did not utilize "help text" for the current field, as determined at state **332**, or after the help text has been displayed and dismissed, the forms engine **124** continues on to a decision state **340**.

If the help button **224** (FIG. **3**) was not selected, as determined at state **330**, the forms engine **124** advances to decision state **340** to determine if the "previous" button **222** has been selected. If so, the forms engine **124** moves to state **342** and pops the last field number that was entered on the stack (created at state **306**). The FDR corresponding to the field number popped from the stack is then accessed, and the forms engine **124** loops back to decision state **314** to process the field.

If the previous button **222** (FIG. **3**) was not selected, as determined at state **340**, the forms engine **124** advances to a decision state **350** to determine if the "Next" button **220** has been selected. If so, the forms engine **124** moves to a decision state **351** to determine if the current field is a text or numeric field type. If not, the forms engine **124** skips state **352** because the other field types insert their data into the form data array when the user makes selections in the answer box. If the current field type was determined at state **351** to be either text or numeric, the forms engine **124** moves to state **352** and saves the response(s) or answer(s), i.e., the resultant field data, provided by the user in the answer box portion **164** into the form data array (created at state **308**) at the offset(s) identified for the current field. Each field of the electronic form that requires a response from the user of the form has one or more associated offsets, as defined in the FDR, into the form data array. After the field data is saved at state **352** or if the field type is other than text or numeric, as determined at state **351**, the forms engine **124** proceeds to the "execute script" function **354**. Function **354** performs the step(s) of the script for the current field which may include validating the response(s) of the user and determining the next field to be processed. Function **354** will be further described hereinbelow.

Upon return from the execute script function **354** (FIG. **11**), the forms engine **124** checks the status of several flags or indicators that may be set during execution of the script file during function **354**. The condition of the flags is communicated to the forms engine **124**. Moving to a decision state **356**, the forms engine **124** determines if an "exit" flag was set as a result of performing an exit statement during function **354**. If so, the forms engine proceeds to an end state **358** which signifies that the form has been completed.

If the exit flag is not set, as determined at state **356**, the forms engine **124** proceeds to a decision state **360** to determine if a "stack" flag is set. If so, the forms engine **124** loops back to state **314** to determine the field type of the new field addressed and accessed at either state **732** (due to a NEXT statement) or state **742** (due to a GOTO statement) during execution of function **354**.

If the stack flag is not set, as determined at state **360**, the forms engine **124** proceeds to a decision state **362** to determine if a "restart" flag was set as a result of performing a RESTART statement during function **354**. If so, the forms engine **124** loops back to state **306** to begin processing the currently selected form from the beginning. During RESTART of execute script **354**, the form data array is saved in PDA non-volatile static RAM at a state **752** before returning to the forms engine **124**.

If the restart flag is not set, as determined at state **362**, the forms engine **124** proceeds to a decision state **364** to determine if an "end" flag was set. The end flag is set due to reaching the end of script during function **354**. If so, the forms engine **124** loops back to decision state **320** to check for additional input or action by the user. If the end flag is not set, as determined at state **364**, the forms engine **124** proceeds to state **366** to perform any error handling necessary, for example, due to an error flag set during execution of the forms engine **124** or any called functions, such as array indices out of bounds, division by zero or other error conditions. Upon completion of error handling at state **366**, the forms engine proceeds to the end state **358**.

If the Next button **220** (FIG. **3**) was not selected, as determined at state **350**, the forms engine **124** advances to a decision state **370** to determine if the user has made changes to the answer box **164** portion of the display (FIG. **3**) response(s). If so, the forms engine **124** moves to a "process answer box input" function **372**. Function **372** initiates drawing the selections or input made by the user in the answer box portion **164** of the screen display **162** (FIG. **3**) while completing the current field and performs validation, e.g., makes sure input falls within the constraints of the field. The form data array and, if necessary, the field status record are also updated during the answer box input function **372**. Function **372** will be further described in conjunction with FIG. **10**. Upon return from function **372** or if decision state **370** proves to be false (no answer box input), the forms engine **124** loops back to decision state **320** to wait for user input or selection of one of the buttons **220–226** (FIG. **3**) in the control portion of the screen display **162**. In the current preferred embodiment, the PDA **104** shuts itself off after a user defined period of inactivity. When turned back on, the user continues where he or she was previously.

Table 1 shows an example stack for the example Customer Comment Card shown in FIGS. **2–7**, while Table 2 shows an example form data array. Both the stack and the array are shown when the form is completed by the user. The values shown in the stack and form data array correspond with the user selections indicated in the example FIGS. **3–7**.

TABLE 1

Field Order Stack

6	←←← Top of Stack
5	
4	
3	
2	

TABLE 2

Form Data Array

Offset	1	2	3	4	5	6	7	8	9	10	11	12	13
Value	5/13/94	3	1	1		2	1	3	2		5	1	3

V. DRAWING FIELDS

Referring now to FIGS. *9a* and *9b*, the "draw field" function **316** will be described. The draw field function **316** draws the screen display for the current field being processed. As previously mentioned, the user of the form is required to respond to a "sentence" in a prompt box portion (as shown by the example field illustrated in FIG. 3) of the screen display **162** by making a selection or writing an answer in an answer box portion **164**. The sentence in the prompt box portion can be a statement, a command, a question or other similar construct, and may be an incomplete sentence, e.g., a phrase.

Beginning at a start state **400**, the forms engine **124** moves to state **402** and displays the prompt from the prompt field of the current field description record (FDR) in the prompt box portion **163** at the top of the screen display **162** (FIG. 3). Proceeding to a decision state **404**, the forms engine determines if the field type of the current FDR is "multiple response". If so, the forms engine **124** advances to a decision state **406** to determine the number of responses allowed for the current field by examining the number of offsets defined in the current FDR. For example, field **4** referenced by numeral **148** (FIG. 2*c*) defines one offset (thus allowing one response), while field **6** referenced by numeral **152** defines two offsets (thus permitting two responses). Moving to a decision state **408**, if more than one response is allowed for the current field, the forms engine **124** moves to state **410** and draws check boxes and assorted answers from the answers array of the FDR, beginning at the top of the answer box **164** (FIG. 7) until either all answers have been drawn or no more answers will fit on the display screen **106** of the PDA **104**. Check boxes and the process of drawing them are well known in the field of graphics technology, e.g., Visual Basic available from Microsoft Corp. uses and draws checkboxes.

If the current FDR only defines one response, i.e., one offset, as determined at state **408**, the forms engine proceeds to state **412** and draws radio buttons and associated answer text from the answer array of the FDR beginning at the top of the answer box **164** (FIG. 5) until either all answers have been drawn or no more answers will fit on the display screen of the PDA **104**. Radio buttons are round option buttons used for making a selection, such that only one radio button can be selected within a group of radio buttons at one time. At the completion of state **410** (drawing check boxes) or state **412** (drawing radio buttons), the forms engine **124** moves to a decision state **414** to determine if there are too many possible answers or responses to fit on the screen **106** of the PDA at one time. If so, the forms engine proceeds to state **416** and draws an "A to Z indexing bar" at the bottom of the answer box so the user can navigate the remaining answers onto the display screen of the PDA. The answers are arranged in alphabetic order to facilitate use of the indexing bar. Alternatively, scroll arrows **228**, **230** (FIG. 3) can be used to scroll into view the additional answers (one screenful at a time) into the answer box.

If all the answers fit on one display screen of the PDA, as determined at state **414**, or after the indexing bar is drawn at state **416**, the forms engine proceeds to a decision state **418** to determine if a field status record exists for the current field. If not, the forms engine proceeds to a create field status record function **420** to generate a record for the current field. In the presently preferred embodiment, several field types utilize a field status record, namely multiple response, multiple dichotomy and matrix. Several fields may be utilized within a field status record depending on the field type defined in the current FDR. The matrix field type and the multiple dichotomy field type use an answerState field and the multiple response field type uses a usedOffsets/answerState and two additional fields, described below, if more than one offset is defined in the FDR.

As previously mentioned, the field status record is created and used by the forms engine **124** to record which answers or responses are selected by the user. Referring to the example screen and records of FIG. 7 (for a multiple response field type with multiple offsets), field status record **194** includes an usedOffsets/answerState field **202**. The usedOffsets field **202** indicates which offsets, of the offsets defined in the corresponding FDR, are used. The usedOffsets field **202** utilizes an "n" by one array, where "n" is the number of check boxes in the field, to indicate which offset corresponds to each checked box. The array **202** essentially indicates which box or button of a plurality of response boxes or buttons is selected. The array **202** is a single element if only one response is allowed, but has as many elements as checkboxes if multiple responses are allowed. All the array elements are initially set to NIL. The array **202** indicates the position of selected checkboxes, wherein the first element of the array corresponds to the first checkbox in the list of checkboxes and so forth. When a checkbox is selected by the user, the first offset number from an availOffsets array of the field status record is used to overwrite the contents of the usedOffsets array corresponding to the selected checkbox.

In the example of FIG. 7, offsets **12** and **13** are defined in the FDR and two boxes are checked on the display at the first and third positions (beginning at the top of the answer list, with the first position selected first. The usedOffsets field is indicated by [12,NIL, 13,NIL,NIL] in this example, where NIL indicates that the position is not used.

Because more than one offset is defined in the corresponding FDR for this multiple response field, two additional fields are utilized in the field status record: boxesChecked **198**, availOffsets **200**. The value of the BoxesChecked field **198** indicates how many check boxes are currently checked.

In the example of FIG. 7, two boxes are checked (as seen in answer box portion **164**). The availOffsets field **200** indicates which offsets, of the offsets defined in the corresponding FDR, are yet unused. Initially, the offsets from the offset field of the current FDR are copied into the availOffsets array. In the example of FIG. 7, since two offsets are defined in the FDR and two boxes are already checked, there are no available offsets which is indicated by [NIL, NIL].

At the completion of creating the field status record for the current field at function **420** or if the record was determined to exist at decision state **418**, the forms engine **124** proceeds to a loop of states **422**, **424**, **426** to fill in any check marks for the displayed field. At state **422**, for each check box visible in the answer box portion **164** of the display screen, the forms engine **124** determines at a decision state **424** if a check mark is to be placed in the check box at state **426**. At decision state **424**, the forms engine uses the answerState array, if multiple dichotomy is the current field type, or usedOffsets array, if multiple response is the current field type, of the field status record to determine if the position indexed by the check box number currently evaluated contains a value greater than zero. If so, a check mark is drawn in the check box at state **426**. If not, the forms engine loops back to state **422** to evaluate the next visible check box. The loop **422**–**426** continues until all visible check boxes have been evaluated, at which time the forms engine **124** moves to state **428** and returns to the forms engine function (FIG. **8**).

If the forms engine **124** determines at decision state **404** that the field type of the current FDR is not "multiple response", the forms engine advances to a decision state **430** to determine if the field type is "multiple dichotomy". If so, the forms engine **124** proceeds to state **410**, wherein check boxes are drawn on the answer box portion **164** of the display screen of the PDA as seen in the example of FIG. **4**. State **410** and the remaining states of the flow for multiple dichotomy are the same as for multiple response discussed above. A field status record **168** utilizes an answerState field **170** that contains an array with as many elements as offsets defined in the corresponding FDR **146**. All the elements of the array are initially set to NIL. When a checkbox is selected by the user, the value of the position in the array corresponding to the selected checkbox toggles from NIL to one (1), if the box was initially blank, or from 1 to NIL, if the checkbox was already checked. Thus, in the example of FIG. **4**, the first and second of three checkboxes is shown to be checked, so the answerState array is [1, 1, NIL].

If the forms engine **124** determines at decision state **430** that the field type of the current FDR is not "multiple dichotomy", the forms engine advances to a decision state **440** to determine if the field type is "text". If so, the forms engine **124** proceeds to state **442** and draws three lines at the top of answer box portion **164** of the display screen **106** for the user to write on with the stylus. Moving to state **444**, the forms engine **124** draws a QWERTY-style keyboard image in the bottom part of the answer box. Moving to state **446**, the forms engine **124** checks the form data array to see if an answer has already been stored for this field (storing operation done at state **352**, FIG. **8**) at a decision state **448**. If the field has already been answered, the forms engine **124** moves to state **450** and draws the value or characters stored in the form data array on the lines (text) or line (numeric) previously drawn (states **442** or **462**) on the screen **106**. If the field is not already filled in, as determined at state **448**, or after completion of state **450**, the forms engine **124** moves to state **428** and returns to the forms engine function (FIG. **8**).

If the forms engine **124** determines at decision state **440** that the field type of the current FDR is not "text", the forms engine **124** advances to a decision state **460** to determine if the field type is "numeric" If so, the forms engine **124** proceeds to state **462** and draws one line at the top of answer box portion **164** of the display screen **106** for the user to write on with the stylus. Moving to state **464**, the forms engine **124** draws a numeric keypad image in the bottom part of the answer box. Moving to state **446**, the forms engine **124** checks the form data array to see if an answer has already been stored for this field. The remaining states for the numeric field type, states **448**, **450**, and **428**, have been described above. An example of a numeric type field is shown in the example of FIG. **3**.

If the forms engine **124** determines at decision state **460** that the field type of the current FDR is not "numeric", the forms engine **124** proceeds through off-page connector B **468** to a decision state **470** on FIG. **9b** to determine if the field type is "matrix". If so, the forms engine **124** proceeds to state **472** and generates a row of bubbles on the screen **106** with associated text from the answer field of the current FDR, with as many bubbles as there are values in the result array of the FDR. This operation is repeated for as many rows as there are offsets defined in the offset array of the current FDR. For example, the FDR **150** for the example matrix field shown in FIG. **6** has five values in the "result" field and has five offsets defined in the "offset" field. Thus, a 5×5 matrix is drawn in the answer box portion **164** of the example screen display **184**. Moving to state **473**, the forms engine **124** draws column headings in **473** based on the range array in the current FDR.

Proceeding to a decision state **474**, the forms engine **124** determines if a field status record **186** exists for the current field. If not, the forms engine advances to a "create field status record" function **476** to generate the record. The record **186** utilizes an answerState array **188** that is a one by "n" array with as many elements "n" as there are offsets defined in the FDR for the current field. Each element of the array is initially set to NIL. In the example of FIG. **6**, five offsets, corresponding to the five rows in the matrix, are defined in the FDR **150**. The first (top) row corresponds with the first element of the array, and so forth. If a bubble is selected in a particular row, the column number of the selected bubble (left-most column number is one) is used as the value of the element in the answerState array. A matrix row without a selected bubble has the corresponding element in the answerState array as NIL.

If the field status record was determined to exist at state **474**, or at the completion of "create field status record" function **476**, the forms engine **124** proceeds to a loop of states **478**, **480**, **482** to mark any previously selected bubbles for the displayed field. At state **478**, for each row of bubbles visible in the answer box portion **164** of the display screen, the forms engine **124** determines at a decision state **480** if a mark is to be placed in the designated bubble at state **482**. At decision state **480**, the forms engine uses the answerState array of the field status record to determine if the element of the array at index [bubblerow] is not NIL. If true, at state **482**, the forms engine **124** marks the bubble in the current row of the matrix at the bubble column identified by the value stored at index [bubblerow] in the answerState array. If decision state **480** evaluates to false (the element at index [bubblerow] is NIL), the forms engine **124** accesses the next row of bubbles at state **478**. The loop **478**–**482** continues until all visible bubble rows have been evaluated, at which time the forms engine **124** moves to state **484** and returns to the forms engine function (FIG. **8**).

If the forms engine **124** determines at decision state **470** that the field type of the current FDR is not "matrix", the forms engine advances to a decision state **490** to determine if the field type is "counter". If so, the forms engine **124** proceeds to state **492** and draws a button at the top of answer box portion **164** of the display screen **106**. This button contains a number used as a counter, such that every time the button is selected, the number is incremented by one. In another embodiment of the invention, the counter can be initialized to begin at a selected number, such as zero or one, and count up, or be initialized to a different number, such as 100, and count down to yet another selected number. In another embodiment of the invention, the forms engine **124** draws a timer on the display, in addition to the counter. The timer can be set to begin at zero and time a predetermined time interval, or begin at a selected amount of time, such as an hour, and notify the user after the hour has expired by an alarm or sound, such as a "beep" sound. Various other counter and timer combinations are contemplated.

Moving to state **494**, the forms engine **124** checks the form data array at the offset corresponding to the current "counter" field to see if the field already has a value stored at a decision state **496**. If the field does not already have a value stored, the forms engine **124** moves to a state **498** and draws the stored value on the button drawn at state **492**. If either the field does have a value stored in the form data array, as determined at state **496**, if the field type is not "counter", as determined at state **490**, or at the completion of drawing the value at state **498**, the forms engine **124** moves to state **484** and returns to the forms engine function (FIG. **8**).

VI. ANSWER BOX INPUT

Referring now to FIGS. **10***a*, **10***b* and **10***c*, the "answer box input" function **372** will be described. After a selected field has been drawn on the screen **106** of the PDA **104**, the user of the electronic form responds to the prompt located in the prompt portion **163** of the display. Answer box input function **372** responds to the user's selections and performs the operations necessary to display the selections and update the form data array for the current field and the field status record (if necessary).

Beginning at a start state **520**, the forms engine proceeds to a decision state **522** to determine if the current field displayed has a field type of "multiple response" If so, the forms engine **124** moves to a decision state **523** to determine if more than one response is allowed, i.e., is more than one offset defined in the current FDR. If only one offset is defined for this field, the forms engine **124** proceeds through off-page connector D **525** to state **527** on FIG. **10***c*. At state **527**, the forms engine **124** sets answerState array, which in the case of a single response is a single element, to the button number of the button selected. The button numbers begin with one, beginning at the top of the display. Moving to state **529**, the forms engine **124** sets the form data array indexed by the offset defined in the FDR to the result value in the FDR result array corresponding to the selected button. In the example shown in FIG. **5**, the second button is selected, so the value (i.e., two) of second element in the result array is written to the form data array at offset **6**. Upon completion of state **529**, the forms engine **124** moves to state **531** and returns to the forms engine function (FIG. **8**).

Returning attention now to decision state **523** on FIG. **10***a*, if more than one offset is defined by the current FDR, the forms engine **124** advances to a decision state **524** to determine if the checkbox selected by the user is already checked. If so, the forms engine **124** moves to state **526** and removes or deletes the check from the checkbox on the screen display. The following three states, **530**, **532**, **534**, revise the field status array record and form data array for this field. At state **530**, the forms engine **124** removes the offset associated with the current checkbox from the usedOffsets array and adds the offset to the availOffsets array in the field status record. Proceeding to state **532**, the forms engine **124** sets the form data array at the offset freed up at state **530** to NIL. Then at state **534**, the forms engine **124** decrements the boxesChecked field in the field status record by one. Upon completion of state **534**, the forms engine **124** moves to state **536** and returns to the forms engine function (FIG. **8**).

Returning attention now to decision state **524**, if the checkbox selected by the user is not already checked, the forms engine **124** moves to a decision state **540**. At decision state **540**, the forms engine **124** determines if the value of the boxesChecked field of the field status record plus one is less than or equal to the number of offsets defined in the FDR for the current field. If so, the forms engine **124** proceeds to state **542** and draws a check on the screen display in the checkbox selected by the user. The following three states, **546**, **548**, **550**, revise the field status array record and form data array for this field. At state **546**, the forms engine **124** removes the first non-nil element from the availOffsets array and adds the offset to the usedOffsets array at the index [checkbox number] in the field status record. Proceeding to state **548**, the forms engine **124** sets the form data array at the new offset, added to usedOffsets at state **546**, to the value of the result array of the FDR at index checkbox number. Moving to state **550**, the forms engine **124** increments the boxesChecked field in the field status record by one. Upon completion of state **550**, the forms engine **124** moves to state **536** and returns to the forms engine function (FIG. **8**).

Returning attention now to decision state **522**, if the field type is not "multiple response", the forms engine **124** advances to a decision state **560** and determines if the field type is "multiple dichotomy". If so, the forms engine **124** proceeds to state **562** and writes the answerState field at index [checkbox number] in the field status record for the current field to the result of the Boolean exclusive-OR of the value presently at answerState[checkbox number] with one. In essence, this operation toggles the value of the checkbox selected by the user in the answerState array. Moving to a decision state **564**, the forms engine **124** determines if the checkbox selected by the user is already marked as checked. If so, the forms engine **124** moves to state **566** and removes or deletes the check from the selected checkbox, i.e., if the user selects a checkbox that is already marked, the mark is cleared from the checkbox. Proceeding to state **568**, the forms engine **124** sets the form data array at index [offset array in the FDR of the current field which is indexed by [checkbox number]] to NIL. For example, referring to FIG. **4**, the offset array field in the FDR **146** for the current field is [3,4,5]. If checkbox number two (the second from the top) is selected by the user, the second element in the offset array is indexed, yielding the value four (4). Thus, offset four in the form data array would be set to NIL (since the checkbox was previously checked, as shown in FIG. **4**). Upon completion of state **568**, the forms engine **124** moves to state **536** and returns to the forms engine function (FIG. **8**).

If the checkbox selected by the user is not already checked, as determined at decision state **564**, the forms engine **124** proceeds to state **570** and draws a check in the selected checkbox. Moving to state **572**, the forms engine **124** sets the form data array at index [offset array in the FDR which is indexed by [checkbox number]] to one. Upon completion of state **572**, the forms engine **124** moves to state **536** and returns to the forms engine function (FIG. **8**).

Returning attention now to decision state **560**, if the field type is not "multiple dichotomy", the forms engine **124** proceeds through off-page connector C **578** to a decision state **590** on FIG. **10*b*** and determines if the filed type is "text". If so, the forms engine **124** proceeds to state **592** and determines if the user has written on the lines displayed on the screen **106** of the PDA **104** or has tapped on the keyboard image displayed on the screen. The user can use either the keyboard image to pick the desired characters to be "typed" on the three lines or the stylus to write the desired characters on the lines. If the characters are written, as determined at state **592**, the PDA **104** utilizes an internal built-in recognition function to interpret the handwriting and display the handwriting on the lines.

If the user has tapped on the keyboard image, as determined at state **592**, the forms engine **124** either inserts the letter pressed at the insertion point to the text string or performs the action, e.g., move forward, move backward, delete. For example, a particular field may be looking for a single letter, such as "c" as an answer or may alternately build up a word or sequence of words (a single letter at a time if using the keyboard or a word at a time if writing) in the text string. If a displayable character was tapped by the user, the character is displayed on the lines of the PDA screen **106**. Upon completion of state **596**, the forms engine **124** moves to state **598** and returns to the forms engine function (FIG. **8**).

Returning attention now to decision state **590**, if the field type is not "text", the forms engine **124** proceeds to a decision state **610** and determines if the field type is "numeric". If so, the forms engine **124** proceeds to state **612** and executes states **612** and **614** or **612** and **616** in a manner similar to states **592**–**596** for the field type "text". The differences are that only one line is shown on the PDA screen **106** for a numeric field type, and a number is built up one digit at a time (or the answer could be a single digit) instead of building a text string.

Returning attention now to decision state **610**, if the field type is not "numeric", the forms engine **124** proceeds to a decision state **630** and determines if the field type is "matrix". If so, the forms engine **124** proceeds to state **632** and determines if the button selected by the user is already marked or selected. If so, no further action is performed on the row in the matrix that has the selected button. The forms engine **124** moves to state **598** and returns to the forms engine function (FIG. **8**).

If the button pressed by the user is not already marked as selected, as determined at decision state **632**, the forms engine **124** proceeds to state **634** and removes or deletes the mark from the previously selected bubble in the same row as the mark currently selected by the user. Moving to state **636**, the forms engine **124** draws a mark in the newly selected bubble in the row. Proceeding to state **638**, the forms engine **124** changes the answerState field of the field status record for the current field at index [Row of Button] to the column number (column one is left-most on the display) of the newly selected bubble. Then, at state **640**, the forms engine **124** changes the form data array indexed at the offset array field (in the FDR for the current field) that is indexed by the Row of Button to the value of the result stored in result field of the current FDR indexed by the Column of Button. The Row of Button is the row in the matrix of the button newly selected by the user.

For example, referring to FIG. **6**, the offset array field in the FDR **150** is [7,8,9,10,11]. If the user just selected the last (fifth) bubble in the fifth row, the Row of Bubble is 5. Therefore, the fifth position in the offset array field is indexed by Row of Bubble to yield offset **11** (eleven). This offset eleven then is used to index the form data array at index eleven, which is changed to the value stored in the result array at index Column of Button, which in our example is 5. At the completion of state **640**, the forms engine **124** moves to state **598** and returns to the forms engine function (FIG. **8**).

Returning attention now to decision state **630**, if the field type is not "matrix", the forms engine **124** proceeds to a decision state **650** and determines if the field type is "counter". If so, the forms engine **124** proceeds to state **652** and increments the element located in the form data array indexed at the offset in the FDR for this field by one and then draws this number in the button displayed on the PDA screen **106**. In other words, the counter value is incremented by one, and the value is stored in the form data array at the offset for this field and also displayed on the PDA screen. At the completion of state **652**, or if the field type is not "counter", as determined at state **650**, the forms engine **124** moves to state **654** and returns to the forms engine function (FIG. **8**).

VII. EXECUTE SCRIPT

The "execute script" function **354** will now be described. The execute script function **354** is called by the forms engine **124** (FIG. **8**) either as a result of determining that the field type of the current field is "script" (state **314**), or after the Next button is pushed (state **350**) and the field data for the current field is saved in the form data array (state **352**). In either case, the script field of the FDR for the current field is executed by the function **354**. The script field contains a script or small program written in a scripting language supporting flow of control statements, such as IF/THEN/ELSE, and over 100 functions. The script may be as short as a single statement, or contain an entire procedure. A script can provide functions for the electronic form such as data validation, field navigation, context sensitive help, data formatting, alert sounds, and dialog boxes. Data validation may be performed with respect to any information available to the forms engine, which includes information derived from the functions defined in the scripting language, the forms data array, and information available from the PDA operating system. An example of data that can be incorporated into a validation statement is the current date. The current date could be compared to a date provided by the user, and the user input could be accepted or rejected based on the comparison.

The execute script function utilizes the grammar rules provided in Table 3:

TABLE 3

Script Language Grammar Rules - Backus Naur Form (BNF)

```
The rules presented here are defined with the following conventions:
nonterminal
TERMINAL
[ optional ]
{ one | two | three } any of these choices
[ zero or more] *
[one or more ]+
< description >      This is a description of what is allowed
GRAMMAR RULES
```

TABLE 3-continued

Script Language Grammar Rules - Backus Naur Form (BNF)

```
input
        statement-list
statement-list
        statement-list statement
statement
{ function | if-expression | for-expression | while-expression |
variable-declaration }
function
        name ( [ parameter-list ] )
parameter-list
        { parameter | parameter , parameter-list }
parameter
        { string | expression }
if-expression
        IF boolean-expression THEN statement-list [ELSE statement-list]
        ENDIF
for-expression
        FOR name GETS expression TO expression ENDFOR
while-expression
        WHILE boolean-expression DO statement-list ENDWHILE
variable-declaration
        { VAR name | VAR name GETS expression }
boolean-expression
        {   boolean-function
          | expression    boolean-operator    expression
          | boolean-expression    AND    boolean-expression
          | boolean-expression    OR    boolean-expression
          | NOT boolean-expression
          | expression }
boolean-operator
        { = | <> | > | < | >= | <= }
expression
        {   expression operator expression
          | function
          | number
          | realnumber
          | OFFSET number
          | OFFSET ( expression ) }
operator
        { + | – | * | DIV | MOD | & }
name
        [character ]+
number
        [ – ] [ digit ]+
realnumber
        [ – ] [ digit ]+ · [ digit ]* [{ e | E } [ – ] [digit ]+ ]
digit
        { 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 }
string
        " character-sequence "
character-sequence
        [{string-character | escape-sequence}]*
string-character
        < any ascii character code 32–127 except " or \ >
escape-sequence
        { \ { " | \ | n | t }
character
        < A to Z and a–z >
```

Referring now to FIG. 11, portions of the execute script function **354** relevant to the forms engine **124** are presented in a flowchart format. The execute script function **354** begins at a start state **700**, accesses the first statement in the script, and proceeds to a decision state **702** to determine if the statement is part of a control structure, e.g., IF/THEN/ELSE. If so, the forms engine **124** performs the action necessary to implement the control structure at state **704**, and advances to state **706** to access the next statement in the script file based on the action of any control structure that may be active, e.g., loop, IF/THEN. The forms engine **124** then loops back to state **702** to evaluate the new statement in the script.

If the statement is not part of a control structure, as determined at decision state **702**, the forms engine **124** proceeds to decision states **720–810** to check the statements in the script by type. At decision state **720**, the forms engine **124** determines if the statement is an EXIT statement. If so, the forms engine **124** advances to state **722** and permanently saves the form data array in the PDA **104**. After the data is saved at state **722**, the forms engine **124** moves to state **724** wherein an "exit" flag is set and a return is performed to the forms engine function (FIG. **8**). The exit flag denotes that the EXIT statement was performed by the execute script function **354**.

If the current statement is not an EXIT statement as determined at decision state **720**, the forms engine **124** proceeds to a decision state **730** to determine if the statement is a NEXT statement. If so, the forms engine **124** advances to state **732**, pushes the current field number on the stack (created at state **306**, FIG. **8**), and accesses the FDR of the next field in numerical order, i.e., moves to the next field of the electronic form. After the stack push and next field access at state **732**, the forms engine **124** moves to state **734** wherein a "stack" flag is set and a return is performed to the forms engine function (FIG. **8**). The stack flag denotes that a navigation-type statement was performed by the execute script function **354**.

If the current statement is not a NEXT statement as determined at decision state **730**, the forms engine **124** proceeds to a decision state **740** to determine if the statement is a GOTO (numeric expression) statement. If so, the forms engine **124** advances to state **742**, pushes the current field number on the stack, and accesses the FDR of the field number provided or determined by the statement, i.e., skips or jumps to the field of the electronic form pointed to by the numeric expression in the GOTO statement. After the stack push and next skip field access at state **742**, the forms engine **124** moves to state **734** wherein the "stack" flag is set and a return is performed to the forms engine function (FIG. **8**).

If the current statement is not a GOTO (numerical expression) statement as determined at decision state **740**, the forms engine **124** proceeds to a decision state **750** to determine if the statement is a RESTART statement. If so, the forms engine **124** advances to state **752** and permanently saves the form data array in the PDA **104** (as performed at state **722**). After the data is saved at state **752**, the forms engine **124** moves to state **754** wherein a "restart" flag is set and a return is performed to the forms engine function (FIG. **8**). The restart flag denotes that the RESTART statement was performed by the execute script function **354**.

If the current statement is not a RESTART statement as determined at decision state **750**, the forms engine **124** proceeds to a decision state **760** to determine if the statement is an "other returning" statement. "Other returning" statements designates other statements not listed in the flowchart of FIG. **11** that are processed by the execute script function and optionally perform some action, including communication based actions, and return to the forms engine function. If state **760** is true, the forms engine **124** advances to state **762** and optionally performs the action corresponding to the "other returning" statement, if an action is required. After the optional action at state **762**, the forms engine **124** moves to state **764** wherein a optional flag, shown as "other", is set and a return is performed to the forms engine function (FIG. **8**). Thus, a generic returning function has been described.

If the current statement is not an "other returning" statement as determined at decision state **760**, the forms engine **124** proceeds to a decision state **770** to determine if the statement is a BEEP statement. If so, the forms engine **124** advances to state **772** and initiates a beep sound that is produced by the PDA **104**. Of course, other sounds can be readily produced in other embodiments. After the beep sound is produced at state **772**, the forms engine **124** proceeds to state **706** wherein the next statement in the script file is accessed, as previously described.

If the current statement is not a BEEP statement as determined at decision state **770**, the forms engine **124** proceeds to a decision state **780** to determine if the statement is a MESSAGE (STRING EXPN) statement. If so, the forms engine **124** advances to state **782** and displays a dialog box with the "STRING" message on the PDA screen **106**. The forms engine **124** then proceeds to state **706** wherein the next statement in the script is accessed.

If the current statement is not a MESSAGE (STRING EXPN) statement as determined at decision state **780**, the forms engine **124** proceeds to a decision state **790** to determine if the statement is an "other non-returning" statement. "Other non-returning" statement designates other functions not listed in the flowchart of FIG. **11** that are processed by the execute script function and perform an action but do not return to the forms engine function. If state **790** is true, the forms engine **124** advances to state **792** and performs the action corresponding to the "other non-returning" function. The forms engine **124** then proceeds to state **706** wherein the next statement in the script file is accessed.

If the current statement is not an "other non-returning" statement, as determined at decision state **790**, the forms engine **124** proceeds to a decision state **800** to determine if the statement is a SET statement, in the format SET [VARIABLE]TO [EXPN], e.g., SET data[3] TO 5 or SET data[10]TO 12+min(data[1],data[2]). If so, the forms engine **124** proceeds to state **802** and assigns the right-hand side of the SET statement, i.e., the expression (EXPN) after the word "TO", to the variable after the word "SET." Moving to state **706**, the forms engine **124** accesses the next statement in the script file, as described above.

If the current statement is not a SET statement as determined at decision state **800**, the forms engine **124** proceeds to a decision state **810** to determine if the end of the script is reached. If so, the forms engine **124** moves to state **812** wherein an "end" (of script) flag is set and a return is performed to the forms engine function (FIG. **8**). The end flag denotes that the end of the script was reached by the execute script function **354**. Upon return to the forms engine function, the forms engine **124** evaluates the flags and then waits for user input. If the end of the script has not been reached, as determined at decision state **810**, the forms engine **124** proceeds to state **706** wherein the next statement in the script is accessed.

VIII. BENEFITS OF THE FORMS SYSTEM

Complex forms, such as questionnaires, can be created quickly and easily. Part of the forms system **100** includes the forms creation program that runs on a Macintosh or Windows based PC. This application allows the user to create a form for use with the Newton PDA **104** (FIG. **1**). To create a form, the form designer simply points and clicks to define a number of fields of different types. For each field, the designer specifies certain attributes such as the field type, a question or prompt, a list of answer choices, a help message, a control script, and others. All attributes are defined with visual editors providing maximal ease of use. This information is then easily transferred to the Newton via serial cable. After data has been collected by filling out the form with the Newton, the data is transferred from the Newton back to the host PC where it is available for use with common data analysis programs.

No paper forms are needed. At no time in the data collection process is paper used. From the time the form is designed to the time data is collected, both the form and data exist only in digital format. This eliminates the high cost of form duplication and also eliminates labor and equipment costs associated with scanning forms using OMR or OCR. The elimination of paper forms also rids the data collection process of expensive manual data entry that was previously needed to transfer data from paper forms to a computer.

Designers can create intelligent forms that cannot be completed incorrectly. Field scripts allow form designers to implement data validation, skip patterns, data formatting, alert sounds, dialog boxes, and communications abilities. Scripts are responsible for actually walking the user through the entire form and providing interactive feedback. This technique brings to data collection a level of ease and accuracy previously unattainable in the industry.

As improvements are made in PDA technology, the flexibility of the forms system **100** will allow it to take advantage of new features. For example, as handwriting recognition improves, open ended questions can be answered more easily by allowing users to write text directly on the screen rather than using on-screen keyboards. Further improvements in wireless technology will also allow the forms system **100** to improve the speed at which data can be collected and analyzed.

IX. OPTIONAL SYSTEM CONFIGURATION

In yet other embodiments, other structures, arrays, portable computers, operating systems or algorithms can be used. The general system, method and procedures would remain the same. For example, field types will be added allowing forms to collect information using external peripheral devices such as bar code readers, digital cameras, and microphones. With these additions, forms could be created that are capable of collecting visual and audible data, something never possible with paper forms.

The forms system **100** described herein finds application in many environments, and is readily adaptable for use therein. The system finds use in any application in which data is collected procedurally or algorithmically. For example, the system could be used to automate a personality profile based on a line of questioning. After answering a series of questions, the user would be provided with results calculated from the answers he gave. The system would also find application in a product order entry environment. Product order forms could be filled out using the PDA and then sent directly to a vendor where they would be automatically processed because they arrived in digital format. The system would also find use automating quality control checklists common on manufacturing floors.

X. SUMMARY OF THE ADVANTAGES OF THE PRESENT INVENTION

Eliminates paper forms from a previously paper intensive process;

Eliminates need for expensive scanners used to process paper forms;

Eliminates labor expense by eliminating manual data entry, form coding, and form scanning;

Significantly reduces form creation and modification cost by eliminating layout necessary for paper forms;

Allows form designers to create intelligent forms that can eliminate the opportunity for errors common in previous methods;

Data collectors are 'walked' through forms and can get help at any time, thus reducing training time and improving accuracy;

Data can be collected, compiled, and analyzed in an extremely short span of time and with a level of ease and accuracy not previously attainable.

While the above detailed description has shown, described, and pointed out the fundamental novel features of the invention as applied to various embodiments, it will be understood that various omissions and substitutions and changes in the form and details of the device illustrated may be made by those skilled in the art, without departing from the spirit of the invention.

What is claimed is:

1. A system for completing an electronic form, comprising:

a portable unit including a processor, a graphics display, an input device mechanism and a memory;

an electronic form stored in the memory of the unit comprising a plurality of form descriptors that define displayable items, wherein each form descriptor includes a self-contained script program indicative of the next displayable item to be displayed, wherein each script program stores and retrieves data to and from the memory and wherein the order of display of the displayable items is defined by the script programs;

a forms engine to display a single displayable item on the graphics display described by one of the form descriptors, wherein the displayable item includes a sentence and a set of possible response entries, wherein the script program of the displayable item is interpreted by the forms engine, and wherein the one displayable item utilizes the entire display; and

wherein the input device mechanism accepts one or more of the responses selected by a user of the unit.

2. The system defined in claim 1, wherein the input mechanism comprises:

a stylus;

a screen responsive to contact with the stylus and providing signals indicative to each contact; and

software in communication with the screen for responding to each contact of the screen.

3. The system defined in claim 1, wherein the forms engine is capable of receiving signals from an external peripheral device.

4. The system defined in claim 3, wherein the peripheral device includes one of the following: barcode reader, digital camera, and microphone.

5. The system defined in claim 1, wherein the form descriptor includes a type field.

6. The system defined in claim 5, wherein one of the field types is script.

7. The system defined in claim 6, wherein the initial initiates an action.

8. The system defined in claim 7, wherein the action includes a calculation.

9. The system defined in claim 7, wherein the action initiates a communication function.

10. The system defined in claim 9, wherein the communication function includes a communications device.

11. The system defined in claim 10, wherein the communications device comprises a modem.

12. The system defined in claim 10, wherein the communication device comprises a point-to-point communication link.

13. The system defined in claim 10, wherein the communication device comprises a wireless device.

14. The system defined in claim 1, wherein the displayable item requires an answer box input by the user.

15. The system defined in claim 14, wherein the answer box input is indicative of one of a plurality of types.

16. The system defined in claim 15, wherein one answer box input type includes a counter function.

17. The system defined in claim 16, wherein the counter function displays a button and a number on the graphics display, and wherein the number is changed when the button is selected by the user.

18. The system defined in claim 17, wherein the number is initialized to a predetermined value.

19. The system defined in claim 17, wherein the number is incremented when the button is selected by the user.

20. The system defined in claim 17, wherein the number is decremented when the button is selected by the user.

21. The system defined in claim 1, wherein the displayable item includes a next button.

22. The system defined in claim 21, wherein selection of the next button initiates evaluation of an expression to validate the response by the user.

23. The system defined in claim 22, wherein a successful evaluation of the expression to validate the response initiates an evaluation of an expression to determine the next form descriptor.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 5,704,029
DATED : December 30, 1997
INVENTOR(S) : Gerald V. Wright

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 30,
Line 5, please change “initial” to -- script --.

Signed and Sealed this

Fifteenth Day of October, 2002

Attest:

JAMES E. ROGAN
Director of the United States Patent and Trademark Office

Attesting Officer

PAT-A

US007822816B2

(12) **United States Patent**
Payne

(10) **Patent No.: US 7,822,816 B2**
(45) **Date of Patent: Oct. 26, 2010**

(54) **SYSTEM AND METHOD FOR DATA MANAGEMENT**

(75) Inventor: **J. David Payne**, Broken Arrow, OK (US)

(73) Assignee: **Macrosolve, Inc.**, Tulsa, OK (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 613 days.

(21) Appl. No.: **10/643,516**

(22) Filed: **Aug. 19, 2003**

(65) **Prior Publication Data**

US 2004/0034684 A1 Feb. 19, 2004

Related U.S. Application Data

(60) Provisional application No. 60/404,491, filed on Aug. 19, 2002.

(51) **Int. Cl.**
G06F 15/173 (2006.01)

(52) **U.S. Cl.** **709/206**; 709/203; 709/224

(58) **Field of Classification Search** 709/203, 709/223, 224
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,666,553 A		9/1997	Crozier	
5,684,990 A		11/1997	Boothby	
5,842,195 A	*	11/1998	Peters et al.	707/1
5,943,676 A		8/1999	Boothby	
6,141,664 A		10/2000	Boothby	
6,151,581 A	*	11/2000	Kraftson et al.	705/3
6,163,811 A	*	12/2000	Porter	709/247
6,212,529 B1		4/2001	Boothby et al.	
6,405,218 B1		6/2002	Boothby	

(Continued)

FOREIGN PATENT DOCUMENTS

WO	WO 01/84433 A	11/2001

OTHER PUBLICATIONS

Petsas, et al., "WAP-Based personalised health care services"; Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2001 Conference Proceedings. (EMBS) Instanbul, Turkey, Oct. 25-28, 2001, Annual International Conference of the IEEE Engineering in M. vo.l. 1 of 4, Conf. 23, Oct. 25, 2001 pp. 3536-3539.

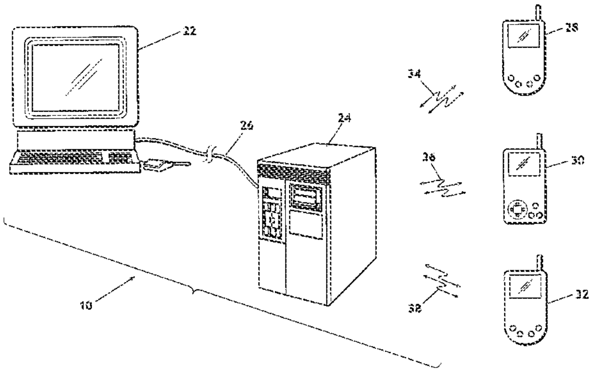
(Continued)

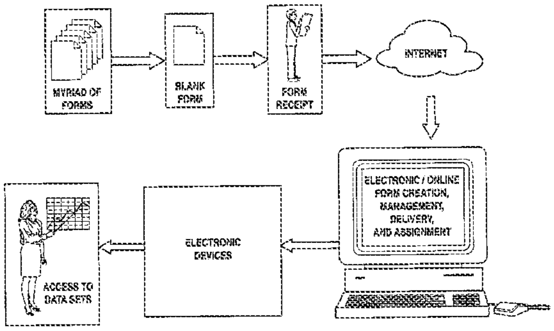
Primary Examiner—Khanh Q Dinh
Assistant Examiner—Nghi V Tran
(74) *Attorney, Agent, or Firm*—Fellers, Snider, Blankenship, Bailey & Tippens, P.C.

(57) **ABSTRACT**

A method for the management of data collected from a remote computing device including the steps of: creating a questionnaire; transmitting the questionnaire to a remote computer; executing the questionnaire in the remote computer to prompt a user for responses to questions of the questionnaire; transmitting the responses to a sever via a network; making the responses available on the Web. Preferably, computers used in connection with the inventive method are loosely networked in that network connections between computers are not always available and, when a connection is not available, data is stored at a node of the network and transmitted at the earliest time when a connection is available. In one preferred embodiment, the inventive method is used to collect survey data and to make the responses to the survey available to a client in virtually real time over the Internet.

14 Claims, 6 Drawing Sheets





U.S. PATENT DOCUMENTS

6,421,717	B1	7/2002	Kloba et al.	709/219
6,826,540	B1 *	11/2004	Plantec et al.	705/10
7,310,350	B1 *	12/2007	Shao et al.	370/466
2001/0056374	A1 *	12/2001	Joao	705/14
2002/0007303	A1 *	1/2002	Brookler et al.	705/10
2002/0107931	A1 *	8/2002	Singh et al.	709/206
2002/0143610	A1 *	10/2002	Munyer	705/12
2002/0160773	A1 *	10/2002	Gresham et al.	455/431
2003/0093405	A1 *	5/2003	Mayer	707/1
2003/0126010	A1 *	7/2003	Barns-Slavin	705/10
2003/0198934	A1 *	10/2003	Sendowski et al.	434/350
2004/0117244	A1 *	6/2004	Scott	705/12
2004/0210472	A1 *	10/2004	Lew et al.	705/10
2005/0009465	A1 *	1/2005	Ross et al.	455/2.01

OTHER PUBLICATIONS

Giirardot, M., et al., "Efficient Representation and Streaming of XML Content Over the Internet Medium", Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on New Yrok, NY, USA, Jul. 30, 2000, pp. 67070.

Anonymous; "WAP Binary XML Content Format", Internet Document [Online], Jun. 24, 1999, pp. 1-14.

* cited by examiner

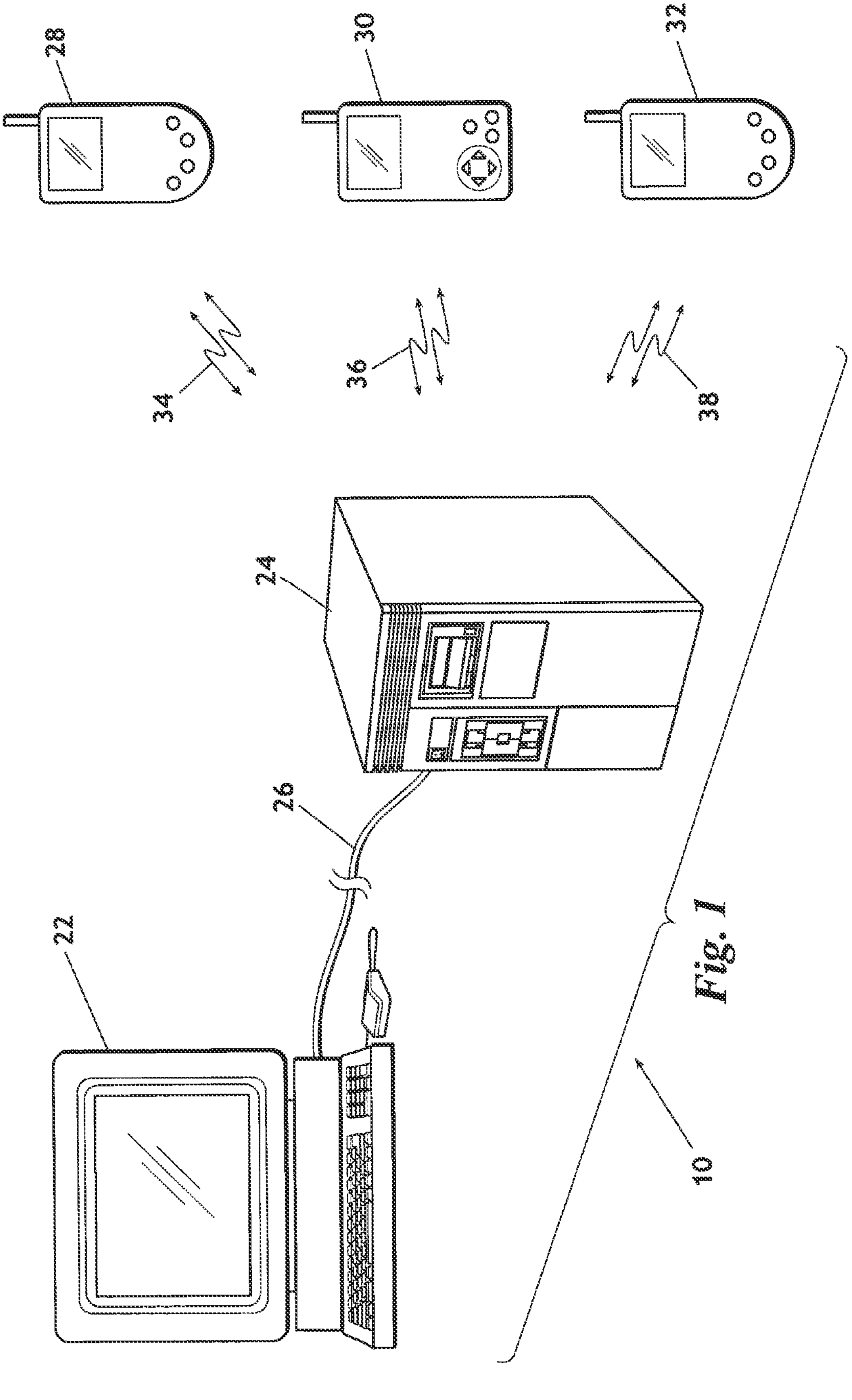


Fig. 1

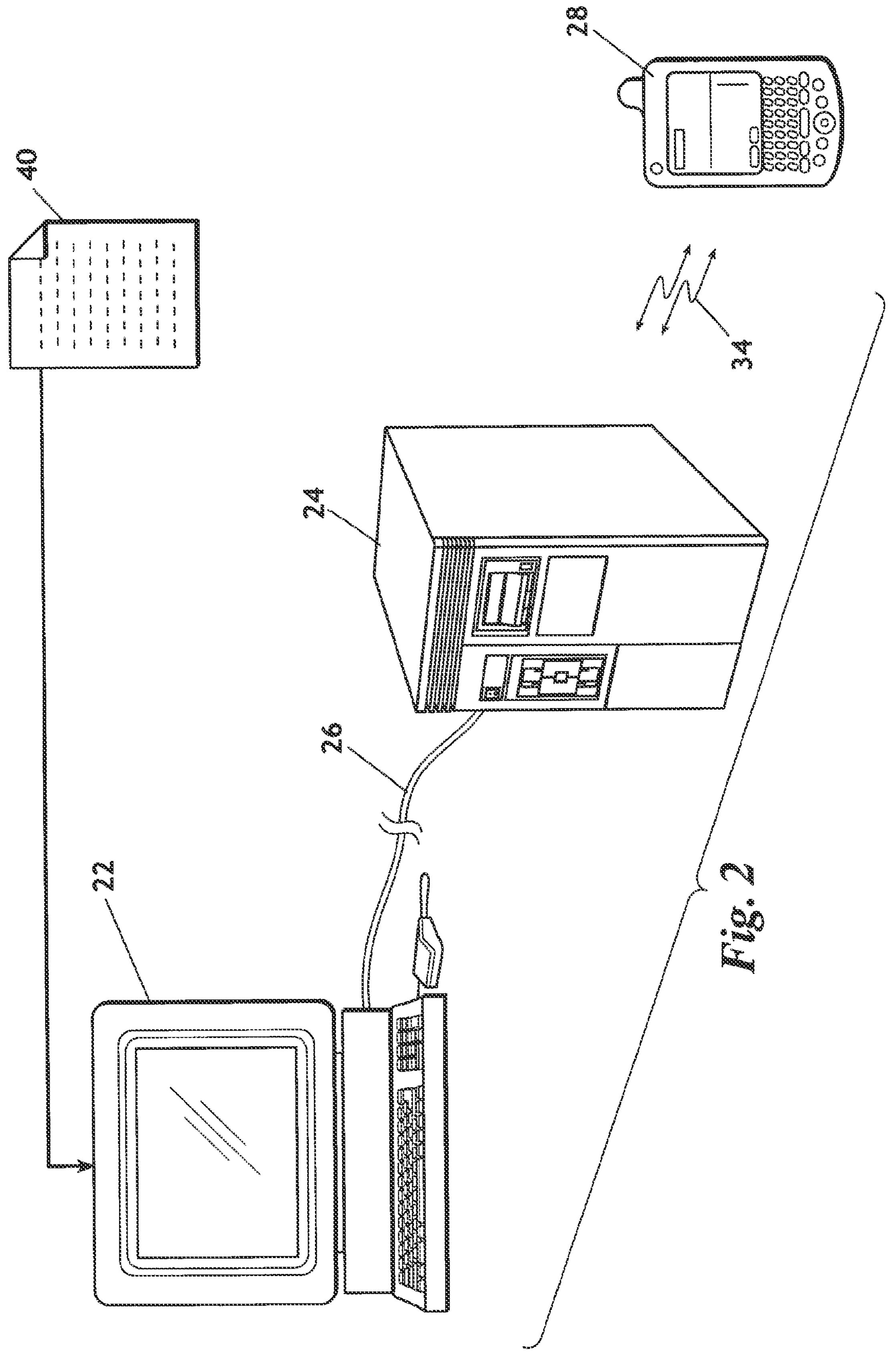


Fig. 2

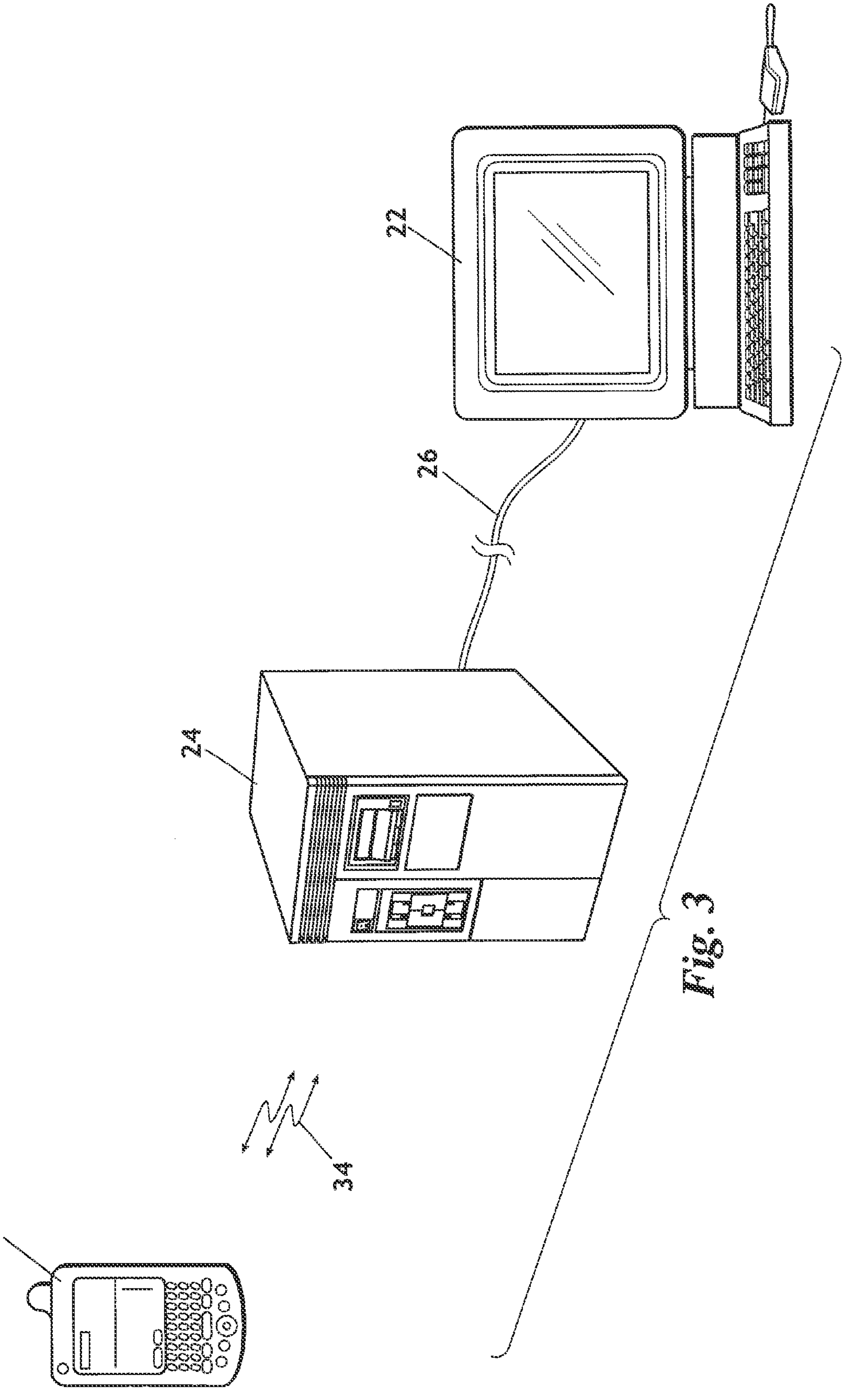


Fig. 3

Fig. 4
(PRIOR ART)

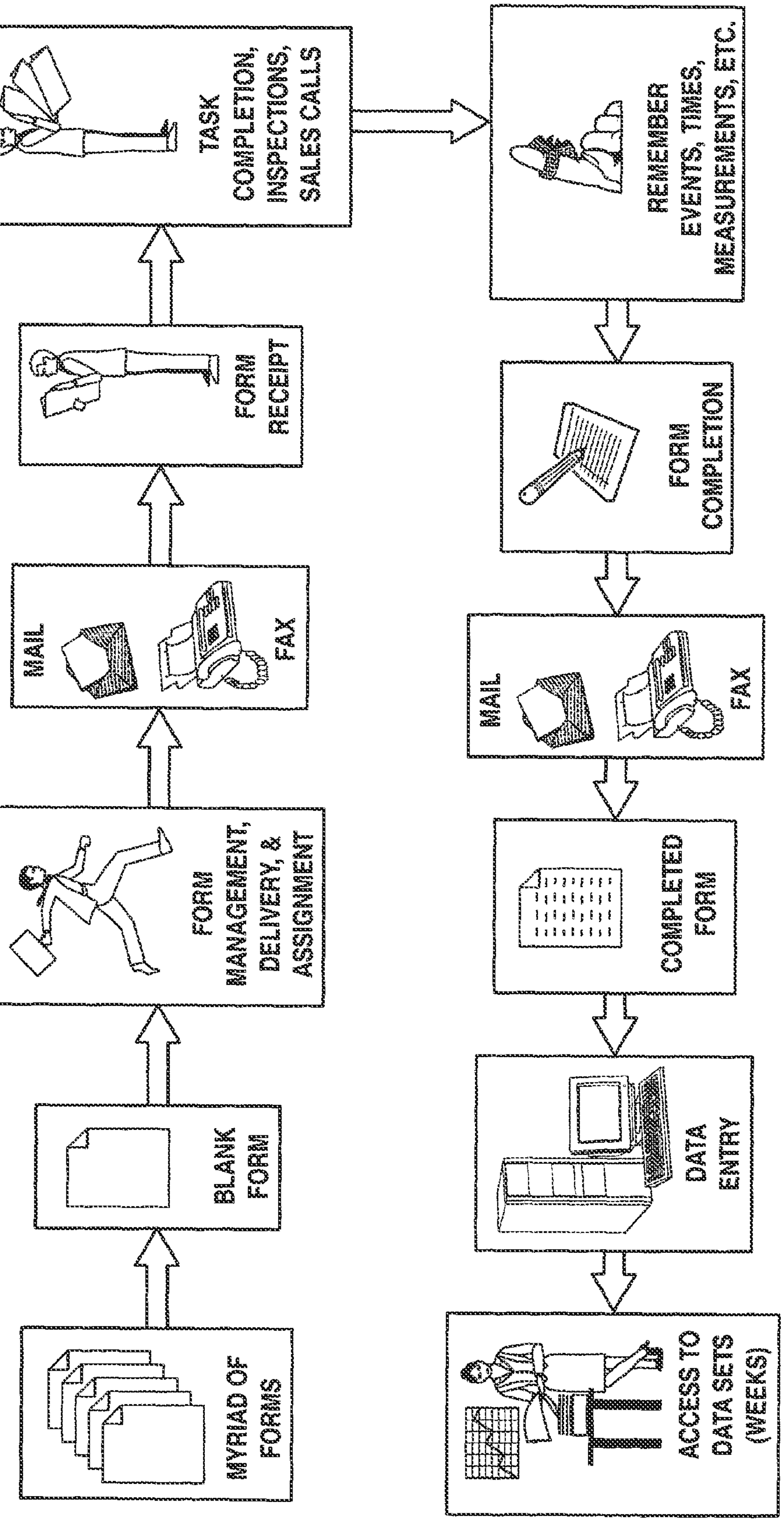
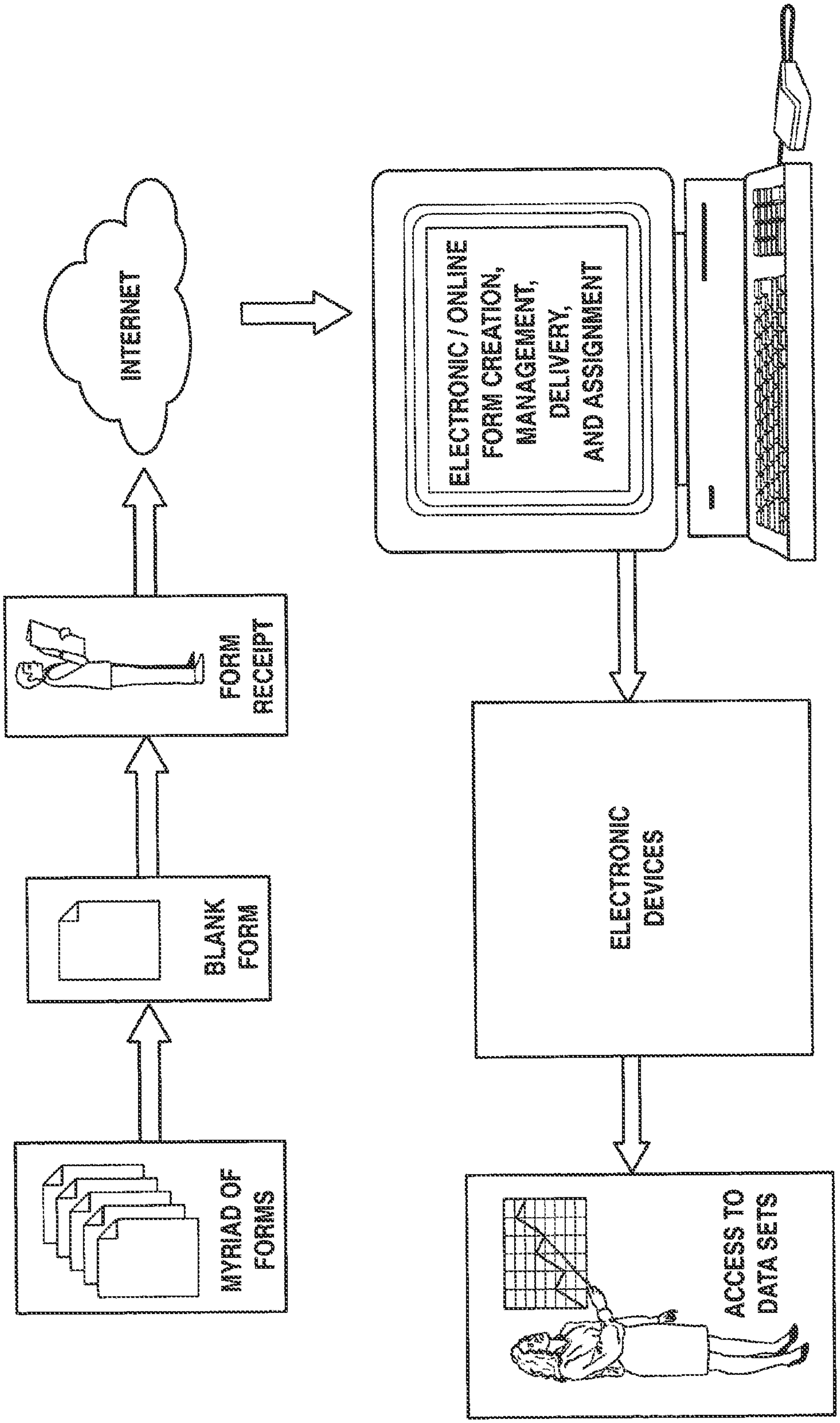

MYRIAD OF FORMS
BLANK FORM
FORM MANAGEMENT, DELIVERY, & ASSIGNMENT
MAIL
FAX
FORM RECEIPT
TASK COMPLETION, INSPECTIONS, SALES CALLS
REMEMBER EVENTS, TIMES, MEASUREMENTS, ETC.
FORM COMPLETION
MAIL
FAX
COMPLETED FORM
DATA ENTRY
ACCESS TO DATA SETS (WEEKS)

Fig. 5


MYRIAD OF FORMS
BLANK FORM
FORM RECEIPT
INTERNET
ELECTRONIC / ONLINE FORM CREATION, MANAGEMENT, DELIVERY, AND ASSIGNMENT
ELECTRONIC DEVICES
ACCESS TO DATA SETS

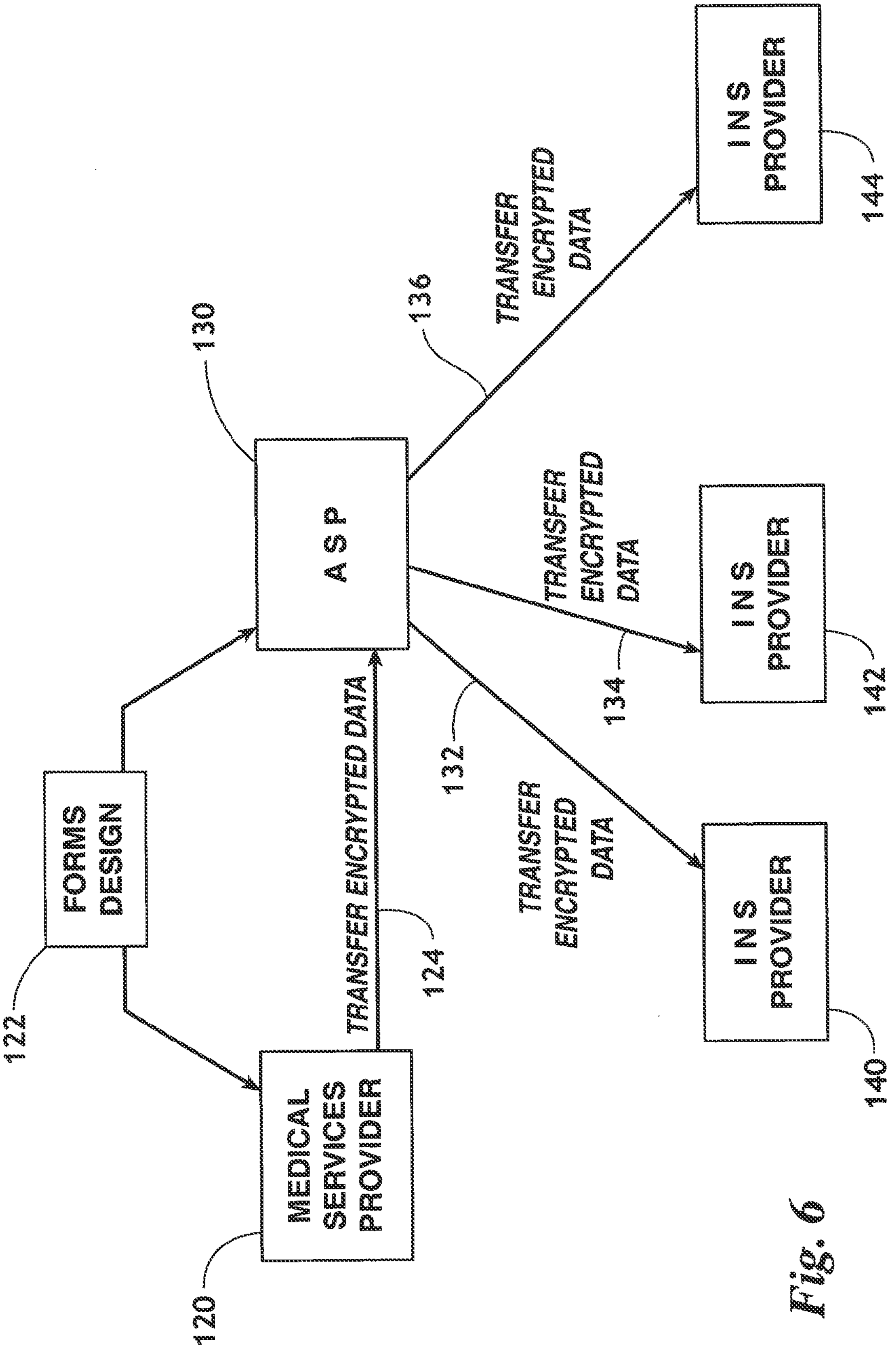

FORMS DESIGN
122
MEDICAL SERVICES PROVIDER
120
TRANSFER ENCRYPTED DATA
124
A S P
130
TRANSFER ENCRYPTED DATA
132
TRANSFER ENCRYPTED DATA
134
TRANSFER ENCRYPTED DATA
136
I N S PROVIDER
140
I N S PROVIDER
142
I N S PROVIDER
144

Fig. 6

SYSTEM AND METHOD FOR DATA MANAGEMENT

CROSS REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application No. 60/404,491 filed Aug. 19, 2002.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a system of computing devices for the collection and management of information. More particularly, but not by way of limitation, the present invention relates to a system for collecting and managing information including a plurality of computer devices loosely networked to a server and an operating system for a computer which provides a number of features favorable for use in the inventive system.

2. Background of the Invention

Virtually all business software applications involve the collection of information in some form or another. Where information is gathered away from the convenience of a desktop, workers have traditionally entered the information on paper forms. This data is then entered into a computer in a second step. This extra step leads to delays and inaccuracies which are costly and, more importantly, unnecessary.

Handheld computers are well known in the art. In fact such computers are presently available from numerous manufacturers offering a vast assortment of operating systems and hardware configurations. While such devices come in a host of variations, generally handheld computers include an LCD display, a method for gathering manual input, storage, and a variety of machine interfaces, i.e., an IR link, a USB port, a serial port, etc.

As with their desktop, and laptop counterparts, a handheld computer will also include an operating system which provides an operator interface, file management, and standardized I/O, as well as facilitating the running of application programs. Thus far, handheld operating systems mimic those of desktop and laptop systems, despite the fact that handheld devices are typically used in a different manner and have radically different resources.

As with other types of computers, handheld computers suffer from compatibility issues, especially in the operation of application programs. Generally speaking, software programs must typically be tailored to a specific family of processors and to a specific operating system. Most applications are developed in a high level language and then compiled for a specific target processor. As different manufacturers select different processors, an application written for one family of processors must be recompiled to execute in a processor of a different family. Even when two manufacturers select compatible processors, if they chose different operating systems, applications written for one device will probably not run correctly on the other device. Since the operating system provides access to the various hardware resources and manages the file system, it is almost unfathomable that the operating systems of independent authors would be compatible, unless one specifically set out to copy the other. Thus, particular applications tend to grow up around a particular family of devices which share an operating system and, unfortunately, the application may not be available for non-compatible devices.

Another issue which is common to all computers is the transferability of stored information, specifically, the ability to move files from machine-to-machine. While most handheld computers include an infrared port for communicating with other infrared devices, including other handhelds, files transferred in such a manner may not be usable by software on the receiving device. This is especially true of information formatted for a particular application such as a word processor, spread sheet program, data base manager, or the like.

To overcome the necessity of compiling a program for a particular machine, an application may be written in an interpreted language, or a language which can be compiled to produce an intermediate language (i.e., a language that falls somewhere between source code and object code) such as i-code or tokens. In such a scheme, each device is provided with a run-time package which can execute the compiled i-code or tokens, the runtime package having been written for that particular device, thus, only the run-time package needs to be modified in order to port a program to a new computing environment. Once the run-time package is installed, any application authored in the language and which has been compiled to i-code will run on the target device. Unfortunately, such languages typically lack effective optimization and generally do not provide a broad range of support for hardware resources. Regardless of the language selected, whether compiled, interpreted, or whatever, software coding requires at least a nominal degree of programming skill to create the application program.

Perhaps because handheld computers are not as evolved as their desktop counterparts, or because it is typically cumbersome to enter information, or maybe due to the lack of a true front-runner in operating systems, handheld computers have not inspired the full range of software products available for larger computers. The result has been a rather limited selection of retail software applications for handheld devices as compared with their desktop counterparts, such software is aimed primarily at organizational tools, e-mail, and games. However, at the other end of the spectrum, custom programs tailored for a specific customer, handheld computers are gaining momentum in replacing manual forms which are often filled-out in remote areas, away from a desktop, i.e. manufacturing inventory, quality inspections, delivery systems, and the like. One reason for the increasing movement toward the use of handheld computers for data gathering tasks is that they can be easily transported to the source of the data and have the information directly entered into them, thereby eliminating the potentially error-prone step of manual data entry of information on previously completed paper forms. Eliminating the extra step additionally saves unnecessary labor, and allows the data to be entered in a more timely fashion.

Due to their incredible portability, handhelds are particularly well suited to this type of data gathering, despite an obvious lack of software infrastructure in this area. The present trend is for a business to commission the authoring of a custom program aimed at a particular need. While the cost of such an application is usually high, the accuracy of the information, the timeliness of the information, and the accessibility of the information are likely worth the cost. In fact, while such systems may seem cost prohibitive to develop, in many cases the actual cost on a per-data-entry basis may prove to be relatively small, especially in light of the timeliness and accuracy associated with real time data collection.

To develop software for a handheld computer, a custom program is typically developed and tested on a larger system. When the developer is satisfied with the program, it is compiled for a particular target device and transferred to handheld devices through a communication link. If users are using more than one type of device, the same program must be tested and compiled for each type of device. If a change is required, the developer must make the change on the development system and re-transfer the entire program to each target device.

In a typical data gathering application, information is entered into custom designed forms on the handheld computer. Eventually, the data entered in the handheld finds its way to a database, which is typically located on a server which is accessible to those needing the information or from which it may be accessed by other programs such as accounting systems, materials management programs, etc. Present day servers are well suited to the task of information management and generally provide broad access to and searchability to collected data.

One problem area in such systems becomes apparent when the data is transferred from the handheld to the server. While it would seem that wireless interfaces and handhelds were made for each other, the marriage of the two is not without its own set of problems. Wireless interfaces fall into a number of different categories. At one extreme is the infrared ("IR") port often found on handheld devices. The range of this type of interface is usually limited to a few feet and typically supports transfer rates of 115 kbaud, or less.

Another method for wireless communication is via a wireless local area network or "WLAN." A typical example of a WLAN is that defined by the IEEE 802.11 standard. When a handheld computer is equipped with a WLAN interface, the device can communicate with other computers also equipped with a WLAN interface, or even computers networked to a WLAN equipped computer by a wired network. Typically, WLAN interfaces provide a range of several hundred feet. As long as a handheld is within the range of another WLAN equipped computer, the network connection is continuous. Wireless local area networks sport data rates from a few thousand bits per second up to at least 52 million bits per second, depending on the particular standard employed.

Yet another known wireless interface for handheld computers is a CDPD interface, CDMA interface, GSM interface, or similar wireless interface or modem. While there are some variations, these systems are often built around a cellular phone network and provide coverage similar to that of a cellular phone, typically national, or even international, coverage. Such interfaces will experience the same gaps in service as can be expected with a cell phone. While such systems provide an exceptionally wide area of coverage, they typically do so at limited bandwidth, e.g. 19.2 kbaud.

Of course handheld devices are not limited to wireless communications. Typically such devices can be connected to another computer through a universal serial bus ("USB") connection, an RS-232 connection, an Ethernet connection on a properly equipped device, or similar hardwired connection. While these interfaces range from moderately paced to the extremely fast, they are exceptionally reliable, at least while the connection is in place. Unfortunately, few environments are well suited to tethering a handheld to allow a continuous wired connection.

It can be seen that perhaps the greatest drawback to using a handheld for data gathering as part of a larger system are the limitations of the data link: 1) it is unlikely that the data link will always be available; and 2) the bandwidth of most of the practical wireless options is restrictive. Presently there are two methods for dealing with the problem of data link availability. In one scheme, data is transmitted as it is collected. The advantage of such a scheme is that the database is updated in real time and represents current data. The disadvantages are, for all practical purposes, the scheme is limited to systems using a wireless interface and when the wireless link is not operational, generally data cannot be entered.

Alternatively, entered data can be stored locally on the handheld and transmitted in a batch process when a link is established. The advantage of this system is that it is tolerant of gaps in the communication link and works well with wired transfers of data. Unfortunately, data is not delivered in real time and the data base may be somewhat stale, depending on the length of time between the collection of data and the presence of the link.

The issue of bandwidth may be problematic on several fronts. If programs are updated periodically, the entire program must be sent and the time to reload may be objectionable. In the opposite direction, if large amounts of data are collected, it may be time consuming to send the data collected from the handheld to the server, particularly when performed in a batch fashion.

It is thus an object of the present invention to provide an operating system for a handheld computer which will allow a program to execute on any handheld computer.

It is a further object of the present invention to provide an operating system for a handheld computer wherein programming changes will only necessitate incremental transfers of program instructions.

It is still a further object of the present invention to provide an operating system for a handheld computer wherein files may be transferred among devices without a translation or conversion.

It is yet a further object of the present invention to provide an operating system for a handheld computer wherein programming steps and data are tokenized to reduce the load on a communication channel of finite bandwidth.

It is yet a further object of the present invention to provide a system of networked computers in which modifications to a computer program for a remote computer are sent in real time to the remote computer and are implemented immediately and seamlessly without the requirement of user installation.

SUMMARY OF THE INVENTION

The present invention provides a system and method for the management of information which solves the problems and alleviates the needs discussed above. In its broadest sense, the present invention is a method designed to accomplish the following:

1) Allow any computer(s) (desktops, laptops, handhelds, portables, etc.) to be used to capture information;
2) Transfer the information to a data center (via file transfer methods such as a network, to include, but not necessarily, Internet based) in a form that the data center can recognize;
3) Allow another computer(s) to access the information and download it from the data center in a format that can be readily used regardless of the format in which the original information was gathered.

In a preferred embodiment, a server is loosely networked to a plurality of computers (handheld, laptop, or desktop). Each computer is equipped with an operating system which allows common programming to execute on any device, regardless of hardware differences or native operating system differences among the plurality of devices.

With regard to the present invention, the term "loosely networked" is used to describe a networked computer system wherein devices on the network are tolerant of intermittent network connections and, in fact, tolerant of the type of network connection available. In particular, if any communication connection is available between devices wishing to communicate, network transmissions occur normally, in real time.

If a network connection is unavailable at that moment, the information is temporarily stored in the device and later transmitted when the connection is restored. Unless otherwise specified, hereinafter the terms "network" or "networked" refer to loosely networked devices.

Thus, the operating system may be thought of as device indifferent and communication channel indifferent. In the preferred embodiment, any computer can execute any program developed for the inventive system and will communicate with other members of the system through any communication method the device can find available.

The operating system provided in each computer device allows the use of a common instruction set in any such device, regardless of compatibility issues between the devices, wherein "instruction set" is used herein to mean the commands, tokens, etc., that are recognized by the operating system as valid instructions. Unlike conventional computer programs, the operating system employed in the inventive system allows incremental changes to the program without the need to reload the entire program. Additionally, a programming change made at a central office will automatically propagate to loosely networked computers dispersed throughout the field.

In one aspect of the invention, branching logic depending on the programs are created for handheld or other computer devices by simply entering questions and providing response specification, in the form of a questionnaire, for the end-user. Thus, no particular programming skill is required to generate programs for data gathering. As will be appreciated by those familiar with data collection, data can be collected by posing a series of questions, or otherwise prompting for specific input from the user, as in the manner used with paper forms. A number of useful subsystems, which may already be present in the handheld device, or easily added later, may be utilized so that at least some of the information which is responsive to the designed questionnaire may be collected automatically rather than entered manually, e.g., time and date, position information if the device includes a GPS receiver, etc.

In another aspect of the present invention, the program and user responses are coded in such a fashion as to substantially reduce the bandwidth requirements of the network connection. Since many of the networking options for handheld devices provide limited bandwidth, best use may be made of the available throughput by coding, or tokenizing, program information and responses.

In practice, a program is created by entering a series of prompts and providing direction for how the system is to respond to particular responses. This process of data gathering may then be performed by a person having no programming skill whatsoever. The program may then be sent to all, or selected, computer devices on the network. Those devices having a connection may immediately be updated. Those devices in which a network connection is temporarily not available will be updated when the connection is next restored.

The user of the computer device is then prompted for specific input. As the user enters data, if the network connection is available, the information is immediately sent to the server. If the network connection is unavailable, the information is stored locally in the handheld device and sent upon restoration of the network connection. At the server, the information is typically processed upon receipt such that users of the data have real time, or virtually real time, information available.

In another aspect of the invention, the inventive system may be provided to end users according to an application service provider ("ASP") business method. ASPs are an emerging trend in the computer software industry. Traditionally, a company seeking a software solution would either acquire a preprogrammed package which suits its needs or commission the programming of custom software. In many instances preprogrammed software is unavailable or requires too many compromises to be attractive. In either case, the software is purchased and, invariably, represents a large capital expense to the company. Once purchased, modifications, evolutional upgrades, changing management practices, and the like, result in additional expenses to keep the software up-to-date.

In contrast an ASP typically provides software on a pay-as-you-go basis. An ASP typically provides custom, or semi-custom software to companies. Each user is billed for the time it uses the software. The advantages to the end user are obvious. There is no crippling up-front expense, modifications and upgrades are the responsibility of the ASP, if the software does not perform satisfactorily the customer simply walks away and never incurs large expenses. On the ASP side, the software vendor enjoys recurring income and the ability to adapt the same software model to numerous customers. While differing slightly from the traditional ASP model, the present invention is particularly well suited to a per-transaction billing model.

With regard to the current system, an ASP can provide a web site which allows users to build an application on line, possibly without incurring any expense. Once the customer is satisfied with the program, it can automatically be deployed to designated computer devices within the system. When a user provides input, the computer device can find a direct network connection to the ASP or, more likely, find an Internet connection and report the data to the ASP via the Internet connection. Once received at the ASP, the data can be processed and is available for viewing or use by the client virtually instantly via the Internet. Thus, data entered at any location may be viewed by the client in real time, worldwide.

Further objects, features, and advantages of the present invention will be apparent to those skilled in the art upon examining the accompanying drawings and upon reading the following description of the preferred embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 provides a diagram of the inventive system.

FIG. 2 provides a diagram of the inventive system as used for form creation.

FIG. 3 provides a diagram of the inventive system as used for information collection and review.

FIG. 4 depicts a sequence of tasks for collecting data through the use of prior art systems.

FIG. 5 depicts the tasks of FIG. 4 utilizing the inventive system.

FIG. 6 is a work flow diagram of an embodiment utilizing the inventive system.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Before explaining the present invention in detail, it is important to understand that the invention is not limited in its application to the details of the construction illustrated and the steps described herein. The invention is capable of other embodiments and of being practiced or carried out in a variety of ways. It is to be understood that the phraseology and terminology employed herein is for the purpose of description and not of limitation.

Referring now to the drawings, wherein like reference numerals indicate the same parts throughout the several views, a diagram of the inventive system is shown in FIG. 1. Typically, the system for data management 10 includes: at least one server 24 preferably having an Internet connection 26; a plurality of handheld computers 28-32 operated remotely from server 24, each handheld 28-32 including a network connection 34-38, respectively, for loosely networking handhelds 28-32 to server 24; and a computer 22 connected to the Internet for providing administration of the system and for reviewing data collected by the system.

Server 24 is shown preferably connected to the Internet 26 and loosely networked to handheld computers 28-32 through connections 34-38, respectively. As will be apparent to those skilled in the art, network connection 26 could instead be local area network or a private wide area network. Similarly, connections 34-38 may be any one of a number of optional connections which ultimately connect a remote device to server 24. By way of example and not limitation, connection 34 could be a simple dial up connection through a conventional telephone line to connect handheld 28 directly to server 24. At the same time, connection 36 could be an infrared (IR) connection between handheld 30 and a desktop computer (not shown) which in turn, is connected to server 24 via the Internet. Connection 38 could be a wireless modem, i.e., a CDPD interface, a CDMA interface, a GSM interface, an analog cellular modem, or the like, which either establishes a direct connection with server 24 or establishes an Internet connection to reach server 24 via the Internet. Other options would include a wireless LAN connection, a direct RS-232 connection, a docking station connected to a desktop computer, etc. It should be noted that, regardless of the type of connection, handhelds 28-32 are ultimately connectable to server 24 in a loosely networked fashion.

It should be noted that handheld computers 28-32 need not be the same type, or even compatible devices. As a part of the inventive system each remote device, preferably a handheld computer, is provided with an operating instruction system ("OIS") which overlays its native operating system. Once equipped with the OIS, a remote device can be programmed according to methods described hereinafter. Any program developed under the inventive system will run on any handheld computer equipped with the OIS and files on one such handheld will transfer freely to any other handheld or any computer connected to the inventive system.

As noted above, with regard to the present invention, the term "loosely networked" is used to describe a networked computer system wherein devices on the network are tolerant of intermittent network connections. In particular, if any communication connection is available between devices wishing to communicate, network transmissions occur normally, in real time. If a network connection is unavailable, the information is temporarily stored in the device and later transmitted when the connection is restored. Unless otherwise specified, hereinafter the terms "network" or "networked" refer to loosely networked devices.

It should also be noted that the inventive system is indifferent as to the particular type of communication channel used for connections 34-36. Thus, by way of example and not limitation, while connection 36 might today be an IR link to a desktop computer which accesses server 24 via the Internet, tomorrow, handheld 30 might establish a connection 36 with server 24 via a CDPD interface. The particular link selected will be the first available link.

The inventive system may be thought of as taking on two distinct modes of operation. First, as shown in FIG. 2, the system provides an administrative function. From any computer 22 connected to the Internet 26, a client can access server 24 to administer the inventive system. Administration involves tasks such as form creation, management, and validation; user setup, and management of system security.

In terms of the present invention, handheld computers are favored for their portability and their usefulness in gathering data from the field, whether the field is a stockroom for a manufacturing facility, a production floor, a delivery site for a product, etc. More generally, field locations are typically areas where people work without the convenience of a desktop.

According to the preferred arrangement, data may be gathered by prompting the user via the handheld 28 with a series of questions or statements, each of which calls for a response. This series of questions or statements will have been constructed on computer 22 and reduced to tokenized form for transmission to the handheld 28. For purposes of the instant disclosure, the series of questions/statements will collectively be referred to as a questionnaire. As will be discussed in greater detail below, the questionnaire is actually designed to include internal branching logic which is implemented by the OIS. Hence, with regard to the present invention, the terms "program" and "form" are used interchangeably with questionnaire.

An important aspect of the invention is the ease with which a client can create a form and distribute the form to the appropriate handheld devices in the field. Continuing with FIG. 2, typically a client uses a computer 22 having access to the Internet 26 to communicate with server 24. As part of the administrative function provided by system 10, computer 22 provides a web-based interface which allows a client to create a questionnaire. As a first step, preferably, the client selects a type of question from a list of standard question types. This list would include alternatives for the way the question is posed to the user, for example visual or vocal, and the type of answer to expect, whether yes/no, multiple choice, narrative, numerical, etc.

As the client creates a list of questions, symbols from a tool bar may be used to control conditional branching based on the user's response. As the client enters questions and selects response types, server 24 builds a stack of questions and responses, and assigns indices, or tokens, which point to each question or response. Each token preferably corresponds to a logical, mathematical, or branching operation and is preferably selected and made a part of the questionnaire through a graphical user interface. By this mechanism, a user is able to create a series of questions, the precise nature of which is dependent on the user's responses. For example, the questionnaire designer might desire to create a form that asks the user different questions; depending on whether the user was male or female. In order to do this, the designer would enter the questions ("Are you a man or woman?"); select a response (a "pop up" list of two entries male and female); select a token (branch if "male"); assign that token to this question; and, specify an "end" location for the "branch" (i.e., the first question asked of "males").

When the questionnaire 40 is complete, server 24 sends the stack of questions and defined responses to the appropriate handheld devices, as represented by handheld 28, via the loosely networked connection 34. In addition, server 24 sends the operating logic for that questionnaire, which is simply a list of tokens which point to the questions and responses to each question as well as tokens for program control or math operations. As will be apparent to those skilled in the art, if a question or response is repeated within the questionnaire, only a pointer need be repeated in the program list, not the entire question.

According to another preferred arrangement, there is provided a system, substantially as defined above, wherein the questionnaire which is transmitted to the handheld can be incrementally updated on each networked handheld **28**, rather than resending the entire questionnaire. For example, if a question is modified or replaced, the new question and a new list are the only information which need to be transmitted to the handheld device **28**. This incremental update capability dramatically reduces the quantity of computer instructions required to update a form. It should be noted that, if connection **34** is present, the program update will take place virtually at the same time the client finishes questionnaire **40** at computer **22**. If the network connection is unavailable, the update will happen automatically as soon as the connection **34** is restored.

Turning next to FIG. **3**, in a preferred embodiment the user will initiate the execution of the questionnaire according to instructions previously provided to him or her. For example, the user might be instructed to initiate the questionnaire as soon as he or she pulls into the drive-in lane of a take-out eatery. This would be the case if the questionnaire were designed to collect information regarding service at that establishment. In such an event, the questionnaire might contain questions related to service time, cleanliness, friendliness of the employees, etc., all of which would potentially be of interest to the owner/client. The user will preferably respond to each question in turn, the questions being presented according to the logic defined by the client and built into the questionnaire. In some instances, the text of the question might instruct the user to perform acts and/or wait until a certain event happens before responding (e.g., "Pull up to the take-out window. How long was it before you received your order?") The user's responses to the items in the questionnaire are stored within the handheld **28** as they are collected. In some cases, the questionnaire logic might allow the user to skip questions and (optionally) return to them later. Additionally, the questionnaire designer might include a token that initiates a final review of the data collected from the user in this instance to make certain that all "required" questions, (which have preferably been so designated by marking them with the appropriate token) have been answered. Failure by the user to respond to a required question will result in the OIS prompting the user again for a response.

Several options are available for the transmission of responses from handheld **28** to server **24**. First, regardless of the availability of connection **34**, responses may be stored locally at handheld **28** until the form is fully completed and then sent as a batch to server **24**. This transfer may optionally occur automatically, or upon direction of the user as specified by the client during the creation of the form. If the link is not available at the time of completion of the form, transmission will be automatically delayed until connection **34** is restored.

Alternatively, selected responses, or all responses, may be configured to transmit immediately upon entry, assuming of course that connection **34** is available. This option is particularly important where the user of handheld **28** has entered information which might be indicative of a problem with a process or indicate an emergency. Again, if connection **34** is unavailable, immediate transmissions will also be delayed until a connection is available.

As data from a handheld is received at server **24** it is processed, as necessary, and placed in a database where it can be accessed via the Internet **26**. A client can then use a computer **22** with Internet access to review or use the data from virtually anywhere in the world.

Turning next to FIG. **4**, wherein a prior art system built around paper forms is shown, in the past, a paper form had to be created, printed, and delivered to a user of the form. Armed with the form, the user had to complete the assigned task and complete the form reflecting observations made during the task. If the user delayed in filling out the form, these observations were subject to the inaccuracies associated with human memory. A completed form was then typically delivered to yet another person for data entry before the information was finally available to others in the company. As will be appreciated by those familiar with such operations, whether a result of unreadable forms or a result of human error at data entry, this step is responsible for a significant level of errors.

Turning next to FIG. **5**, in contrast to prior systems, with the present system, a form may be entered on-line, the form is automatically sent to the handheld computer of the user, usually within seconds, the user enters data directly at the location of the user's assigned task, eliminating memory errors, and made available to others in the organization in virtually a real time fashion. Thus, not only is the data almost instantly available, at least two sources of error, the memory of the user and data entry, have been eliminated.

An example of where the inventive system is particularly useful is in the area of mystery shoppers. Many restaurant chains and retail chain stores employ mystery shoppers to patronize one of the chain's establishments and report on the experience. In the area of fast food, a mystery shopper might, for example, use the drive through window to purchase a breakfast sandwich and a cup of coffee.

Prior to the trip to the restaurant, an employee of the client restaurant develops a questionnaire and enters it on the web site of the ASP that is providing the mystery shopper support service. In this case, the restaurant is interested in the waiting time of their patrons, the service provided to their patrons, and the quality of the food served. A questionnaire is designed to elicit such information from the shopper/user. The results of the mystery shopper's experience will be compared to quality standards established for the entire chain and used to rate the franchisee/owner of particular restaurants.

As the mystery shopper enters the parking lot, the shopper will be prompted to enter a store number or location. If the handheld computer is equipped with a GPS receiver, this information could be entered automatically. Of course the time and date from the computer's real time clock are preferably recorded in the form. As the shopper reaches the end of the drive through line, she starts a timer on the hand held computer, preferably by "tapping" on the face of the handheld in the appropriate region of the screen. When the speaker is reached, the first timer is stopped and a second timer is started.

If the shopper is asked to wait before ordering, a second timer is started and a third timer is started. Upon a request for her order, the mystery shopper stops the previous timers and yet a fourth timer is started. She orders her breakfast sandwich and coffee and pulls forward in line. While sitting in line, the handheld computer asks if the speaker could be clearly understood, if the menu was in good shape, and if the area around the menu appeared neat and clean.

Upon reaching the window, the shopper presses a button which stops the fourth timer and starts a fifth timer. As her money is taken, the fifth timer is stopped and yet a sixth timer is started. She pays with a twenty dollar bill and, upon receiving her change, notes the accuracy of her change, whether the person at the window is pleasant, stops the sixth timer and starts a seventh timer.

Upon receiving her food the seventh timer is stopped and she pulls into a parking place to sample the food and measure the temperature of the coffee with a temperature probe attached to her handheld computer. After entering her impression of the sandwich, the computer asks a few questions about the number of cars in the parking lot and the general appearance of the store.

As the shopper enters the last response, the CDPD modem attached to her handheld contacts the ASP and delivers the collected data which is forwarded to a database where it is accessible by the staff of the restaurant chain, only seconds after the shopper has taken her first bite of the sandwich.

Note that the user's interaction with the handheld in the previous example was all defined by logic that the client has incorporated into the questionnaire when it was designed. The text of the directions to the user (e.g., "Pull up to the drive-in window.") has been designed into the questionnaire. Additionally, preferably there will be tokens that represent "timers" which are designed to make it easy for the user to enter elapsed time information in response to a question (e.g., the user might be asked to tap the screen a first time to start the timer running and a second time to stop it, with the elapsed time being automatically calculated and stored as a response to a client question). Clearly, a goal of the instant system is to provide a client with the tools necessary to quickly and easily construct a complex questionnaire which presents the user with questions which are adaptively selected according to the wishes of the designer.

A second example of where the inventive system is particularly useful is the area of transfer by a medical service provider of a patient's medical information to an insurance company following treatment. In this example, data is exchanged between computers (handhelds, desktops, laptops, etc.) at different locations in a secure manner without providing an outside party access to the secure internal computer network of the medical service provider (MSP). The medical service provider is preferably a hospital, however, it is understood that this term could include clinics, minor emergency centers, physician's offices or any such provider of medical care/treatment.

Modern medical service providers are continually striving to develop methods of transferring medical records and data to insurance companies for rapid claims processing which requires the minimum of manual forms generation, handling, processing, and data entry. Moreover, pressure, both publicly and legislatively, is being applied to the healthcare industry as a whole to protect the privacy of this data including confidential patient information. As a result, transmission of medical information in secure, generally encrypted formats is required. However, such methods of data transfer require a high level of coordination between the medical service providers and the insurance companies, both of which are reluctant to allow the other, and especially third parties, access to their databases and network hardware necessary to achieve these levels of coordination.

The present system can be employed to manage the data flow in a manner that provides secure data transfer between parties without the necessity of either party allowing outside access to its respective data storage systems. In this embodiment, the medical service provider can use the system to design or update the medical forms as described above or contract with the ASP to develop and update such forms.

Referring to FIG. **6**, a system diagram is shown depicting medical services provider **120** (MSP), ASP **130** and insurance companies **140**, **142**, and **144**. As stated, the medical forms can be designed and/or updated seamlessly by the MSP or ASP as shown in **122**. The computers of MSP would be equipped with the inventive OIS thereon to allow forms design, branching logic, and cryptic data transfer at **122**. Once the medical form is designed, medical information can be entered onto the form(s) in the system following treatment by the MSP. Once entered, the data is converted to tokenized form by the OIS for encrypted transfer to the ASP **130** according to step **124**. In this way, a patient's medical data is continuously, seamlessly and securely transferred between MSP **120** and ASP **130**.

Once the ASP **130** receives the tokenized data from MSP **120** pursuant to transfer **124**, the data is stored in a standard database or a database customized for each insurance company within ASP **130**. In a preferred arrangement, the ASP will then alert one or more of the relevant insurance providers **140**, **142**, and/or **144** that data is present and available for immediate retrieval from the database of ASP **130**. In the alternative, the system could be embodied such that insurance providers **140**, **142**, and **144** would periodically query ASP **130** on a set time interval regarding the presence of information.

At the time insurance providers **140**, **142**, and/or **144** are aware that data is present and available from ASP **130**, the insurance provider can access the ASP via a global computer network such as the Internet for retrieval of such information. Typically, access to information maintained by ASP **130** is restricted by password or other similar security measures. Insurance provider **140**, **142**, and/or **144** can then download data from ASP **130** which is either encrypted in a standard format or in a format which is customized for the insurance provider (and may also be encrypted). The download step is depicted by arrows **132**, **134**, and **136**, respectively.

In this embodiment, the customer of ASP **130**, typically MSP **120**, would be billed for the transaction or by the volume of data transmitted.

Accordingly, a secure method of transfer of medical information between MSP **130** and insurance providers **140**, **142**, and/or **144** is defined using the method and apparatus of the present invention.

By way of example and not limitations, various preferred embodiments of the instant invention will include a number of desirable features or traits such as: 128-bit CerticomT end-to-end wireless security; ability of the administrator to clean erroneous data; all data and administrative transactions on one or more secure servers; form question responses are time stamped; centralized online repository of all form responses; the complete form is available for review or update on the Web; context-sensitive help; from the customer's perspective, the system is scalable and flexible; users, questionnaires, and responses manageable as groups; data exportation to CSV, XLS, XML, as well as any other format or external application; ability to define multiple form administrators; forms deployable wirelessly over the Internet; error checking for dropped connection in a loosely networked environment; a provider of the service can offer secondary services such as form design consulting services; partially completed forms can be saved and restarted; OIS allows data to be gathered in virtually any form factor, i.e. web, handheld, phone, laptop, and the like; the client can inspect individual responses from a form; multiple forms can be made available on same device; online data report generation and publishing from gathered responses; optional authentication of users; responses can be subjected to bounding and validation logic; real-time accessibility to form responses from an Internet connected desktop; responses retrievable or accessible anywhere in the world via a provider's web site; robust question branching logic; unlimited administrative control of the user, e.g. a user can be prevented from completing a form more than once; administrative hierarchy allowing some administrators to view other administrators' data, if allowed; web based service eliminates the need for client installation; archival of old forms and responses; user interfaces brandable with corporate identity;

ability to clone, or modify, existing forms into a new form; ability to create summary reports with informative charts; customizable reports can be designed to meet clients' specific needs; definable start and stop dates for forms allow control of a time frame over which data can be gathered; ability to include pictures in questions; responses from various forms can be merged into a common report; phone call completion of forms; administrators can be provided with predefined question and form libraries; responses can be reviewed prior to submitting; print form responses from the remote computer; administrative control of questionnaire aesthetics; software developers kit can be provided by the service provider; language controls available during question development, i.e. spell check, thesaurus, translation of multi-language forms, extended character sets, etc.; various events can be triggered from within a form; and reports can be viewed on the remote computer.

Thus, the present invention is well adapted to carry out the objects and attain the ends and advantages mentioned above as well as those inherent therein. While presently preferred embodiments have been described for purposes of this disclosure, numerous changes and modifications will be apparent to those skilled in the art. Such changes and modifications are encompassed within the spirit of this invention.

What is claimed is:

1. A method for managing data including the steps of:

(a) creating a questionnaire comprising a series of questions;

(b) tokenizing said questionnaire; thereby producing a plurality of tokens representing said questionnaire;

(c) establishing a first wireless modem or wireless LAN network connection with a remote computing device;

(d) transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection;

(e) terminating said first wireless modem or wireless LAN network connection with said remote computing device;

(f) after said first wireless modem or wireless LAN network connection is terminated, executing at least a portion of said plurality of tokens representing said questionnaire at said remote computing device to collect a response from a user;

(g) establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server;

(h) after said second wireless modem or wireless LAN network connection is established, transmitting at least a portion of said response from the user to said server via said second wireless modem or wireless LAN network connection; and

(i) storing said transmitted response at said server.

2. The method for managing data of claim 1 further comprising the step of:

(j) translating said response to a format recognizable by a particular computer program; and

(k) accessing the translated response from a computer executing said particular computer program.

3. The method for managing data of claim 1 wherein step (a) includes the substeps of:

(a) creating a questionnaire by:

(i) entering a series of questions into a questionnaire design computer program;

(ii) identifying within said questionnaire design computer program the type of response allowed for each question of said series of questions; and

(iii) identifying within said questionnaire design computer program a branching path in said questionnaire for each possible response to each question of said series of questions.

4. The method for managing data of claim 1 wherein step (b) includes the substeps of:

(b) tokenizing said questionnaire thereby producing a plurality of tokens representing said questionnaire by:

(i) assigning at least one token to each question of said series of questions;

(ii) assigning at least one token to each response called for in said series of questions to identify the type of response required; and

(iii) assigning at least one token to each branch in said questionnaire to identify the required program control associated with said branch.

5. A method for modifying a questionnaire used in data management according to the method of claim 1 including the steps of:

(a) making at least one incremental change to a portion of the questionnaire;

(b) tokenizing said at least one incremental change to said questionnaire;

(c) transmitting at least a portion of said tokens resulting from step (b) to a remote loosely networked computing device, said transmitted tokens comprising less than the entire tokenized questionnaire; and,

(d) incorporating said transmitted tokens into said questionnaire at said loosely networked remote computing device, thereby modifying said questionnaire.

6. A method for managing data according to claim 1, wherein said first wireless modem or wireless LAN network connection and said second wireless modem or wireless LAN network connection are a same wireless modem or wireless LAN network connection.

7. The method of claim 1 further including performing at least the steps (c)-(k) for at least two different remote computing device types using the same tokens.

8. A method for managing data transfers between computers including the steps of:

(a) creating a questionnaire at a first site in a first computer;

(b) tokenizing said questionnaire, thereby producing a tokenized questionnaire;

(c) bringing a remote computer into electronic communication with said first computer;

(d) transmitting said tokenized questionnaire to said remote computer;

(e) removing said remote computer from electronic communication with said first computer;

(f) within said remote computer, using said transmitted tokenized questionnaire to obtain at least one user response;

(g) storing said at least one user response within said remote computer;

(h) modifying said questionnaire with incremental changes at a second computer located at a second site;

(i) placing said remote computer into electrical communication with said second computer;

(j) transmitting said incremental changes from said second computer to said remote computer;

(k) modifying said transmitted tokenized questionnaire in said remote computer with said incremental changes, thereby creating a modified tokenized questionnaire;

(l) removing said remote computer from electronic communication with said second computer;

(m) within said remote computer, using said modified tokenized questionnaire to obtain at least one additional user response;

(n) placing said remote computer into electronic communication with a server;

(o) transmitting said at least one user response to said server;

(p) transmitting said at least one additional user response to said server;

(q) storing said transmitted at least one user response and said at least one additional user response at said server;

(r) preparing a report using any of said at least one user response and said at least one additional user response; and,

(s) displaying at least a portion of said report on a visually perceptible medium;

(t) performing at least steps (d)-(p) using at least two different remote computing device types using the same tokens.

9. The method for managing data transfers between computers according to claim **8** wherein said first computer and said second computer are a same computer.

10. The method for managing data transfers between computers according to claim **9** wherein said server and said first computer are said same computer.

11. A method for collecting survey data from a user comprising the steps of:

(a) creating a questionnaire comprising a series of questions;

(b) tokenizing said questionnaire; thereby producing a plurality of tokens representing said questionnaire;

(c) storing said plurality of tokens on a computer readable medium on a first computer;

(d) placing a handheld remote computing device into electronic communication with said first computer;

(e) transmitting said plurality of tokens to said handheld remote computing device;

(f) taking said handheld remote computing device out of electronic communication with said first computer;

(g) after said handheld remote computing device has been taken out of electronic communication with said first computer,

(g1) executing at least a portion of said plurality of tokens representing said questionnaire on said handheld remote computing device to collect a response from a user, and,

(g2) storing within said remote computing device said response from the user;

(h) placing said handheld remote computing device into electronic communication with a second computer;

(i) transmitting at least a portion of said response stored within said handheld remote computing device to said second computer; and,

(j) forming a visually perceptible report from any of said at least a portion of said response so transmitted.

12. A method for collecting survey data from a user according to claim **11**, wherein step (j) comprises the step of printing a report from any of said response to transmitted.

13. A method for collecting survey data from a user according to claim **11**, wherein said first computer and said second computer are a same computer.

14. A method for modifying a questionnaire used in data management according to the method of claim **11**, further comprising the steps of:

(k) making at least one incremental change to a portion of said questionnaire;

(l) tokenizing said at least one incremental change to said questionnaire;

(m) transmitting at least a portion of said tokens resulting from step (k) to said remote handheld computing device, said transmitted tokens comprising less than the entire tokenized questionnaire; and,

(n) incorporating said transmitted tokens into said questionnaire at said remote computing device, thereby incrementally changing said questionnaire.

* * * * *

PA-C

US006584464B1

(12) **United States Patent**
Warthen

(10) **Patent No.: US 6,584,464 B1**
(45) **Date of Patent: Jun. 24, 2003**

(54) **GRAMMAR TEMPLATE QUERY SYSTEM**

(75) Inventor: **David Warthen**, Albany, CA (US)

(73) Assignee: **Ask Jeeves, Inc.**, Berkeley, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/272,717**

(22) Filed: **Mar. 19, 1999**

(51) **Int. Cl.**^{7} **G06F 17/30**
(52) **U.S. Cl.** **707/4**; 707/5; 707/10
(58) **Field of Search** 707/1–10, 100, 707/532; 704/79; 706/55, 934

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,270,182 A	5/1981	Asija	704/8
4,358,824 A	11/1982	Glickman et al.	707/5
4,730,270 A	3/1988	Okajima et al.	704/2
4,837,689 A	6/1989	Tanaka et al.	704/8
4,839,853 A	6/1989	Deerwester et al.	707/5

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0196064	3/1986
WO	WO 99/66378	12/1999
WO	WO 00/57302	* 9/2000

OTHER PUBLICATIONS

Peter G. Annick, "Integrating Natural Language Processing and Information Retrieval in a Troubleshooting Help Desk", IEEE Expert, vol. 8, No. 6 pp. 9–17 (Dec., 1993).

Anon., "SQL Statement Library List", IBM Technical Disclosure Bulletin, vol. 36, No. 1, pp. 82–83 (Jan., 1993).

Jakobovits et al., "Implementing Multi–level Queries in a Database Environment for Vision Research", SPIE, vol. 2420, pp. 95–103 (1995).

Christopher Wisdo, "Position Paper", AAAI Fall Symposium—AI Applications in Knowledge Navigation and Retrieval, p. 154 (1995).

Swanson, D.R. "Word Correlation and Automatic Indexing Progress Report No. 2." Submission to the Council on Library Resources by Ramo–Wooldridge; Dec. 21, 1959.

Swanson, D.R. "Word Correlation and Automatic Indexing, Phase 1 Final Report: An Experiment in Automatic Text Searching." Submission to the Council on Library Resources by Ramo–Wooldridge; Apr. 30, 1960.

Swanson, D.R.. "Searching Natural Language Text by Computer Machine Indexing and Text Searching Offers an Approach to the Basic Problems of Library Automation." *Science* 132 (1960): 1099–1104.

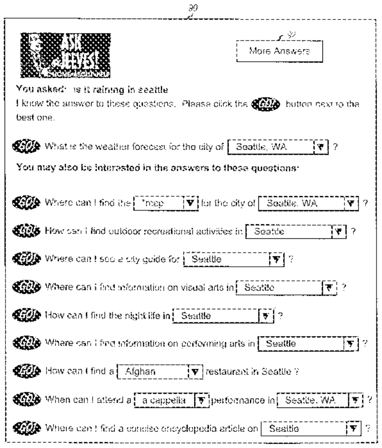
(List continued on next page.)

Primary Examiner—Hosain T. Alam
Assistant Examiner—Anh Ly
(74) *Attorney, Agent, or Firm*—Townsend, Townsend & Crew LLP

(57) **ABSTRACT**

An information server directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer. In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.

9 Claims, 10 Drawing Sheets



You asked: is it raining in seattle
I know the answer to these questions. Please click the GO! button next to the best one.

What is the weather forecast for the city of Seattle, WA ?

You may also be interested in the answers to these questions:

Where can I find the *map for the city of Seattle, WA ?
How can I find outdoor recreational activities in Seattle ?
Where can I see a city guide for Seattle ?
Where can I find information on visual arts in Seattle ?
How can I find the night life in Seattle ?
Where can I find information on performing arts in Seattle ?
How can I find a Afghan restaurant in Seattle ?
When can I attend a a cappella performance in Seattle, WA ?
Where can I find a concise encyclopedia article on Seattle ?

U.S. PATENT DOCUMENTS

4,931,926	A		6/1990	Tanaka et al.	704/8
4,931,935	A		6/1990	Ohira et al.	704/8
5,067,070	A		11/1991	Miyao et al.	707/540
5,115,390	A		5/1992	Fukuda et al.	700/17
5,123,103	A		6/1992	Ohtaki et al.	707/5
5,140,522	A		8/1992	Ito et al.	704/2
5,148,541	A		9/1992	Lee et al.	707/2
5,161,225	A		11/1992	Abraham et al.	707/2
5,197,005	A		3/1993	Shwartz et al.	707/2
5,259,766	A		11/1993	Sack et al.	434/362
5,261,100	A		11/1993	Fujinami et al.	717/122
5,265,065	A		11/1993	Turtle	707/4
5,268,839	A		12/1993	Kaji	704/3
5,297,039	A		3/1994	Kanaegami et al.	707/5
5,338,976	A		8/1994	Anwyl et al.	704/2
5,369,575	A		11/1994	Lamberti et al.	704/9
5,383,392	A		1/1995	Kowalewski et al.	101/183
5,384,894	A		1/1995	Vassiliadis et al.	706/52
5,404,295	A		4/1995	Katz et al.	707/2
5,418,948	A		5/1995	Turtle	707/4
5,442,780	A		8/1995	Takanashi et al.	707/1
5,454,106	A		9/1995	Burns et al.	707/4
5,457,794	A		10/1995	Azumatani et al.	707/2
5,519,608	A	*	5/1996	Kupiec	707/3
5,523,945	A		6/1996	Satoh et al.	704/9
5,542,090	A		7/1996	Henderson et al.	707/2
5,715,444	A	*	2/1998	Danish et al.	707/4
5,802,493	A	*	9/1998	Sheflott et al.	705/1
5,819,265	A	*	10/1998	Ravin et al.	707/5
5,836,771	A		11/1998	Ho et al.	434/362
5,870,737	A		2/1999	Dockter et al.	707/4
5,878,423	A	*	3/1999	Anderson et al.	707/100
5,893,219	A	*	4/1999	Danish et al.	707/3
6,081,774	A	*	2/2000	de Hita et al.	707/3
6,076,088	A	*	6/2000	Paik et al.	707/5
6,289,513	B1	*	9/2000	Bentwich	707/6
6,182,068	B1	*	1/2001	Culliss	707/5

OTHER PUBLICATIONS

Bobrow, D.G. "Natural Language Input for a Computer Problem Solving System." Ph.D. diss., Massachusetts Institute of Technology, 1964. NTIS: ADA015689.

Salton, G. "Information Storage and Retrieval" Scientific Report No. ISR–11 to The National Science Foundation, Jun. 1966.

Rubinoff et al. "Easy English, a Language for Information Retrieval Through a Remote Typewriter Console." University of Pennsylvania, Moore School of Electrical Engineering, Apr. 1967. NTIS: AD660081.

Green, C.G. and B. Raphael. "Research on Intelligent Question–Answering Systems." SRI Project 6001, Scientific Report 1. Stanford Research Institute, Menlo Park, CA, May 1967, NTIS: AD656789.

Stogniy, A.A. and V.N. Afanassiev. "Some Design Problems for Automatic Fact Information Retrieval and Storage Systems." In *Mechanized Information Storage Retrieval and Dissemination,* 289–299. New York: North–Holland, 1968.

Vallee, J.F. et al. "Retrieval Formulae for Inquiry Systems." *Inform. Stor. Retr. 4* (1968): 13–26.

Schank, R.C. "The Use of Conceptual Relations in Content Analysis and Data Base Storage." Tracor Inc., Austin, TX, Jan. 1968. NTIS: AD666992.

Raphael, B. "Research on Intelligent Question–Answering Systems." SRI Project 6001, Final Report. Stanford Research Institute, Menlo Park, CA, May 1968. NTIS: AD671970.

Kellogg, C. "On–Line Translation of Natural Language Questions into Artificial Language Queries." *Inf. Stor. Retr.* 4 (Aug. 1968): 287–307.

Treu, S. "The Browser's Retrieval Game" *American Documentation* (Oct. 1968): 404–410.

Wilde, D.U. "Computer–Aided Strategy Design Using Adaptive and Associate Techniques." American Society for Information Science (ASIS) Annual Meeting, Columbus, Ohio. *Information Transfer* 5 (Oct. 1968):175–178.

Berul, L. "Document Retrieval." Chapter 7 in *Annual Review of Information Science and Technology,* vol. 4 (American Society for Information Science), 204–227. Chicago: Encyclopaedia Britannica Inc., 1969.

Merta, A. "INDORES 4—A Mechanized System of Information Retrieval in the Field of Information Science," 103–106. Hamden, CT: Archon Books, The Shoestring Press, 1970.

Winston, P.H. "Learning Structural Descriptions from Examples." Ph.D. diss., Massachusetts Institute of Technology, 1971.

Cleverdon, C. "Process in Documentation." *Journal of Documentation,* 26 (1): 55–67 (1970).

Spencer, D.A. "A study of the Description Matching Problem." Master's Thesis, Massachusetts Institute of Technology, 1971.

Corbett, L. "Controlled versus Natural Language: A Report on the Great Debate." *The Information Scientist* (United Kingdom): 115–120 (Sep. 1971).

Moreau, M. et al. "Recherché Bibliographiques en Language Naturel." *Automatisme* 16 (Dec. 1971): 628–637.

Winograd, T. "Five Lectures on Artificial Intelligence." Report STAN CS 74–459. Stanford University, Department of Computer Science, Sep. 1974. NTIS: ADA000085.

Belzer, J. et al. "Regeneration of Information Rather than Information Retrieval 'Concept Creation Method'." Final Report on the Regeneration Project. University of Pittsburgh, 1974. NTIS: ADA003617.

Kitov, A.I. and V.V. Kostyuk. "Retrieval of Documents Recorded in Natural Language in Computer Storage." *Automatic Documentation and Mathematical Linguistics* 9 (4): 25–28 (1975).

McMaster et al. "Computer Acquisition of Natural Language: A Review and Prospectus." *Int. J. Man–Machine Studies* 8 (1976): 367–396.

Waltz, D.L. "Natural Language Access to a Large Data Base." *Computers and People* (Apr. 1976): 19–26.

Unknown. "Chat at the Highest Level." *Scientific American* 235 (4): 60, 64 (1976).

Zloof, M.M. "Query–by–Example: A Data Base Language." *IBM System Journal* 4 (1977): 324–343.

Brecht, W. "Anwendungsspezifische Einflußgrößen auf natürlichsprachliches Informationssystem—(Application Specific Aspects of a Natural Language Information System)." *Sprach und Datenverarbeitng: International Journal for Language Data Processing 1* (2): 93–97 (1977).

Lehnert, W. "The Process of Question Answering." Research Report #88. Yale University, Department of Computer Science, May 1977. NTIS: ADA040559.

PHLIQA Project Group. "PHLIQA 1—A Question–Answering System for Data–Base Consultation in Natural English I." *Philips Technical Review,* 38(9): 230–239 (1978–79).

PHLIQA Project Group. "PHLIQA 1—A Question–Answering System for Data–Base Consultation in Natural English II." *Phillips Technical Review,* 38 (10): 269–284 (1978–79).

Schank, R.C. "Computer Understanding of Natural Language." *Behavior Research Methods & Instrumentation* 10 (2): 132–138 (1978).

Dilger, W. and G. Zifonun. "The Predicate Calculus–Language KS as a Query Language." In *Logic and Data Bases,* 377–408. New York: Plenum Press, 1978.

de Heer, T. "Een Sloridge Maar Toegeerijke Zoekstrategie Voor Computers" (A Loose But Indulgent Retrieval Strategy for Computers). The Hague: Institut TNO voor Wiskunde Informatieverwerking en Statistiek, Mar. 1978. NTIS: N8027211.

de Heer, T. "Experiments with Multi–Lingual Queries in a Multi–lingual Data Base." The Hague: Institut TNO voor Wiskunde Informatieverwerking en Statistiek, Jul. 1978. NTIS: N8023209.

Woods, W.A. "Semantics for a Question–Answering System." Ph.D. diss., Harvard University, Division of Engineering, 1979. New York: Garland Publishing, Inc., 1979.

Schank, R.C. and W. Lehnert. "Computer Understanding of Stories." In *Human and Artificial Intelligence,* ed. F. Klix, 135–139. New York: North Holland Publishing Co., 1979.

Winston, P. and R.H. Brown, eds. "Progress in Artificial Intelligence " 1978, vol. 1. Report to the U.S. Office of Naval Research. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1979. NTIS: ADA068838.

Buneman, P.O. "The Multiple Path Problem in Database Schemata." Technical Report. University of Pennsylvania, Wharton School, Department of Decision Sciences, Jan. 1979. NTIS: ADA067147.

Schank, R.C. and J. Kolodner. "Retrieving Information from an Episodic Memory or Why Computers' Memories Should Be More Like People's." Research Report #159. Yale University, Department of Computer Science, Jan. 1979. NTIS: ADA071445.

Gardner, A. et al. "Natural Language Understanding." In *Handbook of Artificial Intelligence,* ed. A. Barr. and A. Feigenbaum. Stanford University, Department of Computer Science, Report No. STAN–CS–79–754, Jul. 1979. NTIS: ADA076873.

Wilensky, R. "Understanding Complex Situations." *Proceedings of the Sixth International Joint Conference of Artificial Intelligence,* Tokyo, 2: 954–959 (1979).

Niehoff, R. et al. "The Design and Evaluation of a Vocabulary Switching System for Use in Multi–Based Search Environments." Final Report to the National Science Foundation. Battelle Columbus Labs, Ohio, Feb. 1980. NTIS: PB80154479.

Berwick, R. "Learning Structural Descriptions of Grammar Rules from Examples." Master's Thesis, Massachusetts Institute of Technology, 1980.

Katz, B. "Three–Step Procedure for Language Generation." A.I. Memo No. 599. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Dec. 1980.

Schank, R.C. et al. "Conceptual Information Retrieval." Research Report #190. Yale University, Department of Computer Science, Dec. 1980. NTIS: ADA095372.

Cullingford, R.E. "Integrated Knowledge Sources for Computer 'Understanding' Tasks." IEEE publication: *Transactions on Systems, Man, and Cybernetics* SMC–11(1):52–60 (Jan. 1981).

Vjekoslav, P. "Neki Problem Jezicnog Komuniciranja S Bazama Podataka." (English abstract on page 285). *Informatologia Yugoslavica* 14 (3–4): 281–285 (Jun. 1982).

Tou, F. "RABBIT: An Intelligent Database Assistant." *Proceedings of the National Conference on Artificial Intelligence,* 314–318. American Association for Artificial Intelligence, Pittsburgh, PA 1982.

Katz, B. and P. Winston. "A Two–Way Natural Language Interface." In *Integrated Interactive Computing Systems,* ed. P.Degano and E. Sandewall, 279–293. New York: North Holland Publishing Co., Sep. 1982.

Koloder, J.L. "Indexing and Retrieval Strategies for Natural Language Fact Retrieval." *ACM Transactions on Database Systems,*8 (3): 434–464 (1983).

Adelsberger, H.H. "Prolog as a Simulation Language." *Proceedings of the Winter Simulation Conference 1984,* ed. S. Sheppard, 501–504 (1984) Publisher unknown: author affiliated with Texas A&M University, Department of Computer Science, College Station, TX.

Brenner, E.H. et al. "American Petroleum Institute's Machine–Aided Indexing and Searching Project" *Science Technology Libraries* 5 (1): 49–62 (1984).

Stevens, L. "Natural–Language and Voice–Recognition Systems." Chapter 3 in *Artificial Intelligence the Search for the Perfect Machine,* 21–39. New Jersey: Hayden Book Co., 1984.

Jones, T.I. "Automatic Search Term Variant Generation." *Journal of Documentation,* 40 (1): 50–66 (1984).

Chamis, A. "Online Database Search Strategies and Thesaural Relationship Models." Ph.D. diss., Case Western Reserve University, 1984.

Pao, Y.–H. and C.–S. Yang. "A Computer Intelligent System for Understanding Chinese Homonyms." *The First International Conference on Computers and Applications,* Beijing, (Jun. 1984): 841–847. The Computer Society Press (IEEE), Catalog No. 84CH2039–6.

Winston, P. "Language Understanding" Chapter 9 in *Artificial Intelligence,* 2nd Edition, 291–334. Reading, MA: Addison–Wesley Publishing Co., Jul. 1984.

Cullingford, R. and M. Passani. "Word–Meaning Selection in Multiprocess Language Understanding Programs." *Transactions on Pattern Analysis and Machine Intelligence* (IEEE), PAMI–6 (4): 493–509 (1984).

Mitkov, R. "The Sublanguage Approach: A Key to Realistic Natural Language Processing." *Proceedings of the Sixth International Conference on Artificial Intelligence: Methodology, Systems, Applications,* ed. P. Jorrand and V. Sgurev. New Jersey: World Scientific, 1984.

Harris, L. "Teaching Computers English Proves Easier Than Training People." *Mini–Micro Systems* 17(15): 163–172 (1984).

Giter, D.M. et al. "Rapid Processing of Symbolic Information in a High–Level Language." *Automatic Documentation and Mathematical Linguistics* (*Nauchno–Tekhnicheskaya Informatsiya, Seriya* 2) 19 (4): 73–75 (1985).

Sebrechts, M. and P.H. Gross. "Programming in Natural Language: A Descriptive Analysis." *Behavior Research Methods, Instruments and Computers 17* (2) 268–274 (1985).

Jarke, M. and J. Krause. "New Empirical Results of User Studies with a Domain Independent Natural Language Query System."In *Artificial Intelligence: Methodology, Systems, Applications,* ed. W. Bibel and B. Petkoff, 149–191. North Holland: Elsevier Science Publishers, B.V., 1985.

Hendrix, G.G. and E.D. Sacerdoti. "Natural Language Processing: The Field in Perspective." Chap. 7 in *Applications in Artificial Intelligence,* ed. S.J. Andriole. USA: Petrocelli Books Inc., 1985, 149–191.

Winston, P. "The LISP Revolution LISP Is No Longer Limited to a Lucky Few." *Byte* (Apr. 1985): 209–217.

Mahugh, D. "Understanding Fourth–Generation Languages." *UNIX/World 2* (6): 36–47 (1985).

Christopher, A. "Artificial Intelligence and Computer Graphics: A Relationship in Its Infancy." *Computer Graphics World* (Aug. 1985): 8–20.

Cheng, C. "Microcomputer–Based User Interface for Library Online Catalogue." Technical Report No. LLL–T–8–85, (Oct. 1985): 4–15. Language Learning Laboratory School of Humanities University of Illinois–Urbana, 1985.

Chamis, A. "The Usefulness of Switching Vocabularies for Online Databases." *Proceedings of the American Society for Information Sciences Annual Meeting,* Oct. 1985, ed. C.A. Parkhurst, 22: 311–314. New York: Knowledge Industry Publications, 1985.

Williamson, M. "Natural–Language Inquiry Systems Part 2."*PC Week* (Oct. 8, 1985): 41–42.

Snyers, D. and D. Thayse. "Artificial Intelligence Methods for Speech Understanding." Communications Technology Symposium, Munich, Apr. 1986, German Chapter of the IEEE. Berlin: VDE–Verlag GmbH, 1986, pp. 103–107.

Cullingford, R. "SAM (Script Applier Mechanism)." In *Readings in Natural Language Processing,* ed. Grosz et al., 627–649. (Los Altos, CA: Morgan Kaufman Publishers, Inc., 1986.

Lehnert, W.G. "A Conceptual Theory of Question Answering." In *Readings in Natural Language Processing,* ed. Grosz et al., 651–657. Los Altos, CA: Morgan Kaufman Publishers, Inc., 1986.

Michalski, R. and P. Winston. "Variable Precision Logic." *Artificial Intelligence 29* (1986): 121–146.

Bates, M. et al. "The IRUS Transportable Natural Language Database Inferface." *Expert Database Systems,* ed. L. Kerschberg, 617–630, Menlo Park: Benjamin Cummings Publishing Co., Inc., 1986.

Johnson, M.C. "A Grammar–Based Technique for Genetic Search and Optimization." Ph.D. diss., The College of William and Mary, 1996.

Frame, C.E. "Personal Computer and Workstation Operating Systems Tutorial." Master's Thesis, Naval Postgraduate School, 1986. NTIS: ADA280132.

Nagao, M. "Artificial Intelligence and Communication." *PTC '86 Proceedings: Telecom—Asia, Americas, Pacific,* 142–146. Pacific Telecommunications Council, Jan. 1986.

Tomiski, E.A. and Klahr, M. "How Artificial Intelligence Has Developed." *Journal of Systems Management* (May 1986): 6–10.

Zaki, M. and R. Salama. "Linkage Versus Integration for Binding Database and Interactive Graphics Systems." *Inform. Systems 12* (3): 271–280 (1987).

Bailey, D.L. "Similarity Networks as a Means of Indexing and Retrieving Descriptions." Bachelor's Thesis, Massachusetts Institute of Technology, Jun. 1986.

van Rijsbergen, C.J. "A New Theoretical Framework for Information Retrieval." *SIGIR Forum* (Proceedings of the 9th International SIGIR Conference in Research and Development in Information Retrieval) 21 (1–2, Fall–Winter 1986–87): 194–200.

Chase, Rosen and Wallace, Inc. "CASPIR Computer Aided Searching and Patent Image Retrieval Users Guide." Examiners Version, Sep. 1, 1996. Alexandra, VA: Chase, Rosen and Wallace, Inc., 1996.

Fagan, J.L. "The Effectiveness of a Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval." *Journal of the American Society for Information Science* 40 (2): 115–132, (1986).

Tennant, H. "Menu–Based Natural Language." In *Encyclopedia of Artificial Intelligence,* vol. 1, ed. S.C. Shapiro et al., 594–597. New York: John Wiley & Sons, 1987.

Fox, E.A. "Testing the Applicability of Intelligent Methods for Information Retrieval." *Information Services & Use* (1987): 119–138.

Grosz, B.J. et al. "Team: An Experiment in the Design of Transportable Natural–Language Interfaces." *Artificial Intelligence* 32 (1987): 173–243.

Katz, B. and R. Brooks, Jr. "Understanding Natural Language for Spacecraft Sequencing." *Journal of the British Interplanetary Society,* 40 (1987): 451–460.

Teskey, F.N. "Enriched Knowledge Representations for Information Retrieval." *Proceedings of the Tenth Annual International ACMSIGIR on Research & Development in Information Retrieval,* New Orleans (Jun. 1987). 43a–43g. Association for Computing Machinery, ACM Press No. 089791–232–2/87/0006/0043a (1987).

Onorato, L. and R. Schvaneveldt. "Programmer–Nonprogrammer Differences in Specifying Procedures to People and Computers." *Journal of Systems and Software 7* (1987): 357–369.

Humphrey, S.M. and N.E. Miller. "Knowledge–Based Indexing of the Medical Literature. The Indexing Aided Project." *Journal of the American Society for Information Science* 38 (3): 184–196 (1987).

Brajnik, G. et al. "User Modeling in Intelligent Information Retrieval." *Information Processing & Management* 23 (4): 305–320 (Feb. 1987).

Chiaramella, Y. and B. Defude. "A Prototype of an Intelligent System for Information Retrieval: IOTA." *Information Processing & Management* 23 (4) 285–3093 (1987).

Shwartz, P. "Natural Language Interfaces and Database Expert Systems." *Proceedings of the Annual Artificial Intelligence & Advanced Computer Technology Conference,* Long Beach, CA (Apr. 1987).

Scown, S.J. "Artificial Intelligence and Natural Language Systems—Part 2." *Computers and People (May–Jun.)*:16–22 (1987).

Fagan, J.L. "Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non–Syntactic Methods." Abstract and selected pages from chapters 1, 3 and 4, Sep. 1987. Also enclosed, Ph.D. Thesis, entire text version, Cornell University, No. 87–868, Sep. 1987.

Gevarter, W. "Introduction to Artificial Intelligence—Someday, Machines Will Be Able to Learn Throughout Their Lifetime: Building Up the Knowledge Base Needed for Advance Reasoning." *Chemical Engineering Progress* (Sep. 1987): 21–37.

Strukhoff, R. "The User Interface: How Does Your Product Look and Feel?" *Electronic and Optical Publishing Review* 7(4): 170–174 (1987).

Florian, D. "SAFIR (Smart Assistant for Information Retrieval)—An Artificial Intelligence Impact on Information Retrieval." Proceedings of the 11th International Online Information Meeting. *Online Information 87* (Dec. 1987), pp. 423–429. Medford, NJ: Learned Information, 1987.

Hendrix, G.G. and B.A. Walters. "The Intelligent Assistant Technical Considerations Invovled in Designing Q and A's Natural Language Interface." *Byte* (Dec. 1987): 251–258.

Schmidhausler F.J. In German: "Von Eliza uber Alice zum Sachmittel–Artificial Intellegence. (From ELIZA, via ALICE to practical use)." *Hard and Soft* (Nov./Dec. 1988): 52–55.

Kipfer, B. and F. Kipfer. In German: "Kunstliche Intelligenz in der Praxis [4] (Artificial Intelligence—Practice Aspects 4)" *Mikro+Kleincomputer 88–6* (1988): 47–55.

Wade, S. et al. "A Comparison of Knowledge–Based and Statistically–Based Techniques for Reference Retrieval." *Online Review* 12 (2): 91–108 (1988).

Katz, B. "Using English for Indexing and Retrieving." Conference Paper. In *User–Oriented Content–Based Text and Image Handling*, 1:314–333. Paris: C.I.D., 1988.

Katz, B. and Levin. "Exploiting Lexical Regularities in Designing Natural Language Systems" (Mar. 1988):316–323.

Katz, B. and B. Levin. "Exploiting Lexical Regularities in Designing Natural Language Systems." Massachusetts Institute of Technology, Artificial Intelligence Laboratory, *A.I. Memo No. 1041* (Apr. 1988): 1–23.

Ford, N. "European Research Letter: Knowledge–Based Information Retrieval." *Journal of the American Society for Information Science* 42 (1):72–74 (1991).

Katz, B. "Using English for Indexing and Retrieving." A.I. Memo No. 1096, (Oct. 1988): 1–30. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, 1988.

Salton, G. and M. Smith. "On the Application of Syntactic Methodologies in Automatic Text Analysis." Association for Computing Machinery, ACM Press, 137–150 (1989). ACM 0–89791–321–3/89/0006/0137.

Lancaster, F.W. et al. "Subject Analysis." *Annual Review of Information Science and Technology* (*ARIST*), ed. M.E. Williams, 24: 35–85. Amsterdam: Elsevier Science Publishers B.V., 1989.

Calzolari, N. and E. Picchi, "Acquisition of Semantic Information from an On–Line Dictionary." (1989): 87–93. Obtained through The British Library Document Supply Centre, Boston Spa, Wetherby, West Yorkshire LS23 7BQ, UK.

Wu, X. et al. "KDA: A Knowledge–Based Database Assistant." *IEEE* (1989): 402–409. IEEE publication No. CH2695–5/89/0000/0402.

Shuman, B.A. "Expert Systems and the Future of Interactive Searching." Proceedings of *National Online Meeting* (May 1989): 405–411. Medford, NJ: Learned Information, 1989.

Lelu, A. "From Data Analysis to Neural Networks: New Prospects for Efficient Databases." *Journal of Information Science* 17 (1991): 1–12.

Katz, B. "Using English for Indexing and Retrieving." Chapter 6 in *Artificial Intelligence at MIT: Expanding Frontiers*, vol. 1, ed. P.H. Winston and S.A. Sheared, 135–163. Cambridge, MA: The MIT Press, 1990.

Rapp and Wettler In German with English summary (p. 27). "Simulation der Suchwortgenerierung im Information–Retrieval durch propagierung in einem konnektionistischen Wortnetz." Nachr. Dok. 41 (1990): 27–32.

Aoe J.–I. "A Compendium of Key Search References." *Sigir Forum* 24 (3):26–42, 1990.

Walker, A. "Expert Database Systems in Prolog." Chapter 4 in *Knowledge Systems and Prolog*, Second Edition, ed., Walker et al, 248–336. Reading, MA: Addison–Wesley, 1990.

Libby, E. DuRoss. "Anaphora in Natural Language Processing and Information Retrieval." *Information Processing & Management* 26 (1): 39–52 (1990).

Lirov, Y. and V. Lirov. "Online Search Plus Logic Programming Equals Subject Bibliography: An Expert Systems Approach to Bibliographic Processing." *Online Review* 14 (1): 3–12 (1990).

McAllister–Harper, D. "Dewey Decimal Classification in the Online Environment: A Study of Libraries in North Carolina." *Cataloging & Classification Quarterly* 11 (1): 45–58 (1990).

Molholt, P. "MARC and the Promise of Artificial Intelligence for Subject Access: Current Limitations and Future Considerations." in *Beyond the Book Extending MARC for Subject Access*, ed. T. Petersen, and P. Molholt, 247–257. Boston: G.K. Hall & Co., 1990).

Monroe, A.N. and K.D. Roe. "What's the Purpose? Functional Access to Archival Records." *Beyond the Book Extending MARC for Subject Access*, ed. T. Petersen, and P. Molholt, 157–170. Boston: G.K. Hall & Co., 1990).

Svenonius, E. "Bibliographical Control." In *Academic Librarians Research Perspectives*, ed. M.J. Lynch, ACRL Publications in Librarianship No. 47: 38–66. Chicago: American Library Association, 1990.

Sano, H. "Facet Tabulation of Index Terms." *Information Processing & Management* 26 (4): 543–248 (1990).

Reynolds, S.J. "Sabbatical: The Pause That Refreshes." *Journal of Academic Librarianship*, 16 (2): 90–93 (1990).

Comaromi, J.P. "Summation of Classification as an Enhancement of Intellectual Access to Information in an Online Environment." *Cataloging & Classification Quarterly* vol. 11(1) 99–102 (1990).

Line, M.B. "Bibliographic Records for Users: From Disordered Superabundance to Cost–Effective Satisfaction." *ASLIB Proceedings* 42 (2): 41–49 (1990).

Fuhr, N. In German with English summary (p. 3). "Zur ubervindung der Diskrepanz zwischen Retrievalforschung und—praxis." *Nachr. Dok*, 41 (1990): 3–7.

Whalen, T.E. and A.S. Patrick. "COMODA: A Conversation Model for Database Access." *Behavior & Information Technology* 9 (2): 93–110 (1990).

Hahn, U. "Topic Parsing: Accounting for Text Macro Structures in Full–Text Analysis." *Information Processing & Management* 26 (1): 135–170 (1990).

Salton, G. et al. "On Application of Syntactic Methodologies in Automatic Text Analysis." *Information Processing & Management* 26 (1): 73–92 (1990).

Berwick, R. and S. Fong. "Principle--Based Parsing: Natural Language Processing for the 1990s." Chapter 12 in *Artificial Intelligence at MIT: Expanding Frontiers,* ed. P.H. Winston and S.A. Shellard, 288–361. Cambridge, MA: The MIT Press, 1990.

Rowley, J.E. "A Comparison Between Free Language and Controlled Language Indexing and Searching." *Information Services & Use* 10 (1990): 147–155.

Black, B. "Computers and Everyday Language." *Computer Age* (Mar. 1990): 46–48.

Korycinski, C. and A.F. Newell. "Natural--Language Processing and Automatic Indexing." *The Indexer* 17 (1): 21–29 (1990).

Casadaban, C. "DEB: A Diagnostic Experience Browser Using Similarity Networks." Conference Paper, *Fifth Conference on Artificial Intelligence for Space Applications,* Huntsville, AL, May 1990: 517–524.

Kerner, J. and R.S. Freedman. "Developing Intelligent Tutoring Systems with A Hypermedia Object–Based Intelligent Educator (HOBIE)." *Proceedings: The Third International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems* (Jul. 1990), 2: 890–897. ACM Publication No. ACM 089791–372–8/90/0007/0890.

Barnett, J. et al. "Knowledge and Natural Language Processing." *Communications of the ACM* 33 (8): 52–71 (Aug. 1990).

Haton, J.P. In French with English abstract (p. 198). "Introduction a l'intelligence artificielle et aux systemes experts." *Memoires et Etudes Scientifique Revue de Metallurgie* (Oct. 1990): 197–208.

Myers, A. et al. "The Facilitation of Problem--Based Learning in Medical Education Through a Computer–Mediated Tutorial Library." Fourteenth Annual Symposium on Computer Applications in Medical Care, Washington, D.C., Nov. 1990: 469–472. IEEE Computer Society Press, Publication No. 0195--4210/90/0000/4069, 1990.

Koschmann, T.D. et al. "Designing Communication Protocols for a Computer–Mediated Laboratory for Problem--Based Learning" Fourteenth Annual Symposium on Computer Applications in Medical Care, Washington, D.C., Nov. 1990: 464–468. IEEE Computer Society Press, Publication No. 0195--4210/90/0000/0464, 1990.

Bergeron, B.P. et al. "A Generic Neural Network--Based Tutorial Supervisor for Computer Aided Instruction." *Fourteenth Annual Symposium on Computer Applications in Medical Care,* Washington, D.C. (Nov. 1990): 435–439 IEEE Computer Society Press, Publication No. 0195–4210/90/0000/0435.

Hancock--Beaulieu, M. "Evaluating the Impact of an Online Library Catalogue on Subject Searching Behaviour at the Catalogue and at the Shelves." *Journal of Documentation* 46 (4): 318–338 (1990).

Winston, P.H. and B.K.P. Horn. "Symbolic Pattern Matching," Chapter 24: 354–365. "Interpreting Transition Trees," Chapter 28: 412–430. "Compiling Transition Trees," Chapter 29: 432–439. "Procedures–Writing Programs and Database Interfaces," Chapter 30: pp. 442–454, in *LISP Third Edition,* ed. P.H. Winston and B.K.P. Horn. Reading, MA: Addison–Wesley Publishing Co., 1991.

Cha, S.K. "Kaleidoscope: A Cooperative Menu–Guided Query Interface (SQL Version)." *IEEE Transactions on Knowledge and Data Engineering,* 3 (1): 42–47 (1991).

Martin, M.M. "Subject Indexing in the New Ethnographic Bibliography of North America." *Behavioral & Social Sciences Librarian* 11 (1): 13–26 (1991).

Koutamanis, A. and V. Mitossi. "Design Information Retrieval." *Delft Progress Report,* 15 (1991/1992): 73–86.

Bell, J.E. and L.A. Rowe. "An Exploratory Study of AD Hoc Query Languages to Databases." *Eighth International Conference on Data Engineering,* Tempe, Arizona (Feb. 1992): 606–613. IEEE Computer Society Press, 1992. Publication No. 0–81686–2545–7/92.

Zijlstra, B.H.A. "Search Strategies in Natural Language: Bringing Down the Barriers to Information Transfer." *AGARD Conference Proceedings 505* (Oct. 1991): 4–1–4–39. Advisory Group for Aerospace Research and Development, AGARD Publication No. CP–505, 1991.

Germain, E. "Introducing Natural Language Processing." *A.I. Expert 7* (8): 30 (1992).

Bliek, A. et al. "NLTIS: Objectives and Benefits." *Delft Progress Report* 15 (1991/1992): 291–297.

Bliek, A. et al. "Overview of the NLTIS." *Delft Progress Report* 15(1991/1992): 299–319.

Bliek, A. et al. "The Object and Entity Model Manager." *Delft Progress Report* 15 (1991/1992): 321–334.

Bliek, A et al. "The Dialogue Manager." *Delft Progress Report* vol. 15 (1991/1992): 335–336.

Bliek, A et al. "Expert Systems for Diagnosis." *Delft Progress Report* 15 (1991/1992): 337–350.

Smolen, A. et al. "Development of the Cardiovascular System: An Interactive Video Computer Program." *Proceedings from Sixteenth Annual Symposium on Computer Applications in Medical Care,* Baltimore, Maryland (Nov. 1992): 847–848. American Medical Informatics Association, AMIA Publication No. 0195–4210/92.

O'Brien, L. "It's Only Logical: The Combination of Declarative Procedural Programming Makes Prolog a Logical Choice." *Computer Language* 9 (12): 61 (1992).

Sheilla E. Desert. "Westlaw Is Natural v. Boolean Searching: A Performance Study." *Law Library Journal,* 85 (1993): 713.

Konig, A. "Data structure and computer program for fast interactive searches of clinical diagnoses." *Med. Inform.* 18 (1) 45–51 (1993).

Pritchard–Schoch, T. "Natural Language Comes of Age." Westlaw Online Information Service, Online 33 (May 1, 1993): 1–15. Westlaw No. 3280292.

Keisch, B. "Interactive Computer–Based Instruction: Basic Material Control and Accounting Demonstration." Brookhaven National Laboratory, Upton, NY, Aug. 1993. NTIS: DE93018669.

Paradaens, J. et al. "Toward a Theory of Spatial Database Queries" Association of Computing Machinery, 1994, pp. 279–288. ACM Publication No. 0–89791–639–5/94/0005.

Feldman, S.E. "Search Natural Language Systems: Searchers Know Thy Engine." *Searcher* (Oct. 1994): 34–38.

DuVal, R.W. "Rotorcraft Handling Qualities and Flight Control System Specification Personal Computer Tutorial Data Base." *Final Report,* (Feb. 1995): 1–32. Advanced Rotorcraft Technology, Inc., 1995, NTIS: ADA308217.

Yuret, D. "Discovery of Linguistic Relations Using Lexical Attraction." Ph. D. diss., Massachusetts Institute of Technology, 1998, pp. 1–56.

Katz, B. et al. "Philip: Integrating Web Resources and Lexicons into Natural Language Query Systems." *IEEE* (Sep. 1999): 255–261. IEEE Publication No. 0–7695–0253–9/99.

Vaccari, E. et al. "DARR: A Free–text Analysis System for the Automatic Documentation of Radiological Reports." *Meth. Inform. Med.* 16(3): 144–153 (1977).

Starr, B.H. et al. "HIKE (HPKB Integrated Knowledge Environment)—A Query Interface and Integrated Knowledge Environment of HPKB." *Intelligent Systems Demos: Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 927–928. Menlo Park, CA: AAAI Press, 1999. Publication No. AAAI–99.

Katz et al. Pages from website re the START system including overview and rules; most pages by Boris Katz. http://www.ai.mit.edu/people/boris/webaccess/node1.html through /node10.html (Feb. 1997). Undated material from the MIT InfoLab Group http://www.ai.mit.edu/projects/infolab.

Dillon, M. and A.S. Gray. "FASIT: A Fully Automatic Syntactically Based Indexing System." *Journal of the American Society for Information Science.* 34 (2): 99–108 (1983).

Freedman. "Using a Text Planner to Model the Behavior of Human Tutors in an ITS." Printed on Oct. 12, 1999 from http://www.cs.indiana.edu/event/maics96/Proceedings/Freedman/freedman.html.

Jansen. "Using an Intelligent Agent to Enhance Search Engine Performance." Printed on Oct. 12, 1999 from http://www.firstmonday.dk/issues/ from vol. 2, No. 3, 1997.

Sneiders. "FAQ Answering on WWW using Shallow Language Understanding." Printed on Oct. 12, 1999 from http://ekd.dsv.su.se/ifip98/faq–answering.htm.

Whitehead. "List of Publications from the DBLP Bibliography Server." Printed on Oct. 12, 1999 from http://www.informatik.uni–trier.de/–ley/db/indices/a–tree/w/Whitehead:Steven__D=.html.

Spoken Language Systems. "A Conventional System Architecture: Galaxy." (1998) Printed on Oct. 12, 1999) from http://www.sls.lcs.mit.edu/sls/whatwedo/architecture.html.

Katz. "START Natural Language Question Answering System" printed on Oct. 12, 1999 from http://www.ai.mit.edu/projects/infolab/start–system.html.

"Computer Networks and ISDN Systems." Printed from http://www.elsevier.com/inca/publications/store/6/0/0/0/7/9/600079.pub.htm.

* cited by examiner

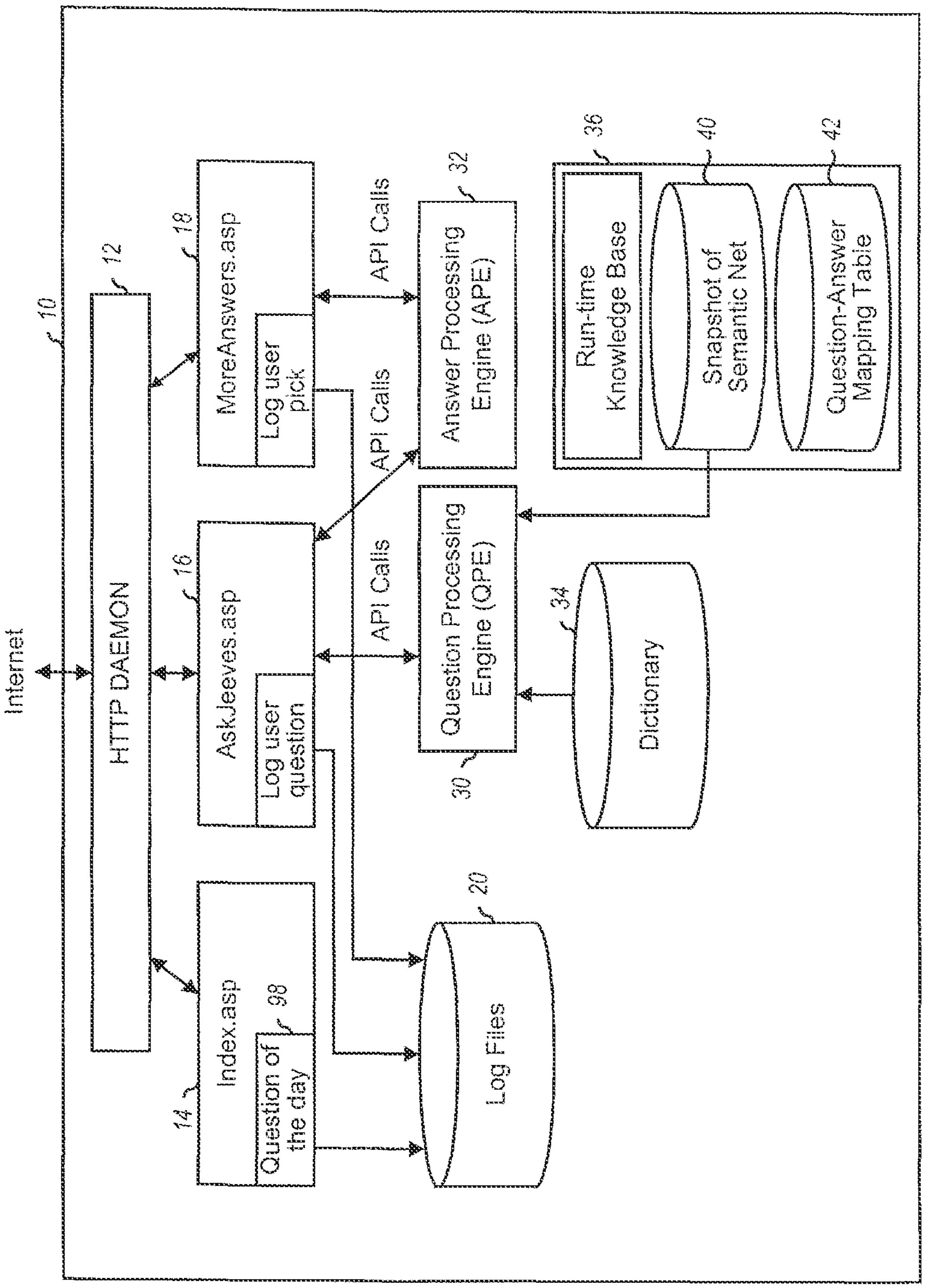

Internet
10
HTTP DAEMON
12
Index.asp
14
Question of the day
98
AskJeeves.asp
16
Log user question
MoreAnswers.asp
18
Log user pick
API Calls
API Calls
API Calls
Question Processing Engine (QPE)
30
Answer Processing Engine (APE)
32
Dictionary
34
Log Files
20
Run-time Knowledge Base
36
Snapshot of Semantic Net
40
Question-Answer Mapping Table
42

FIG. 1(a)

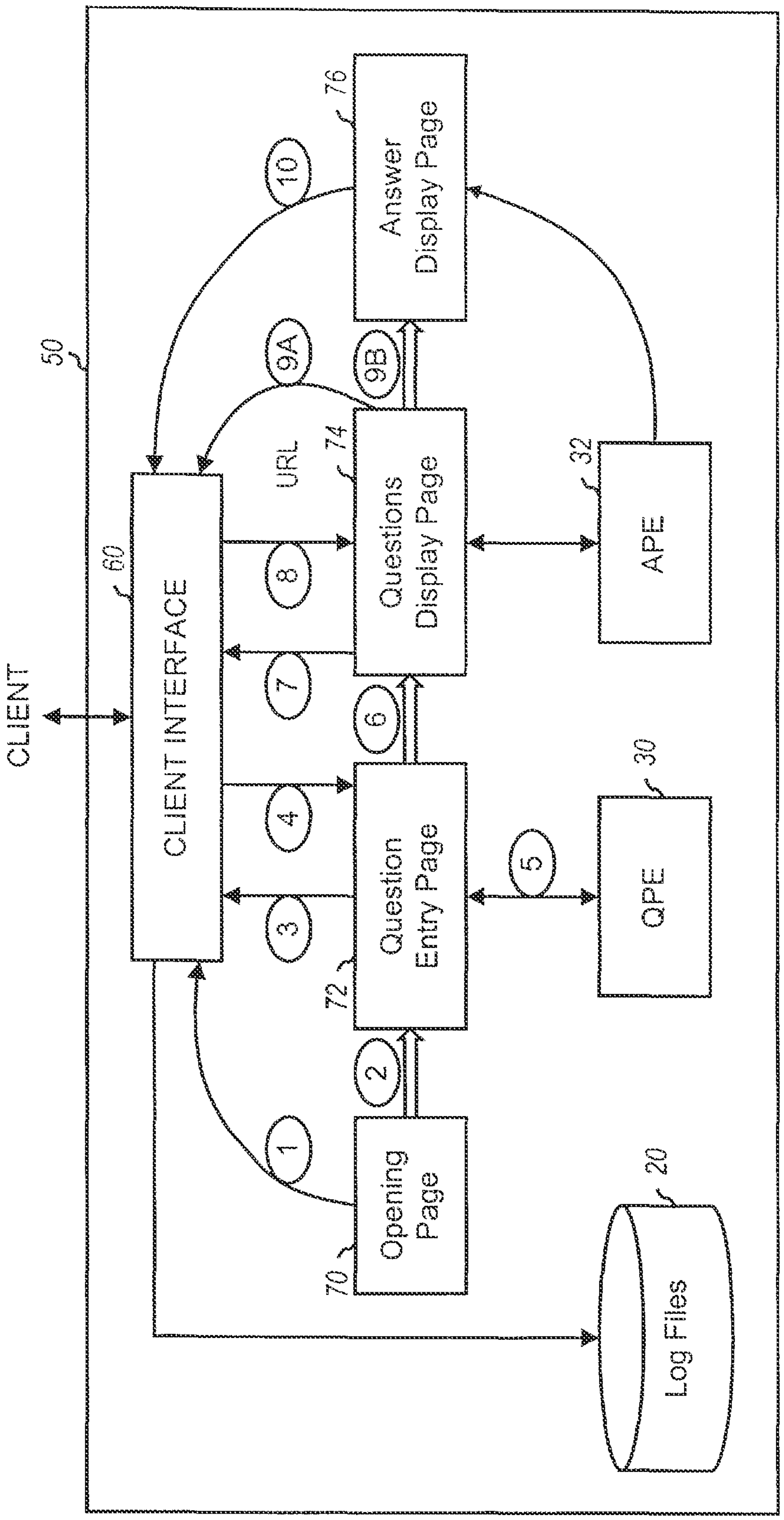
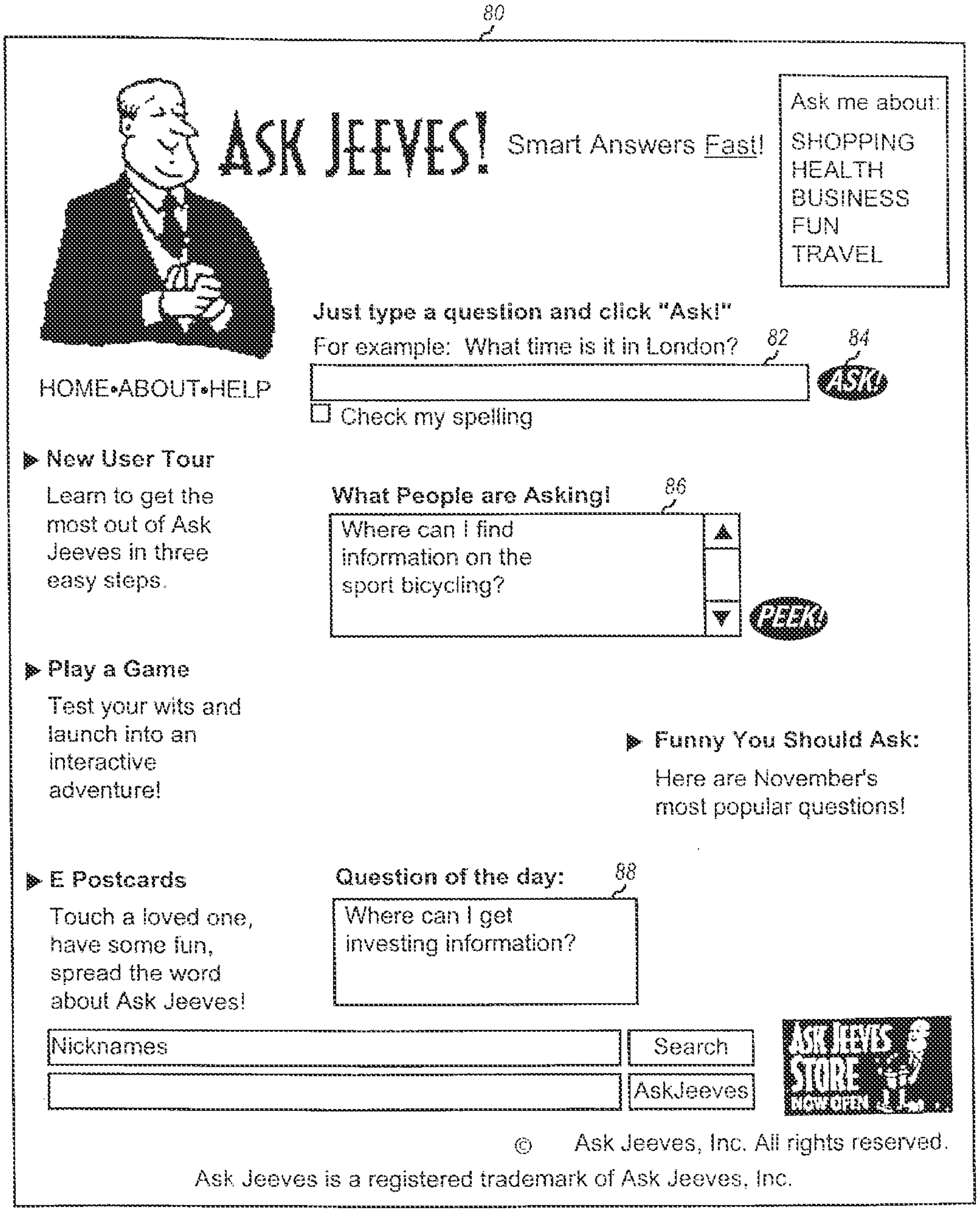

CLIENT
50
CLIENT INTERFACE
60
Opening Page
70
Question Entry Page
72
Questions Display Page
74
Answer Display Page
76
URL
1
2
3
4
5
6
7
8
9A
9B
10
QPE
30
APE
32
Log Files
20

FIG. 1(b)



Smart Answers Fast!

Ask me about:

SHOPPING
HEALTH
BUSINESS
FUN
TRAVEL

Just type a question and click "Ask!"

For example: What time is it in London?

□ Check my spelling

▶ New User Tour

Learn to get the most out of Ask Jeeves in three easy steps.

What People are Asking!

Where can I find information on the sport bicycling?

▶ Play a Game

Test your wits and launch into an interactive adventure!

▶ Funny You Should Ask:

Here are November's most popular questions!

▶ E Postcards

Touch a loved one, have some fun, spread the word about Ask Jeeves!

Question of the day:

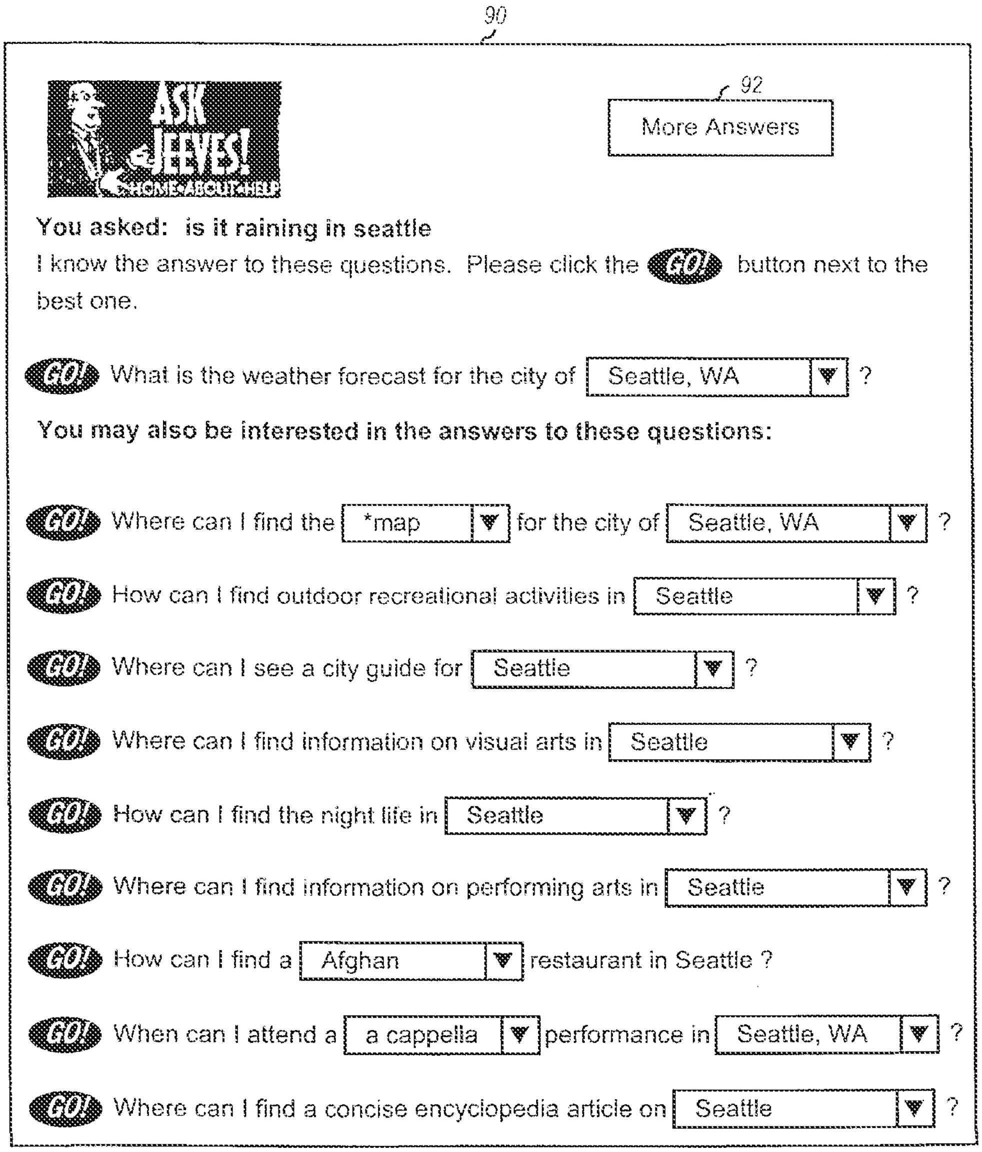
Where can I get investing information?

Nicknames

© Ask Jeeves, Inc. All rights reserved.

Ask Jeeves is a registered trademark of Ask Jeeves, Inc.

FIG. 2



You asked: is it raining in seattle

I know the answer to these questions. Please click the GO! button next to the best one.

GO! What is the weather forecast for the city of Seattle, WA ▼ ?

You may also be interested in the answers to these questions:

GO! Where can I find the *map ▼ for the city of Seattle, WA ▼ ?

GO! How can I find outdoor recreational activities in Seattle ▼ ?

GO! Where can I see a city guide for Seattle ▼ ?

GO! Where can I find information on visual arts in Seattle ▼ ?

GO! How can I find the night life in Seattle ▼ ?

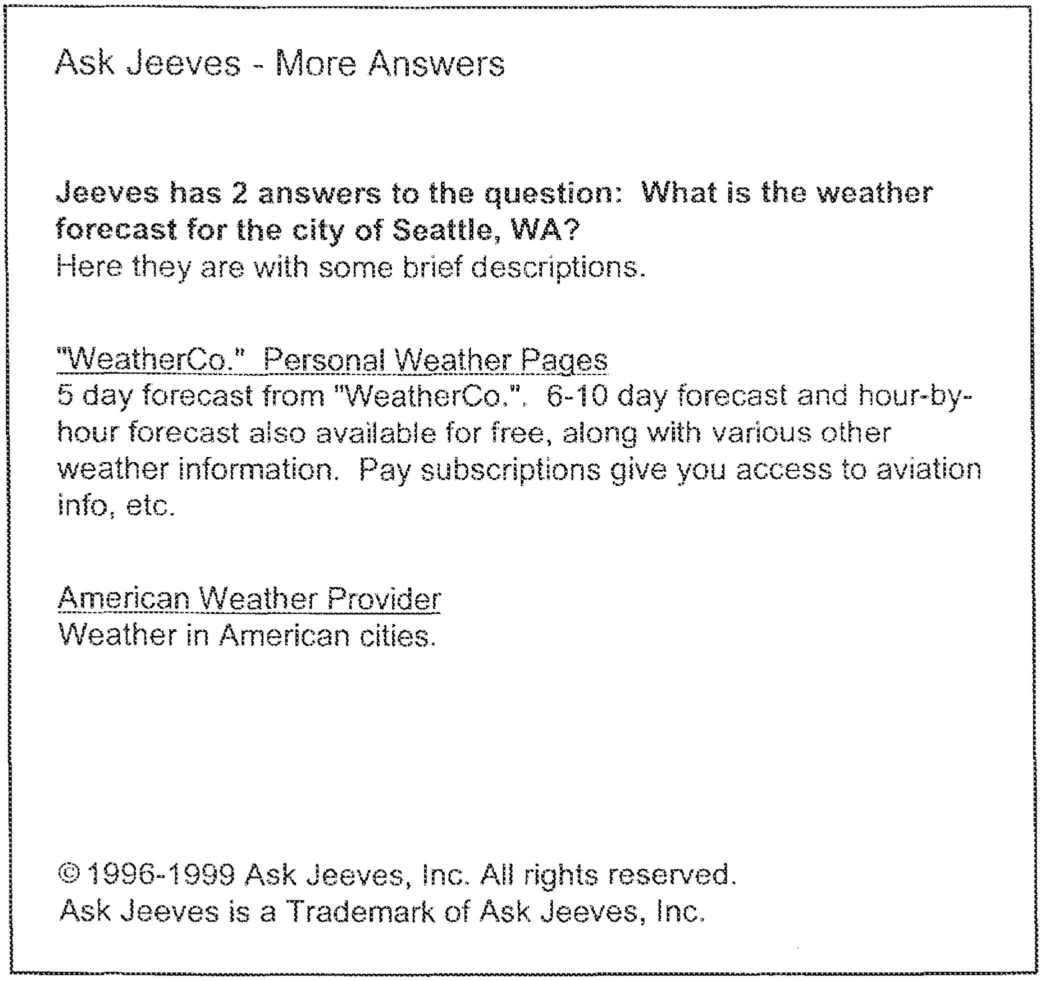
GO! Where can I find information on performing arts in Seattle ▼ ?

GO! How can I find a Afghan ▼ restaurant in Seattle ?

GO! When can I attend a a cappella ▼ performance in Seattle, WA ▼ ?

GO! Where can I find a concise encyclopedia article on Seattle ▼ ?

FIG. 3



Ask Jeeves - More Answers

Jeeves has 2 answers to the question: What is the weather forecast for the city of Seattle, WA?
Here they are with some brief descriptions.

"WeatherCo." Personal Weather Pages
5 day forecast from "WeatherCo.". 6-10 day forecast and hour-by-hour forecast also available for free, along with various other weather information. Pay subscriptions give you access to aviation info, etc.

American Weather Provider
Weather in American cities.

© 1996-1999 Ask Jeeves, Inc. All rights reserved.
Ask Jeeves is a Trademark of Ask Jeeves, Inc.

FIG. 4

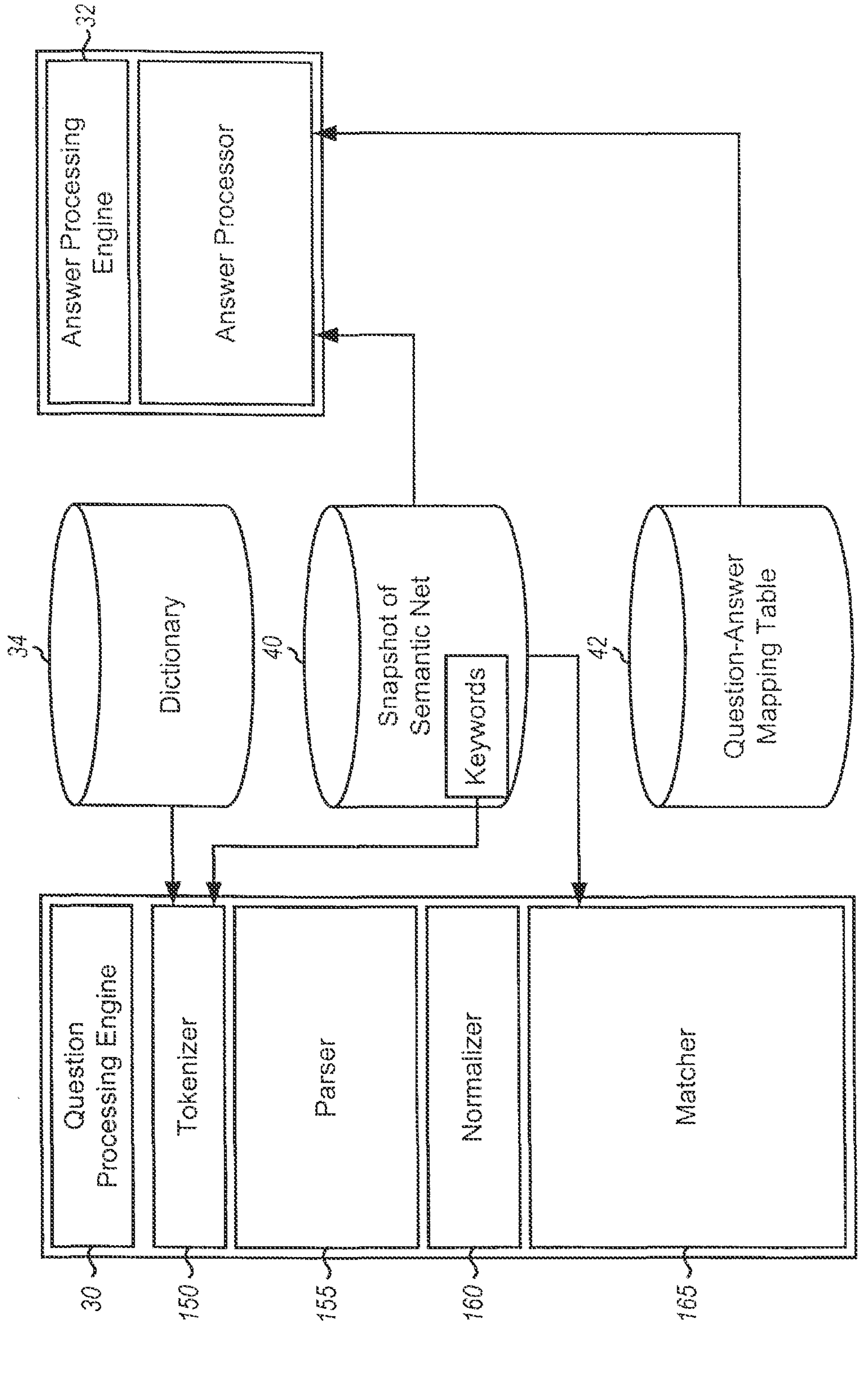

32
Answer Processing Engine
Answer Processor
34
Dictionary
40
Snapshot of Semantic Net
Keywords
42
Question-Answer Mapping Table
Question Processing Engine
Tokenizer
Parser
Normalizer
Matcher
30
150
155
160
165

FIG. 5

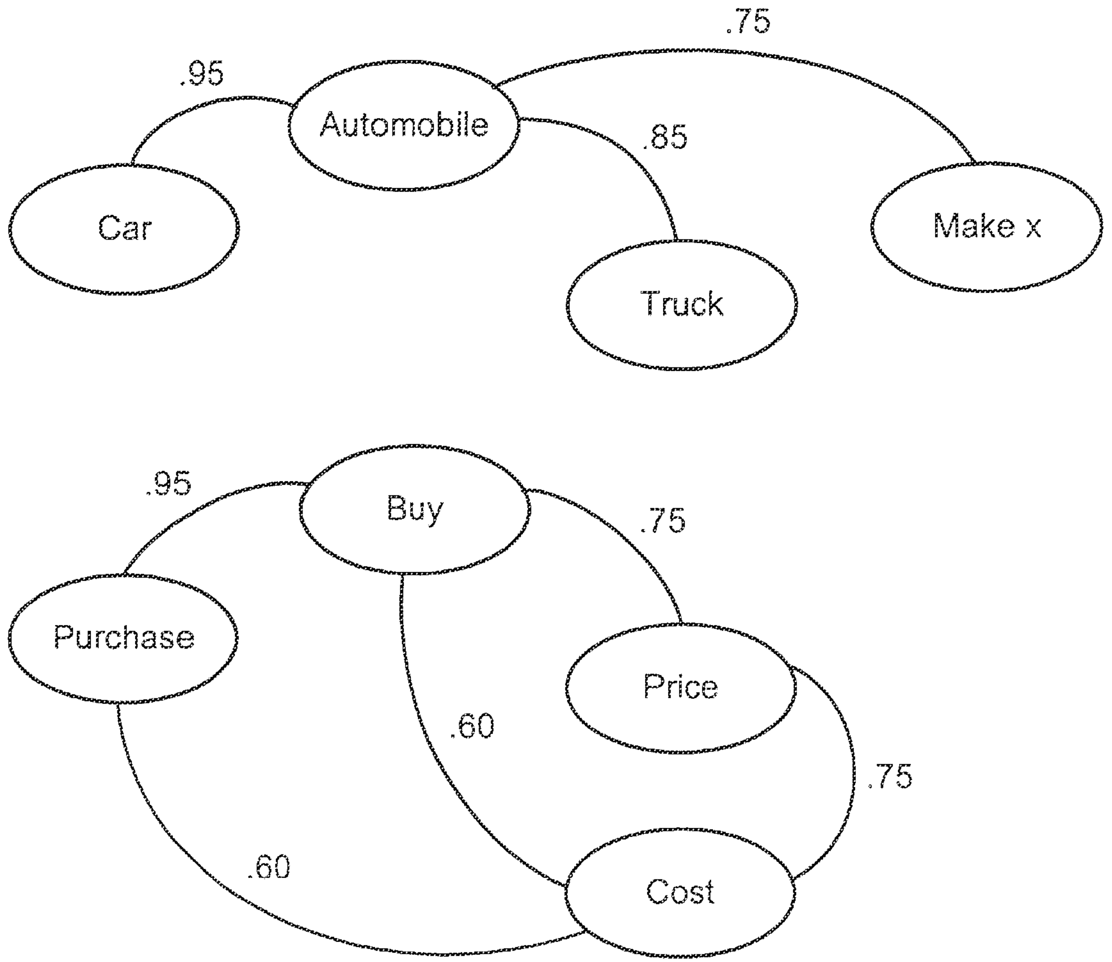

.95
Automobile
.75
.85
Car
Make x
Truck
.95
Buy
.75
Purchase
Price
.60
.75
.60
Cost

FIG. 6

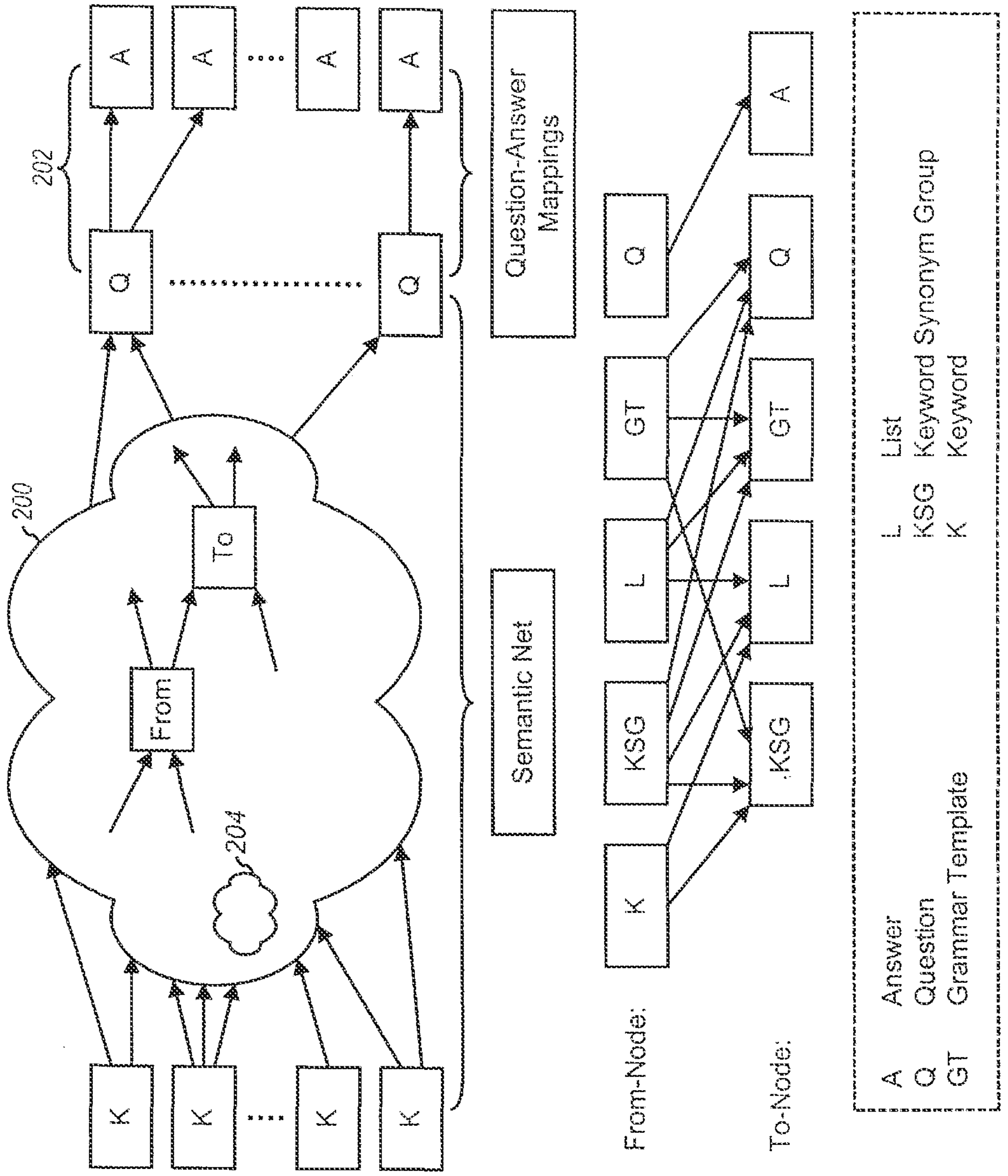

200
202
204
K
K
K
K
From
To
Q
Q
A
A
A
A
Semantic Net
Question-Answer Mappings
From-Node:
To-Node:
K
KSG
L
GT
Q
KSG
L
GT
Q
A
A Answer
Q Question
GT Grammar Template
L List
KSG Keyword Synonym Group
K Keyword

FIG. 7

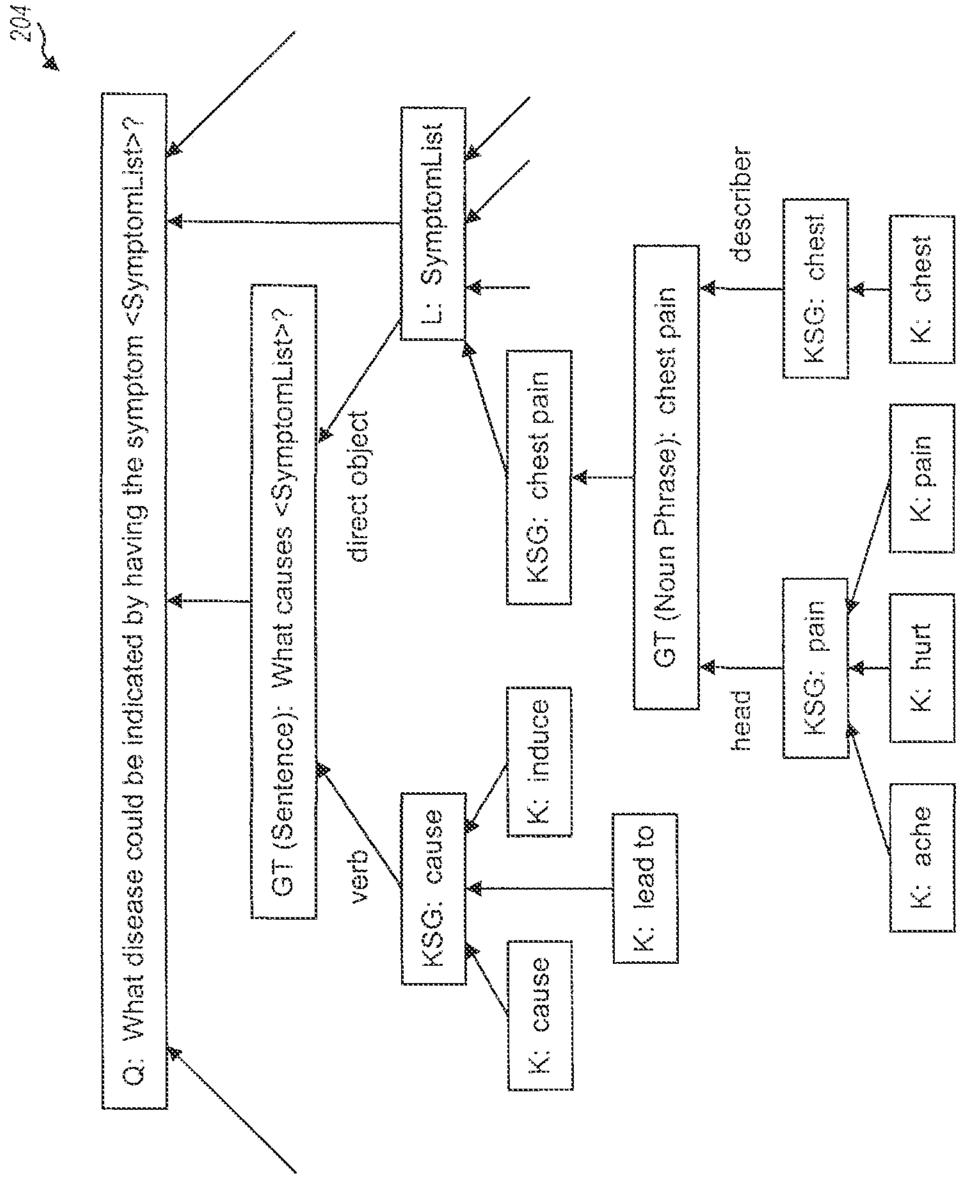

204
Q: What disease could be indicated by having the symptom <SymptomList>?
GT (Sentence): What causes <SymptomList>?
L: SymptomList
direct object
verb
KSG: cause
K: cause
K: induce
K: lead to
KSG: chest pain
GT (Noun Phrase): chest pain
describer
head
KSG: chest
K: chest
KSG: pain
K: ache
K: hurt
K: pain

FIG. 8

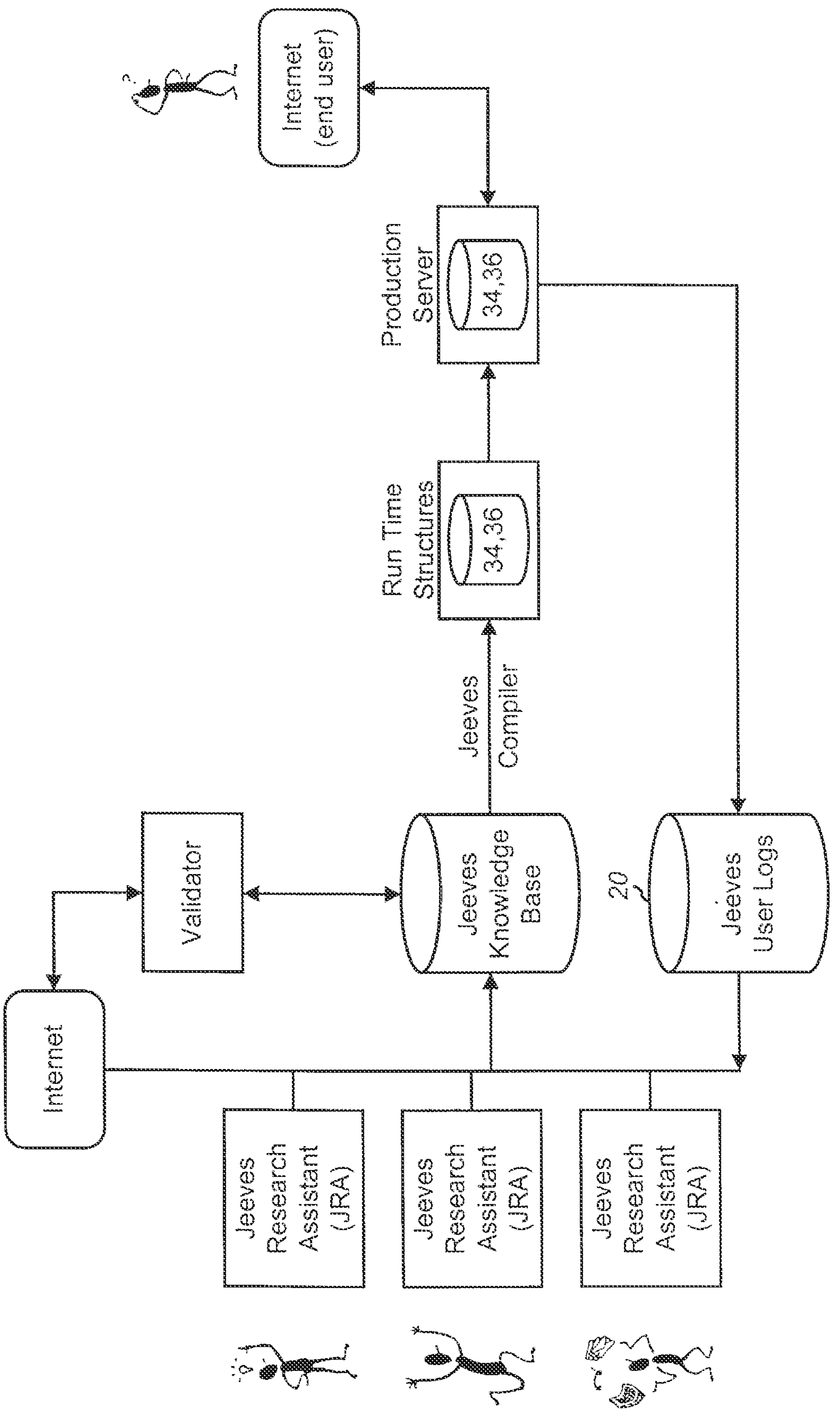

Internet
Validator
Internet
(end user)
Production
Server
34,36
Run Time
Structures
34,36
Jeeves
Compiler
Jeeves
Knowledge
Base
20
Jeeves
User Logs
Jeeves
Research
Assistant
(JRA)
Jeeves
Research
Assistant
(JRA)
Jeeves
Research
Assistant
(JRA)

FIG. 9

GRAMMAR TEMPLATE QUERY SYSTEM

BACKGROUND OF THE INVENTION

The present invention relates to information query systems in general and to interactive query systems more particularly.

With the increasing popularity of the Internet, the global internetwork of networks, many services have come into being that seek to provide information for users of those services. One such service is the GOPHER service, wherein a user is presented with lists of information in the form of menus, and the user is provided with options to select a menu item, which might lead to another menu or to a document of interest, or to initiate a search with a particular set of keywords. The GOPHER system was primarily text based, with the underlying structure being directories on file servers, usually spanning many interlinked and independent servers.

With the advent of a more graphical interface, many such search services continue today, albeit with a graphical interface for use with a Web browser. Here, "Web" refers to the hyperlinked collection of dynamic and static hypertext pages available over the Internet using the HTTP (HyperText Transport Protocol) and commonly referred to as the "World Wide Web" or "WWW", and a Web browser is a client program which allows a user to navigate the Web.

A typical navigation involves setting up an initial query with a set of search terms and viewing the results. If the results are provided as a hypertext page, the user can then select a link on that hypertext page to view the results in more detail.

A disadvantage of searching using search terms is that the English language is imprecise without context and computers are not good at context. For example, asking a human librarian about "freedom and values in the Victorian era" might yield a book of essays on what the social mores were in the late nineteenth century in Europe and North America. However, performing a search with a search engine against a database of documents might result in the computer returning listings of real estate values for homes with Victorian architecture in Freedom, Calif.

Many providers of search services have attempted to automate the process of determining the proper context of a query so that the correct meaning is ascribed to each term. For example, a server might be programmed to note the ambiguity in the term "Victorian" and to further note that "era" refers to a time period and therefore, "Victorian" should be interpreted as the time period. Such processing is quite complex and often still fails to understand the context in which the user asks the question.

SUMMARY OF THE INVENTION

One embodiment of an information server according to the present invention directs users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input. The information server includes a query input processor, a question processor and an answer processor. The query input processor is used for accepting an initial user query. The question processor processes the initial user query to identify a set of possible well-formed questions selected from the question database, where a well-formed question is a question in the database that is coupled to at least one answer reference. The answer reference is typically either an answer or a pointer to a possible location of an answer.

In a specific embodiment, the information server is coupled to the Internet so that users can pose questions using a Web browser from any Internet-connected device. In some systems, the question processor includes a tokenizer for tokenizing the initial user query into a list of words, a parser for generating a syntactic structure from the list of words, a normalizer for reducing the syntactic structure to a canonical syntactic structure, and a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates two variations of an information server according to the present invention;

FIG. 1(*a*) is a block diagram of an HTTP server used to implement the information server and

FIG. 1(*b*) is a block diagram of a generalized server user to implement the information server.

FIG. 2 is an illustration of one possible display for a question entry page.

FIG. 3 is an illustration of one possible display for a question display page; FIG. 3 is an example of a question display page that might result from the processing of the question entry page shown in FIG. 2.

FIG. 4 is an illustration of one possible display for an additional answers page; FIG. 4 is an example of an additional answers page that might result from the processing of the question display page shown in FIG. 3.

FIG. 5 is a block diagram of the QPE and APE of FIG. 1, shown here in further detail.

FIG. 6 is a graph of a portion of a semantic net.

FIG. 7 is an alternate view of a semantic net.

FIG. 8 is a more detailed view of a portion of the semantic net shown in FIG. 7.

FIG. 9 is a block diagram of a computer system for collecting information and generating run-time structures, such as semantic nets, from the collected information.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

The following description describes some embodiments of the invention and these examples are not intended to limit the scope of the invention, which is defined by the attached claims. For example, the embodiments shown in the figure assume that the user asks questions of an information server via a Web browser over an Internet connection, but that need not be the case, as the user might access the information server over a network other than the Internet, or might access a local information server without an intervening network.

FIG. 1(*a*) is a block diagram of one embodiment of an information server 10 that is Internet-based. Information server 10 is an HTTP server, which responds to requests from HTTP clients such as Web browsers. The actual architecture of an Internet HTTP client-server link and HTTP client is not shown, but it should be understood that information server 10 operates on a computer that is designed and configured for Internet traffic, specifically TCP/IP packets encoding HTTP messages and it should be understood that any manner of client computer can be used to operate the Web browser. Since that technology is well-known and readily available, it need not be described further herein.

Information server **10** can either be a dedicated computer, a computing device specifically designed to implement the functions of an information server according to the methods described herein, or information server **10** can be in the form of one or more program code modules designed to run on a general purpose HTTP server as a process of that HTTP server. However implemented, information server **10** usually comprises the components shown in FIG. 1, such as an HTTP daemon **12** and a basic set of active HTML pages, including an opening page **14**, a question page **16** and an additional answers page **18**.

Opening page **14** comprises code and text for presenting the user at an HTTP client (such as a Web browser) with an introduction to information server **10**. Question page **16** comprises code and text for presenting the user at the HTTP client with a question entry page (such as the page shown in FIG. 2) and a question display page (such as the page shown in FIG. 3). Additional answers page **18** comprises code and text for presenting the user at an HTTP client with additional answers to the questions shown in a question display page. The basic pages are Active Server Page (.asp) programs, but other programming languages could be used. Active Server Pages are one known technique for combining Visual Basic (VB) scripts and HTTP data structures and text. In addition to the basic set of pages, other pages might be provided to HTTP clients, depending on the navigation taken by the user through pages available from information server **10**.

As shown in FIG. 1(*a*), actions taken by users in response to prompts on the basic set of pages are logged in log files **20**. Also, as explained below, question page **16** receives questions and presents the questions entered by users to a question processing engine (QPE) **30**. In this example, the communication with QPE **30** is via application programming interface (API) calls. As is well-known in the art of API design, the functionality of QPE **30**, although shown separately, might be implemented as a set of calls from code in question page **16** to program code compiled into question page **16** (as might be the case with an API link library) with the communication between the two being in the form of function calls, or the program code for implementing QPE **30** could be a separately running process with the communication being in the form of interprocess messages.

Question page **16** receives the data needed to construct a question display page from QPE **30** and from an answer processing engine (APE) **32**. Information server **10** also includes a dictionary **34** and a knowledge base **36**, which comprises storage for a semantic net snapshot **40** and a question-answer mapping table **42**. QPE **30** is coupled to dictionary **34** and semantic net snapshot **40** and uses the information obtained from those sources to generate template questions in response to a user-entered question. Template questions are questions that are mapped to answers in question-answer mapping table **42**. Template questions usually, but not always, include parameters, such as the template question "What is the weather like in <city>?" where "<city>" is a parameter. APE **32** is coupled to knowledge base **36** to obtain information from semantic net snapshot **40** and question-answer mapping table **42**.

FIG. 1(*b*) is a block diagram of a more generalized information server **50**. Information server **50** can be implemented in a variety of client-server systems. For example, information server **50** might be built into a kiosk for providing answers to kiosk users' questions, or information server **50** might be connected to a corporate network for answering customer, supplier and/or employee questions. Information server **50** is shown comprising a client interface **60** for providing various pages (**70**, **72**, **74**, **76**) to a client and interconnections between the various pages and QPE **30**/APE **32**.

The circled numbers in FIG. 1(*b*) indication one possible set of steps that might occur in a session with a user at a client. In step **1**, client interface **60** sends an opening page **70** to the client and control is passed to a question entry page **72** (step **2**). Client interface **60** presents the client with question entry page **72** (step **3**) and receives the user's response (step **4**). Information server **50** passes the response to QPE **30**, which returns a set of template questions (step **5**). Control passes to the code corresponding to a questions display page **74** (step **6**), and information server **50** generates the questions display page for presentation to the user. Client interface **60** presents the questions display page to the user (step **7**) and waits for the user to select a template question from the questions display page (step **8**). The user selects a template question, if more than one is presented, and also selects any necessary parameters. Usually, the parameters will be directly related to the question asked, so the desired parameters might already be selected.

Once the user selects a template question, information server **50** uses AE to generate answers to the questions and either presents the user with one or more URL's of sites that answer the initial question (step 9A) and control passes to an answer display page (step 9B) that presents the user with the answer directly (step **10**).

Referring now to FIG. **2**, a typical question entry page **80** is shown. When a user is presented with question entry page **80**, the user will then enter an initial user query, in an entry box **82**, and press a button **84** to begin the process. The user can also view questions being posed by other users in window **86**.

The initial user query can be a natural language question (e.g., "Where can I find information on the sport bicycling?") and may well include grammatical errors, or a set of keywords, such as "info sport bicycling", or a combination of a natural language question and keywords. Keywords might be "noun phrases" such as "King of Spain". When the user presses button **84**, the initial user query is sent to information server **50** and client interface **60** passes the query to QPE **30**. The query is logged to log files **20** for use in further refining information server **50**, as described later in this description.

Typically, the query is in the form of a text string, but in some implementations, the query might be a text string combined with nontext data structures such as a collection of radio button (multiple choice) selections, switches and/or other pointers to selections. It should be understood that many different forms of queries are possible, so long as QPE **30** accepts the form in which the queries are presented. From the initial user query, QPE **30** processes the question to identify a set of template questions. Template questions are questions in a form that allows its context to be easily understood from the question, either because information server **50** has already answered the question to the liking of one or more users, or because an information analyst has input a question-answer template for that question. A question-answer template is a data structure that codes for one or more questions and includes pointers to answers to the coded question. This is explained in more detail in the description below of semantic nets and question-answer mappings (see, for example, FIG. 7). An example of a question answer template is:

Is it raining in <city>? ==> Pointer to weather page with parameter <city>

In that example, queries for the weather in many cities is compressed into one question template, with the parameter <city> being a placeholder for the city of interest. In the template, the answer to the question is a pointer to a weather page (such as a page on a weather Web site) with a parameter equal to the city of interest. Thus, using this question-answer template, information server **50** knows how to answer the question "Is it raining in Seattle?", namely by directing the user to a weather server that accepts a city parameter and passing the weather server "Seattle" as the parameter.

Once QPE **30** identifies the set of template questions, those questions are presented to the user as questions display page **74** for selection. FIG. **3** shows an example display **90** resulting from such a questions display page. From that display **90**, the user can select the desired template question and parameters, or can select a button **92** for more answers, resulting in a display such as that shown in FIG. **4**.

An information server might also handle incidental displays, which are useful or interesting to users but are not necessarily required for the operation of the system. One example, a "question of the day" generator **98** is shown in FIG. **1**(*a*). Generator **98** might select a question from asked questions to display in a question of the day area **88** on question entry page **80** (see FIG. **2**). Other incidental displays might include a real-time question ticker **86** that puts up a random sampling of the questions currently being asked, a display of popular questions in specific categories and the like.

Referring now to FIG. **5**, a block diagram of QPE **30** and APE **32** is shown with QPE **30** comprising a tokenizer **150**, a parser **155**, a normalizer **160** and a matcher **165**. Tokenizer **150** converts the initial user query into a list of words and provides the list to parser **155**. One structure for conversion is an augmented transition network. Another approach to tokenizing is to scan the initial user query and group words into conceptual strings, removing plurals and suffixes. With such an approach, the longest strings can be grouped first, so that they are given greater priority over shorter strings.

Parser **155** identifies the set of possible syntactic structures that could represent the question(s) being asked and passes the structure set to normalizer **160**, with each syntactic structure representing one possible syntactic interpretation of the question. Parser **155** can also deal with adverbs and specialized parsers for ambiguous statements. For example, parser **155** might recognize that the questions "Who is the French president" and "Who is the president of France" has the same underlying syntactical structure.

Normalizer **160** reforms the syntactic structures into canonical forms by replacing synonyms with a canonical term. Using the canonical terms allows for a much more compact set of questions, since the many synonyms do not need to be handled. For example, a user query might use one of the synonyms "drizzle", "storming" or "misting" for raining. By reducing the synonyms to canonical form, the information server does not need to deal with so many questions because, in the above example, four questions collapse into one. Normalizer **160** uses a semantic map, a small portion of which is shown in FIG. **6**, to perform the canonical reduction.

Once normalizer **160** has reduced the structure set to a normalized structure set, normalizer **160** passes the normalized structure set to matcher **165**. Matcher **165** then matches the normalized structure set against semantic net snapshot **40**. The semantic net is changing as more questions are being asked and the semantic net is being refined, so an information server will often use a snapshot of the state of the semantic net at one point in time. However, either a snapshot or a live copy of the semantic net will work for its purposes in information servers, such as server **10** or server **50**. An example of a semantic net is shown in FIGS. **7**–**8**.

By matching the normalized structure set against a semantic net, as described in more detail below in the description of FIGS. **7**–**8**, matcher **165** obtains a list of instantiated questions (template questions with parameter values identified) and provides those to APE **32**. Since the list is of instantiated questions that are based on template questions, they will be found in question-answer mapping table **42** and APE **32** obtains the answers that match the questions.

FIG. **7** illustrates how a semantic net **200** might be organized to be used to map keywords to questions. Once keywords are mapped to questions, the questions are mapped to answers using question-answer mappings **202**. A small portion **204** of semantic net **200** is shown in detail in FIG. **8**.

FIG. **9** is a block diagram of a computer system for collecting information and generating run-time structures, such as semantic nets, from the collected information.

What is claimed is:

1. An information server for directing users of the information server to desired sources of information where the desired sources of information are determined, at least in part, based on user input, the information server comprising:

query input processor for accepting an initial user query;

a question database of template questions, wherein a template question is a question in the database that is coupled to at least one answer reference, the answer reference being one of an answer to the template question or a pointer to a possible location of an answer to the template question;

a question processor that processes the initial user query to identify a set of correlated template questions selected from the question database, wherein the question processor comprises a parser for generating a syntactic structure from the list of words and a normalizer for reducing the syntactic structure to a canonical syntactic structure;

means for presenting at least some of the set of template questions to the user;

means for accepting a user selection of a template question if more than one template question is presented; and

an answer processor which responds to a user selection of a presented template question from the presented set of template questions, wherein the response of the answer processor depends on the at least one answer reference coupled to the user selected question from the presented set of template questions.

2. The information server of claim **1**, wherein the user initial query is a text string comprising a sequence of one or more tokens, wherein a token is one or more words that have meaning together and a token is either a natural language question or a keyword string, which is a text string comprising one or more keywords in an order and zero or more logical connectors.

3. The information server of claim **1**, wherein the question processor comprises:

a tokenizer for tokenizing the initial user query into a list of words;

a parser for generating a syntactic structure from the list of words;

a normalizer for reducing the syntactic structure to a canonical syntactic structure; and

a matcher for matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.

4. The information server of claim 3, wherein the matcher comprises:

a structure mapper;

a substructure mapper; and

a keyword mapper.

5. The information server of claim 1, wherein the question processor comprises confidence weighting logic that compares confidence weights for each of the set of correlated template questions selected from the question database and removes template questions from the set that have a confidence weight below a predetermined threshold.

6. The information server of claim 5, wherein the predetermined threshold is a value corresponding to a drop off of confidence weights of template questions in the set.

7. A method for directing users to desired sources of information where the desired sources of information are determined, at least in part, based on user input using a question database of template questions, wherein a template question is a question in the database that is coupled to at least one answer reference, the answer reference being one of an answer to the template question or a pointer to a possible location of an answer to the template question, the method comprising:

receiving an initial user query;

processing the initial user query to identify a set of correlated template questions selected from the question database, wherein processing comprises generating a syntactic structure from a list of words and a normalizer for reducing the syntactic structure to a canonical syntactic structure;

presenting at least some of the set of template questions to the user;

accepting a user selection of a template question if more than one template question is presented; and

responding to a user selection of a presented template question from the presented set of template questions, wherein the response depends on the at least one answer reference coupled to the user selected question from the presented set of template questions.

8. The method of claim 7, further comprising:

tokenizing the initial user query into a list of words;

generating a syntactic structure from the list of words;

reducing the syntactic structure to a canonical syntactic structure; and

matching the canonical syntactic structure against a semantic network to obtain a weighted list of well-formed questions representative of possible semantic meanings for the initial user query.

9. The method of claim 7, further comprising:

comparing confidence weights for each of the set of correlated template questions selected from the question database; and

removing template questions from the set that have a confidence weight below a predetermined threshold.

* * * * *

PA-D

US 20020007303A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2002/0007303 A1**

Brookler et al. (43) **Pub. Date: Jan. 17, 2002**

(54) **SYSTEM FOR CONDUCTING ELECTRONIC SURVEYS**

(76) Inventors: **Brent D. Brookler**, Seattle, WA (US); **Brian E. Levin**, Seattle, WA (US); **Peter A. Loos**, Seabeck, WA (US)

Correspondence Address:
John F. Letchford
Klehr, Harrison, Harvey, Branzburg, & Ellers
260 South Broad Street
Philadelphia, PA 19102 (US)

(21) Appl. No.: **09/845,700**

(22) Filed: **Apr. 30, 2001**

Related U.S. Application Data

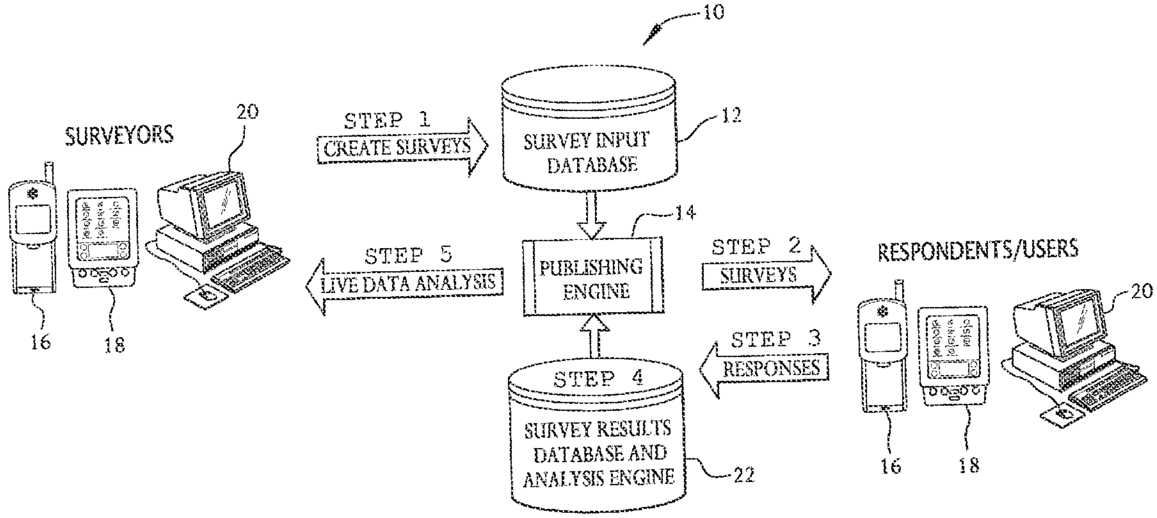
(63) Non-provisional of provisional application No. 60/201,011, filed on May 1, 2000.

Publication Classification

(51) **Int. Cl.**^{7} **G06F 17/60**
(52) **U.S. Cl.** **705/10**

(57) **ABSTRACT**

A system for simultaneous surveying and data collection from multiple types of electronic communication devices. The invention provides an apparatus and process for (1) creating a survey, (2) simultaneously publishing the survey to respondents via a plurality of types of electronic communications devices, and (3) making the results of the survey available to the creator of the survey via communications devices of the creator's choice.



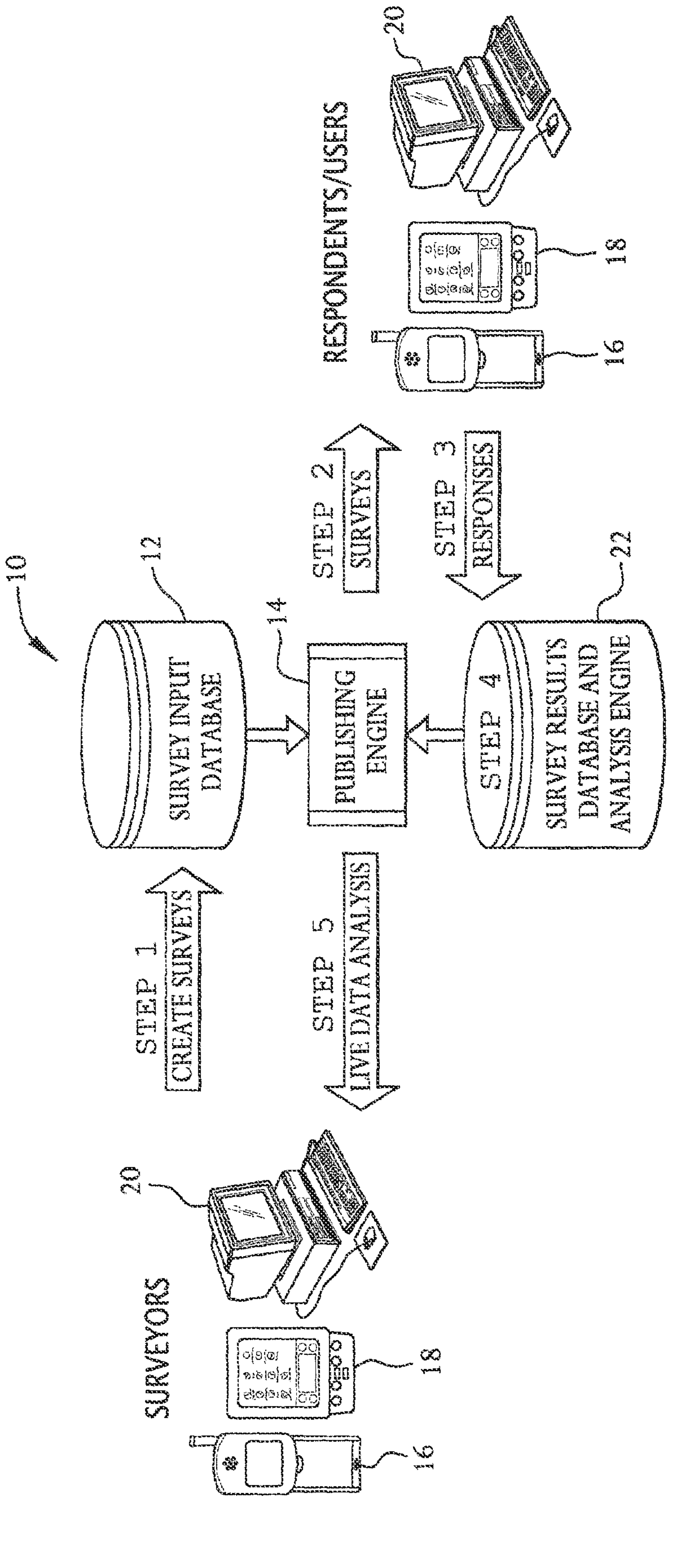

SURVEYORS
16
18
20
STEP 1
CREATE SURVEYS
STEP 5
LIVE DATA ANALYSIS
10
12
SURVEY INPUT DATABASE
14
PUBLISHING ENGINE
STEP 4
SURVEY RESULTS DATABASE AND ANALYSIS ENGINE
22
STEP 2
SURVEYS
STEP 3
RESPONSES
RESPONDENTS/USERS
16
18
20

FIG. 1

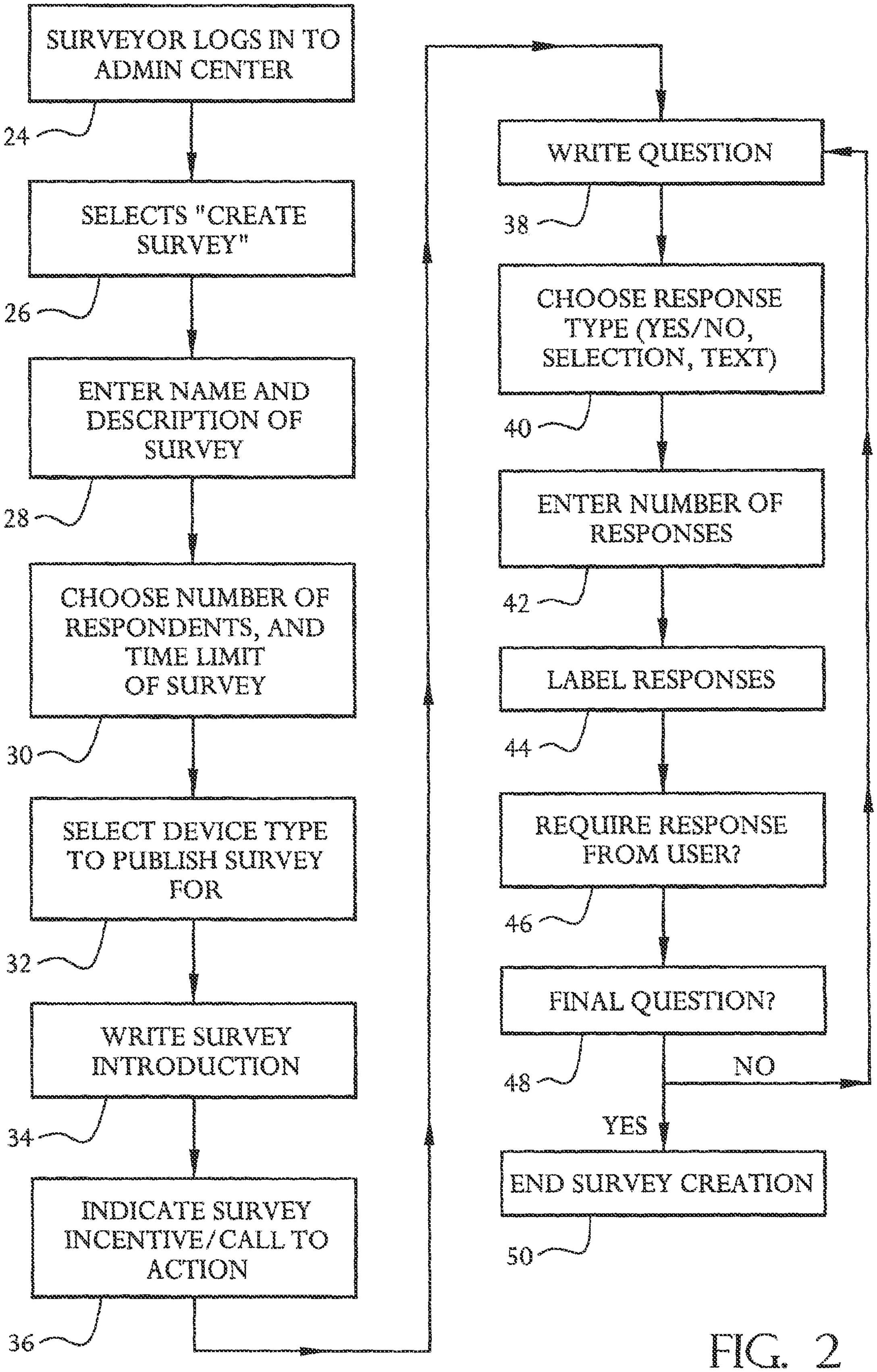

SURVEYOR LOGS IN TO ADMIN CENTER
24
SELECTS "CREATE SURVEY"
26
ENTER NAME AND DESCRIPTION OF SURVEY
28
CHOOSE NUMBER OF RESPONDENTS, AND TIME LIMIT OF SURVEY
30
SELECT DEVICE TYPE TO PUBLISH SURVEY FOR
32
WRITE SURVEY INTRODUCTION
34
INDICATE SURVEY INCENTIVE/CALL TO ACTION
36
WRITE QUESTION
38
CHOOSE RESPONSE TYPE (YES/NO, SELECTION, TEXT)
40
ENTER NUMBER OF RESPONSES
42
LABEL RESPONSES
44
REQUIRE RESPONSE FROM USER?
46
FINAL QUESTION?
48
NO
YES
END SURVEY CREATION
50

FIG. 2

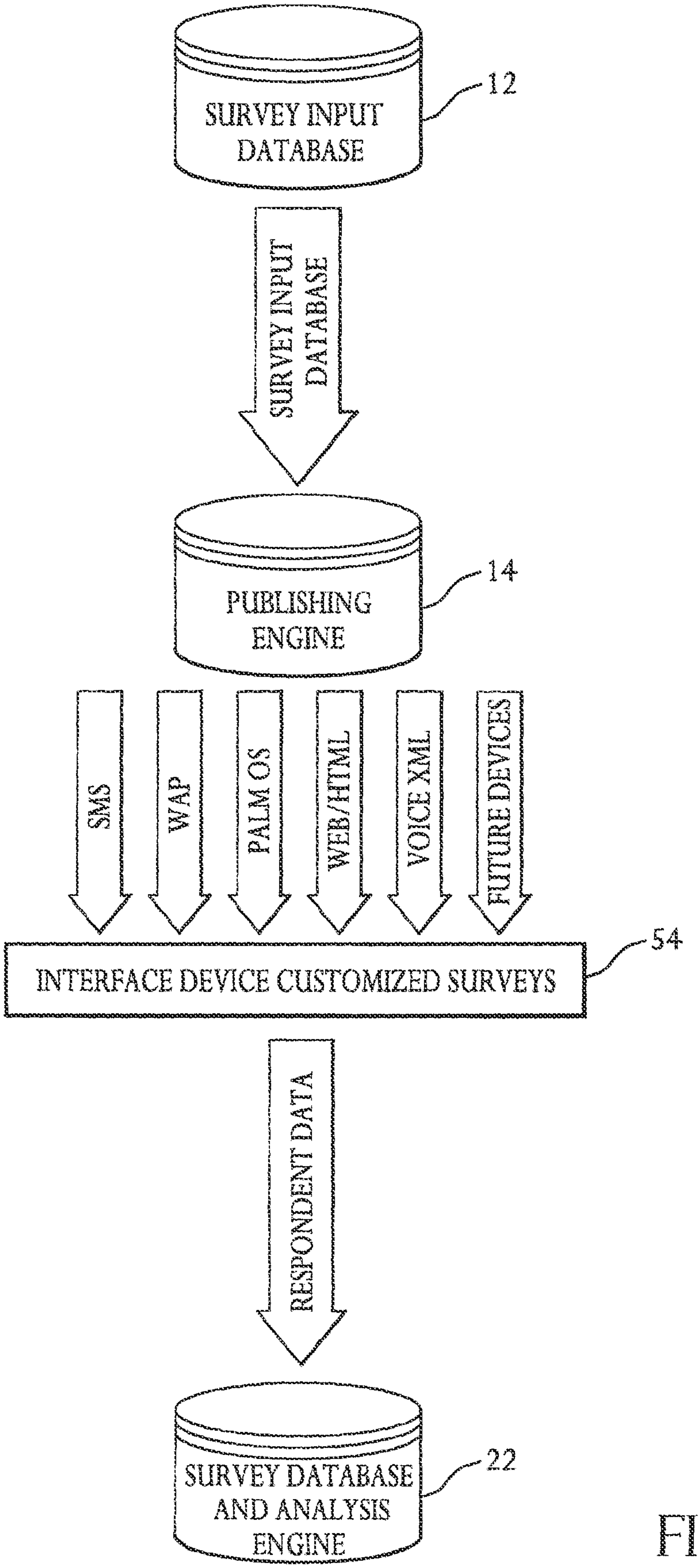

SURVEY INPUT DATABASE
12
SURVEY INPUT DATABASE
PUBLISHING ENGINE
14
SMS
WAP
PALM OS
WEB/HTML
VOICE XML
FUTURE DEVICES
INTERFACE DEVICE CUSTOMIZED SURVEYS
54
RESPONDENT DATA
SURVEY DATABASE AND ANALYSIS ENGINE
22

FIG. 3

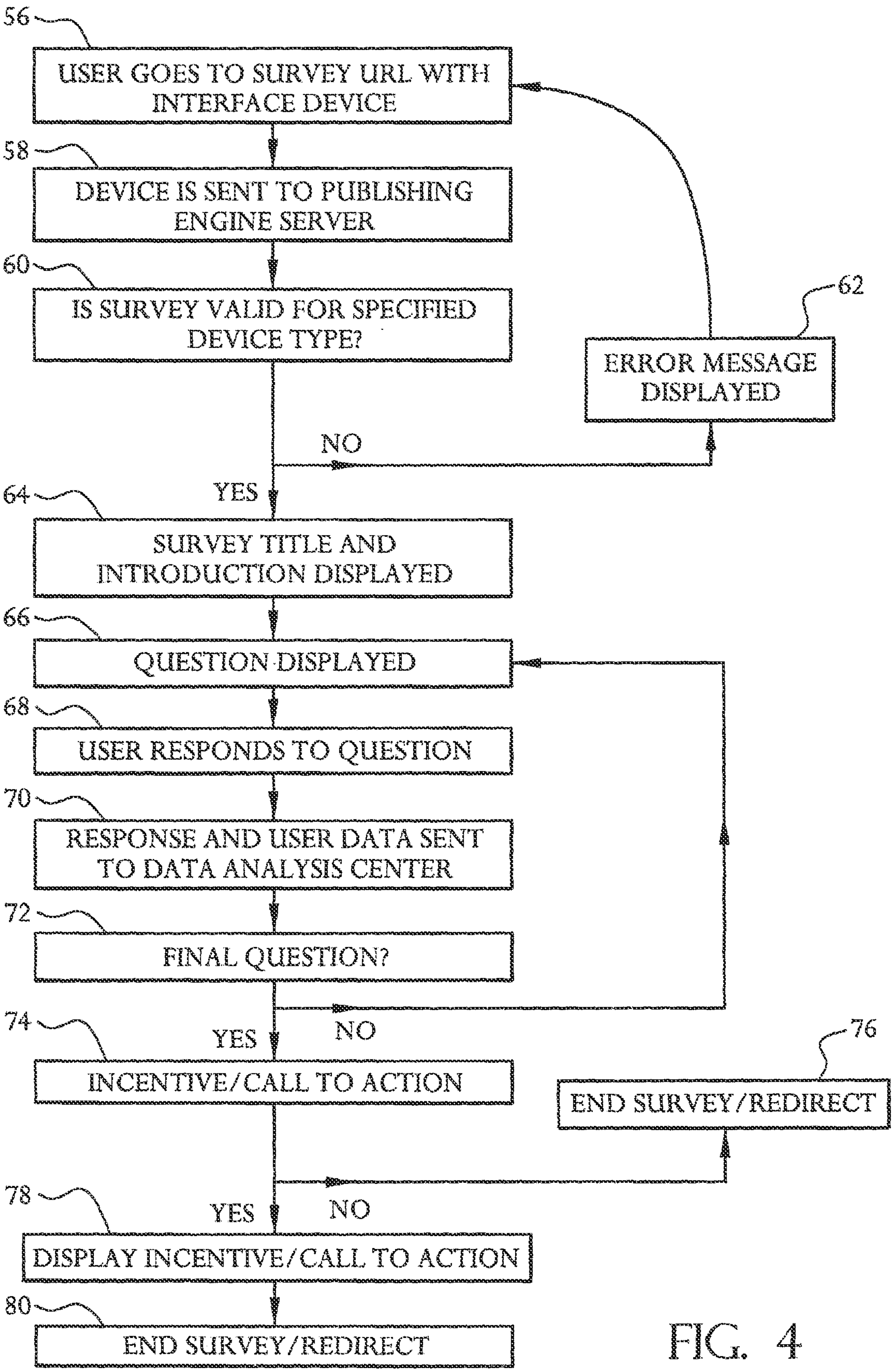

56
USER GOES TO SURVEY URL WITH INTERFACE DEVICE
58
DEVICE IS SENT TO PUBLISHING ENGINE SERVER
60
IS SURVEY VALID FOR SPECIFIED DEVICE TYPE?
62
ERROR MESSAGE DISPLAYED
NO
YES
64
SURVEY TITLE AND INTRODUCTION DISPLAYED
66
QUESTION DISPLAYED
68
USER RESPONDS TO QUESTION
70
RESPONSE AND USER DATA SENT TO DATA ANALYSIS CENTER
72
FINAL QUESTION?
74
YES
NO
INCENTIVE/CALL TO ACTION
76
END SURVEY/REDIRECT
78
YES
NO
DISPLAY INCENTIVE/CALL TO ACTION
80
END SURVEY/REDIRECT

FIG. 4

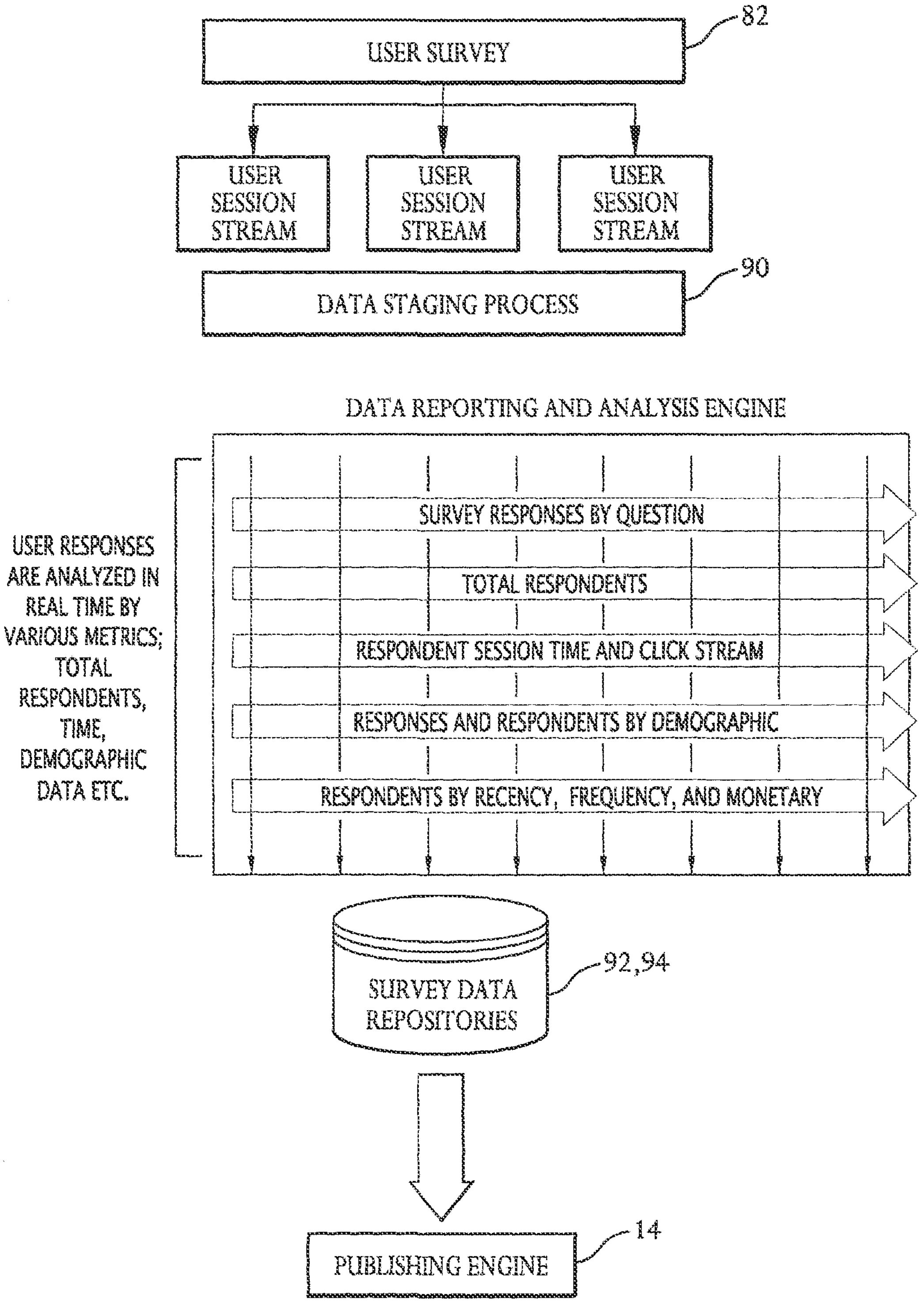

USER SURVEY
82
USER SESSION STREAM
USER SESSION STREAM
USER SESSION STREAM
DATA STAGING PROCESS
90
DATA REPORTING AND ANALYSIS ENGINE
USER RESPONSES ARE ANALYZED IN REAL TIME BY VARIOUS METRICS; TOTAL RESPONDENTS, TIME, DEMOGRAPHIC DATA ETC.
SURVEY RESPONSES BY QUESTION
TOTAL RESPONDENTS
RESPONDENT SESSION TIME AND CLICK STREAM
RESPONSES AND RESPONDENTS BY DEMOGRAPHIC
RESPONDENTS BY RECENCY, FREQUENCY, AND MONETARY
SURVEY DATA REPOSITORIES
92,94
PUBLISHING ENGINE
14

FIG. 5

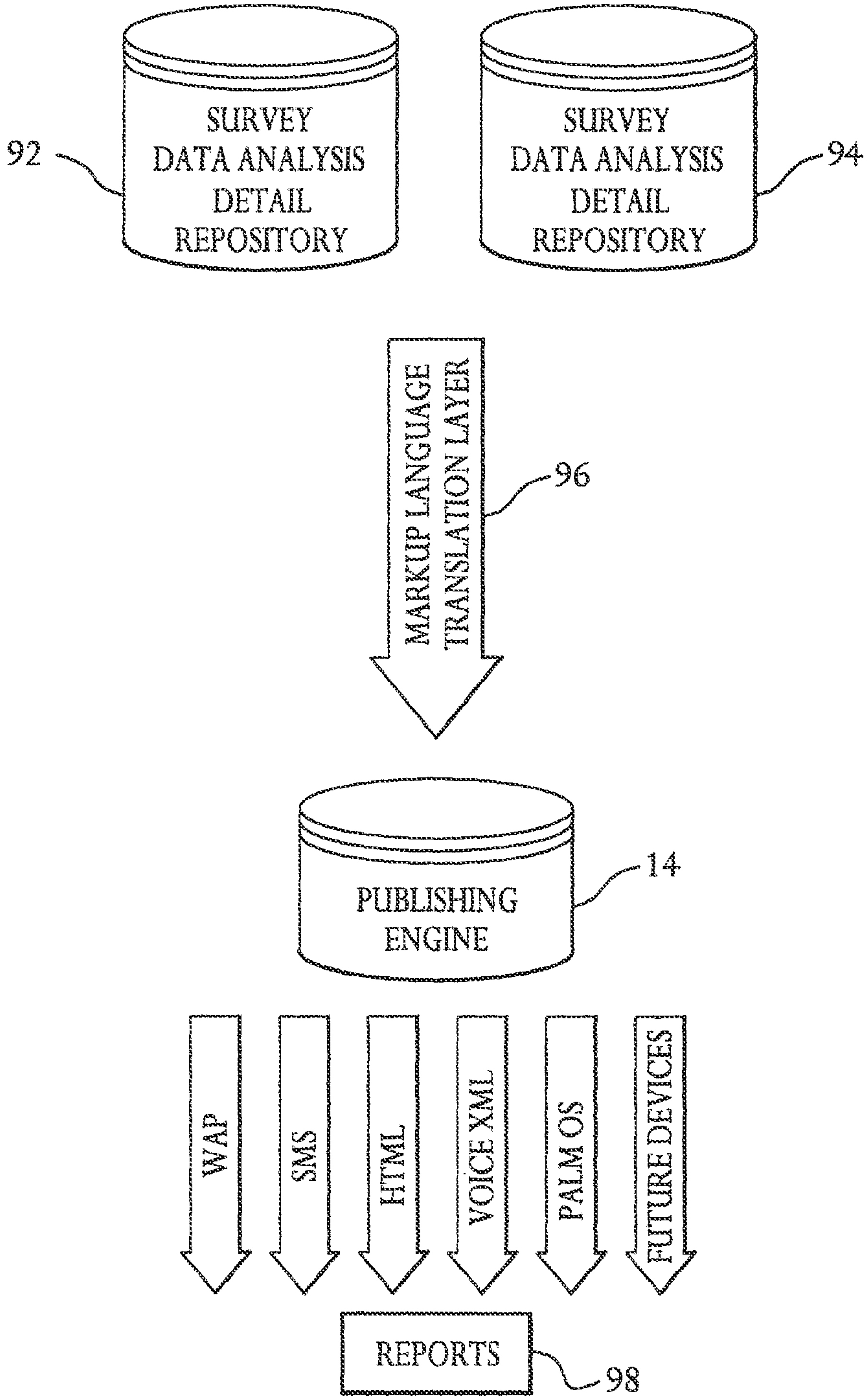

92
SURVEY DATA ANALYSIS DETAIL REPOSITORY
94
SURVEY DATA ANALYSIS DETAIL REPOSITORY
MARKUP LANGUAGE TRANSLATION LAYER
96
14
PUBLISHING ENGINE
WAP
SMS
HTML
VOICE XML
PALM OS
FUTURE DEVICES
REPORTS
98

FIG. 6

SYSTEM FOR CONDUCTING ELECTRONIC SURVEYS

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application claims the benefit of U.S. Provisional Application No. 60/201,011, filed May 1, 2000.

FIELD OF THE INVENTION

[0002] The present invention relates in general to a system including apparatus and methods for conducting surveys and, more particularly, to an electronic survey conducting system simultaneously operable on many kinds of electronic communications devices, including wireless devices.

BACKGROUND OF THE INVENTION

[0003] Surveying public opinion and thoughts has been done for a long period of time. Generally, the techniques for soliciting opinions have improved with improving technology, starting with paper and mail and progressing to telephone surveys and solicitations. Currently, the proliferation of communications media and devices has greatly expanded the opportunities and ways by which to survey targeted groups for their opinions and thoughts. For example, even exclusively within the context of Internet surveying, there are multiple means by which users or respondents communicate with the Internet and can respond to Internet surveys. However, as presently conducted, Internet surveys are device-specific. That is, they are designed to be carried out using only a single one of several classes of Internet-compatible communications devices, e.g., a personal computer (PC) or a wireless device such as a personal digital assistant (PDA) or cellular telephone (cell phone). This limits the scope of the survey to only one means of connectivity to a particular network. This may be problematic for a potential survey respondent who possesses only one type of communications device or who possesses more than one type of communications device but, because of preference, habit or necessity, tends to use one type of device more frequently, oftentimes considerably more frequently, than the others. Under these circumstances, such a potential respondent may not be captured within the pool of respondents whose input may be of importance to the survey administrator. This reduces the number of potential respondents that may be incorporated into the survey which, in turn, reduces the reliability of the survey results. Therefore, there is a need in the art for a system including methods and apparatus for conducting a survey with respondents having multiple means of connectivity to a particular network, whereby the respondents may participate in the survey regardless of the means by which they choose to connect to the network.

SUMMARY OF THE INVENTION

[0004] The present invention provides a system including methods and apparatus for simultaneous surveying and data collection from multiple types of electronic communication devices. The invention provides a process for (1) creating a survey, (2) simultaneously publishing the survey to respondents via a plurality of types of electronic communications devices, and (3) making the results of the survey available to the creator of the survey via communications devices of the creator's choice. More particularly, the process comprises

[0005] (a) creating a survey by writing the survey materials and placing the survey materials into a first database as survey input data;

[0006] (b) wrapping each element of survey input data with desired markup language tags defined in a schema to provide a collection of data in a markup language-wrapped document;

[0007] (c) publishing the markup language-wrapped document parsed using the survey input data, wherein the survey input data are in the form of a collection of markup language-wrapped data, by parsing the markup language-wrapped data against the schema;

[0008] (d) sending the parsed, markup language-wrapped data in output defined style sheets to a plurality of types of interface devices via suitable communications networks;

[0009] (e) receiving survey response data in a second database via suitable communications networks; and

[0010] (f) analyzing and publishing the retrieved data by wrapping the retrieved data in a desired markup language text.

[0011] Preferably, the markup language-wrapped data are further validated against a pre-defined schema. Additionally, the analyzed data is preferably further parsed against a second schema to enable the analyzed data to be accessed by at least one interface device type specified by a creator of the survey.

[0012] The invention further includes a survey publishing system for simultaneous surveying and data collection from multiple interface device types. The publishing system enables a party to create a survey and publish the survey to potential survey respondents having multiple interface device types. Additionally, the publishing system gathers and analyzes the survey results and makes the results available to the survey creator in one or more formats compatible with interface device types of the survey creator's choosing.

[0013] The survey publishing system comprises a survey input database into which the survey creator inputs and stores all relevant information associated with particular survey (including, without limitation, the question(s) to be answered by the respondents, the types of interface devices the surveyor chooses to receive the survey, and the time or numerical response limits of the survey). The system further comprises a publishing engine for transmitting the surveys stored in the survey input database in the desired formats to the desired interface devices. Upon receipt of the surveys, the respondents answer the questions posed therein and transmit their responses to a survey results database and analysis engine of the survey publishing system. From the survey results database and analysis engine, the analyzed survey results are sent to the publishing engine where they may be retrieved by the surveyors using the interface devices of their choice.

[0014] According to the present invention, therefore, surveyors can create surveys that can simultaneously reach multiple types of interface devices; respondents using a variety of interface devices may participate in a survey; and, surveyors receive more complete and meaningful survey information than heretofore available using conventional single device surveys.

[0015] Other details, objects and advantages of the present invention will become apparent as the following description of the presently preferred embodiments and presently preferred methods of practicing the invention proceeds.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The invention will become more readily apparent from the following description of preferred embodiments shown, by way of example only, in the accompanying drawings wherein:

[0017] **FIG. 1** is a symbolic diagram of the essential components of the survey publishing system according to the present invention and the overall process enabled by the system;

[0018] **FIG. 2** is a flowchart illustrating the process by which a surveyor creates a survey;

[0019] **FIG. 3** is a flowchart illustrating the process by which the survey is published to the respondents who will participate in the survey;

[0020] **FIG. 4** is a flowchart illustrating the process by which the survey is implemented and the data is collected from the respondents;

[0021] **FIG. 5** is a drawing showing the data analysis process used to analyze the data collected as shown in **FIG. 4**; and

[0022] **FIG. 6** is a drawing showing the process by which the results of the survey are published to the party who originally requested the survey.

DETAILED DESCRIPTION OF THE INVENTION

[0023] As used herein, the following terms shall have the meanings set forth below.

[0024] Schema is a document that defines tags.

[0025] Tags are definitions or titles of data specific for a particular user/surveyor agent.

[0026] Parser Engine parses data according to tags defined in the schema.

[0027] Style Sheet is an output specification corresponding to the display parameters of an output or interface device.

[0028] Referring to the drawings, wherein like or similar references designate like or similar elements throughout the several views, there is shown in **FIG. 1** the overall system according to the invention, including apparatus and process, for creating and publishing electronic surveys and for collecting and analyzing data generated by the surveys. In **FIG. 1** the survey creators or surveyors are depicted on the left side and the users, or survey respondents, on the right side. The overall process is generally represented as follows:

[0029] Step 1: Survey Creation: The surveyor creates a survey using a survey publishing system which is generally indicated by reference numeral **10**. During this step, the surveyor determines, inter alia, the type of survey, the number of respondents desired (or, alternatively, the time limit of survey), and desired interface device types on which to publish the survey on (PC, PDA, cell phone, etc.). Also, as will be more readily appreciated by reference to **FIG. 2** and its corresponding description, it is at this stage the survey questions are created by the surveyor. The survey information, including the questions, is stored in dedicated database tables in a survey input database **12**, where it can be accessed for later use.

[0030] Step 2: Survey Publishing: This step involves taking the initial survey data (questions, types, formats) and publishing them via suitable communications networks (wired or wireless) to various types of user/respondent computing or interface devices. Since there are many different device types, i.e., cell phone, PDA, PC, as well as future devices, that may be capable of implementing the present survey system, the formatting must be different for each type of device. As described hereinafter, a publishing engine **14** of survey publishing system **10** will determine which type of respondent interface device (e.g., cell phone **16**, PDA **18** or PC **20**) is making a request to respond to a survey. Thereafter, publishing engine **12** will publish the survey over to interface device **16**, **18** or **20** over an appropriate communications network (wired or wireless) in the appropriate format for that device.

[0031] Step 3: Survey Implementation/Data Collection: At this step, the user responds to the survey questions. The response data is transmitted by the survey respondent over the communications network and is collected at a survey results database and analysis engine **22** of survey publishing system **10**. This data includes not only the responses to the survey questions, but user data, such as interface device type, and session information, such as length of survey, time of day, and so on.

[0032] Step 4: Data Analysis: Once the response data is collected from a respondent's device, the unprocessed survey data is stored in "raw data" database tables of the survey results database. Statistical analysis is then performed on the raw data by the data analysis engine using stored procedures. The type of analysis is specified by the surveyor and analyzed or processed data results are stored in dedicated "analyzed data" database tables.

[0033] Step 5: Analyzed Data Publishing and Reporting: Once the data has been analyzed in accordance with parameters prescribed by the surveyor, reports and results are published by publishing engine **14**. As specified by the surveyor, publishing engine **14** makes the survey results and reports available for access by the surveyor via one or more interface devices (e.g., cell phone **16**, PDA **18** or PC **20**). The survey results may be "pushed" by the survey publishing system server to the surveyor in the manner known in the art. Alternatively, and preferably, the survey results and reports may passively reside on the survey publishing system server where they may be "pulled" or retrieved by the surveyor. In the latter case, survey publishing system **10** may be programmed to notify the surveyor via electronic mail message or otherwise that the survey results are available for access. It will be understood that publishing engine **14** will publish different levels and depth of data according to the interface device specified by the surveyor to receive the data, e.g., a cell phone will only get high level reports, whereas a PC will receive detailed analysis.

[0034] **FIG. 2** illustrates the survey creation process. This process preferably occurs in a secure administration environment, such as a password protected web site, where the surveyor can access all of its pertinent information, and build surveys to be presented to users. The survey creation process is outlined below.

[0035] Initially, at step **24**, the surveyor logs in to the administration center's secure website, enters the appropriate user and password information to gain access to the survey publishing system **10** and, upon gaining such access, selects "Create Survey" at step **26**. Thereafter, the surveyor determines all the variable factors of the survey. The variable factors include:

[0036] Name of survey (step **28**)—quite simply, the title of the survey that can be used by both the surveyor and the respondents to refer to the survey.

[0037] Description of survey (step **28**)—information provided by the surveyor for the purpose of describing the survey in a text field.

[0038] Number of respondents desired (step **30**)—at this step, the surveyor selects how many survey responses it wants to limit the survey to. The surveyor may also select the option of "no limit."

[0039] Time limit (step **30**)—the surveyor selects an ending date/time for the poll.

[0040] Priority: time or number of responses (step **30**)—the surveyor chooses which should take priority, the time limit of the survey or the number of responses.

[0041] Type of devices desired (step **32**)—the surveyor selects what kinds of interface devices they want to be able to respond to the survey. This will determine how and on which devices the poll will be published, e.g., cell phone, PDA, PC, as well as future devices.

[0042] Survey Introduction (step **34**)—this enables the surveyor to input a narrative introduction to the survey including information such as the purpose of the survey and the sorts of information that the surveyor is seeking to collect. The survey introduction appears as text area that users will see on their interface devices when they begin to take the survey.

[0043] Survey Incentive/Call to Action (step **36**)—at this point, if desired, the surveyor can create an incentive for a user who responds to the survey. It may be in the form of a text field to be entered by the client and can be presented as a discount, offer, or a uniform resource locator (URL) or hyperlink that can serve as a call to action.

[0044] Write question (step **38**)—this is a text area that enables the surveyor to enter a question to be posed to the user.

[0045] Choose response type (step **40**)—this step allows the surveyor to select the type of responses for the question, e.g., Yes/No, True/false, single selections, multiple selections, text boxes, numerical rating, numerical, Agree/Disagree scale, and so on.

[0046] Enter number of responses (step **42**)—if the chosen response type is not Yes/No or True/False, then the surveyor will specify how many responses to choose from for that question.

[0047] Label responses (step **44**)—this is a text area that enables the surveyor to provide descriptive text for the different responses based on the different question types.

[0048] Require response from user? (step **46**)—this step enables the surveyor to specify whether a response to a particular question is mandatory.

[0049] Final question? (step **48**)—if there are more questions to be created, the surveyor returns to write question step **38** and repeats steps **38-46**). If there are no more questions to be added to the survey, the surveyor so indicates and the survey creation process terminates at step **50**.

[0050] **FIG. 3** illustrates the process of publishing the survey created according to **FIG. 2**. Upon completion of creation of a survey, the survey parameters are stored in dedicated database tables in survey input database **12**. Publishing engine **14** thereafter publishes or "pushes" the survey in the appropriate formats to the interface device types targeted for the survey. To do so, survey publishing system **10** employs a markup language translation layer **52** that wraps each element of survey input data with markup tags defined in a schema to provide a collection of data in a markup language-wrapped document. The markup language translation layer **52** preferably utilizes a plurality of markup language technologies in order to leverage the survey data to multiple types of interface devices. According to a presently preferred embodiment, markup language translation layer **52** utilizes extensible markup language (XML), standard query language (SQL) and dynamic page creation technologies such as JAVA and PERL to achieve the desired objectives. It will be understood that markup language translation layer **52** may include any combination of the foregoing alone or in combination with one or more of hypertext markup language (HTML), wireless markup language (WML), user interface markup language (UIML) or other form of presently existing or yet to be developed standard generalized markup language (SGML) that may be used to realize the objectives of the present invention. Most preferably, the markup language translation layer **52** is preferably readily programmable or configurable to accommodate any markup languages that may be required to push survey data in survey input database **12** in formats appropriate to all presently known and hereinafter developed interface devices.

[0051] According to a presently preferred embodiment, data collected during the survey building or creation process is stored in the survey input database **12** and translated to XML for optimal portability vis-a-vis presently available interface devices. Using publishing engine **14**, the surveyor may opt to publish the survey immediately after creation and simultaneously to all types of devices. In the alternative, the surveyor may choose to delay the launch of the survey or stagger the times at which the survey information is published to the various interface devices selected for participation in the survey. By way of example, publishing engine **14** may at present be configured to publish to cell phones using the wireless application protocol (WAP) (which incorporates WML), short messaging service (SMS) using the global system for mobile communication (GSM) or VoiceXML. Likewise, publishing engine may also publish to computer browsers via HTML and to Palm® devices or other PDAs using PalmOS or other suitable PDA operating systems (or SMS) depending on where the surveyor wishes the poll to be published.

[0052] Preferably, with one data set, all Internet appliances or interface devices are deployed ubiquitously. The following is a brief discussion of the formats having the broadest compatibility with presently available interface device technology.

[0053] WAP

[0054] WAP is primarily WML and WML script, but many different cell phones (User Agents) require slight modifications to the WAP standard. Preferably, markup language translation layer **52** includes a database of User Agents and their respective differences versus standard WAP. The present inventors have discovered that by using XML, customized style sheets may be created for each User Agent to ensure that all devices will work as designed.

[0055] HTML

[0056] For publishing to HTML, it is preferable to use HTML 3.2 to allow some retroactive compatibility with older versions of web browsers. In any event, HTML surveys should be 100% compliant with Netscape Navigator® 4.0 and later and Microsoft Internet Explorer® 4.0 and later, which presently account for more than 90% of the browser market.

[0057] PalmOS

[0058] For PalmOS, publishing engine **14** should be capable of publish using the current Palm-compatible formats including Web Clipping and Palm Query Application (PQA). Using XML or other SGML, it would preferable to create versions of PDA operating systems that are compatible with Palm® and other PDAs.

[0059] Once pushed to the desired interface devices, the formatted surveys reside as customized surveys **54** on the various devices. The respondents may then participate in the surveys and transmit their respondent data over the appropriate communication networks (wired or wireless) to the survey results database and analysis engine **22**.

[0060] **FIG. 4** illustrates the process by which respondents reply to the survey. The process begins when the user, at step **56**, directs his or her interface device to a URL associated with the stored survey. Data is collected in three distinct data streams during the survey implementation/data collection process—user profile, user session, and user response. To verify a particular device type, at step **58** data identifying the responding device is transmitted to the publishing engine **14**. At the publishing engine, the device type is compared, at step **60**, against a database table for acceptable device types. If the device is not found to be valid, an error message is displayed, at step **62**, on the user's device at which point the user may again to participate in the survey at step **56**.

[0061] If the user's device is determined to be valid, the survey begins. Initially, at step **64**, the survey title and description are displayed, followed at step **66** by the first survey question **66**. At step **68** the user responds to the first question and the response and user data is transmitted, at step **70**, to the survey results and analysis engine **22**. The survey questions and user responses thereto continue to be generated in turn until the final question is displayed and responded to at step **72**. At this point, the system logic checks, at step **74**, to determine whether the surveyor has not specified an incentive/call to action to motivate the user to respond to the survey. If not, the survey ends at step **76**, and the user is optionally redirected to the application that the user was using, if any, prior participating in the survey. If so, the incentive/call to action is displayed at step **78** and the survey is terminated at step **80**. Again, the user may be optionally redirected to the application that the user was using, if any, prior participating in the survey.

[0062] Session data is preferably gathered for each user that visits the survey, whether they personalize a user profile or not. Session data desirably includes login time, click stream, time spent on each question, and logout time. Anonymous user profiles are preferably created for each new visit to the survey, unless the user establishes a personalized profile and logs into the survey with a password. All profiles are recorded as a unique numeric value and are used to correlate session data with unique visits. Personalized profiles can contain a variety of additional personal information including e-mail/device address, zip code, age, gender and/or other relevant information.

[0063] User responses to the survey questions are collected and recorded with the unique numeric value of the user profile (anonymous or personalized) to allow for the correlation of user session, profile, and response data. The process of collecting the data from the user is represented in **FIGS. 5 and 6**.

[0064] **FIG. 5** illustrates the process by which the data collected from survey respondents is analyzed. The data analysis process is essential for providing value to the surveyors creating the polls in that it offers them real time top level results and detailed analysis and reporting.

[0065] Using data compression to speed analysis and delivery of results to publishing engine **14** ensures real time results. The completed user survey is transmitted to the publishing engine **14** in a user session stream **84**, a user profile stream **86** and a user response stream **88**. These data streams are compiled in a data staging process **90**. At this point the analysis engine **22** analyzes the data in accordance with reporting requirements established by the surveyor. The analysis engine **22** may analyze the response data in accordance with any criteria chosen by the surveyor, for example, survey responses by question, totals responses, respondent session time and click stream, responses and responses by demographic, and respondents by recency, frequency and monetary (discussed below). Depending on the device used to view published results, additional drill-down and drill-through requests are supported using common key elements in survey data analysis repositories **92** and **94**, discussed below. The functions of the data staging process **88**, analysis engine **22** and survey data repositories **92**,**94** are as follows.

[0066] Data Staging Process

[0067] As the data is received from the three data streams **84**, **86** and **88**, it is immediately consumed by the data staging process **90**. The data staging process **90** cleanses, compresses, and prioritizes the data received to ensure efficient processing by the analysis engine **22**.

[0068] Analysis Engine

[0069] Surveyor reporting requirements, which are established during survey creation, are applied to the data received from the data staging process **90**. Recognizing the priority of each piece of data, the analysis engine **22** produces new dimensions and updates existing dimensions as required, thereby producing detail and aggregate (summary) data streams to the survey data analysis repositories **92** and **94**, respectively.

[0070] Data Repositories

[0071] Survey data analysis detail and aggregate repositories **92**,**94** are inextricably linked through session, profile, and response data. Strong validation combined with indexing and performance tuning provides a reliable and efficient data store to be used by the publishing engine **14**.

[0072] **FIG. 6** illustrates the process by which the results of a survey are reported to the surveyor who requested them. The analyzed data that results from the surveys will be transmitted to the surveyor's administration area, where surveys are created. The process is similar to the publishing of created surveys to users, except the information is published to the surveyor's administration area of publishing engine **14**. The process is outlined below.

[0073] The data contained in data repositories **92**,**94** is converted by a markup language translation layer **96** similar in content and function to markup language translation layer **52** discussed above. That is, markup language translation layer **96** wraps each element of survey response data with markup tags defined in a schema to provide a collection of data in a markup language-wrapped document. Standard and/or customized reports **98** prescribed by the surveyor may be provided to the surveyor in WAP, HTML, PalmOS, SMS, VoiceXML or other formats and style sheets suitable to the surveyor's specified interface device(s), whether presently existing or hereinafter developed. Standard reports may include one or more of the following: total number of respondents to survey, percentage of answers by question, average percentage of questions answered, average session length, average time to respond to each answer, total number of respondents who responded to call to action, and total number of call to actions served.

[0074] Each of the reports will may also be sorted by date/time, by RFM (R=Recency—how recently did the profile (user) visit; F=Frequency—how frequently has the profile visited; M=Monetary—how much in total has the profile spent with the surveyor and/or provided benefit to the surveyor), as well as by gender, gender by age group, age group, home zip code, location (cell), and device type used to respond to the survey.

[0075] Although the following generally sets forth the overall survey conducting process of the present invention as it would be conducted using an XML format, it is contemplated that the present invention may be used in conjunction with any presently known markup language formats currently known or developed in the future. More particularly, the process comprises

[0076] (a) creating a survey by writing the survey materials and placing the survey materials into a first database as survey input data;

[0077] (b) wrapping each element of survey input data with desired markup language tags (e.g., XML tags) defined in a schema to provide a collection of data in a markup language-wrapped (e.g., XML-wrapped) document;

[0078] (c) publishing the markup language (e.g., XML) wrapped document, wherein the survey input data are in the form of a collection of markup language-wrapped (e.g., XML-wrapped) data, by parsing the markup language-wrapped (e.g., XML-wrapped) data against the schema;

[0079] (d) sending the parsed, markup language-wrapped (e.g., XML-wrapped) data in output defined style sheets to a plurality of types of interface devices via suitable communications networks;

[0080] (e) receiving survey response data in a second database via suitable communications networks;

[0081] (f) analyzing the received data; and

[0082] (g) publishing the received data by wrapping the received data in a desired markup language (e.g., an XML) document.

[0083] Preferably, the markup language-wrapped data are further validated against a pre-defined schema. Additionally, the analyzed data is preferably further parsed against a second schema to enable the analyzed data to be accessed by at least one interface device type specified by a creator of the survey.

[0084] Although the invention has been described in detail for the purpose of illustration, it is to be understood that such detail is solely for that purpose and that variations can be made therein by those skilled in the art without departing from the spirit and scope of the invention as claimed herein.

What is claimed is:

1. A process for conducting an electronic survey, said process comprising the steps of:

creating an electronic survey;

sending said survey to a plurality of users having different types of electronic interface devices; and

accessing user data generated in response to said survey using at least one type of electronic interface device.

2. The process of claim 1 further comprising accessing said user data using plurality of types of electronic interface devices.

3. The process of claim 1 further comprising storing said survey on a first database and receiving said user data on a second database.

4. The process of claim 3 further comprising analyzing said user data prior to said step of accessing.

5. The process of claim 4 wherein said step of analyzing comprises analyzing said user data in accordance with criteria established by a creator of said survey.

6. A process for conducting an electronic survey, said process comprising the steps of:

(a) creating a survey by writing the survey materials and placing the survey materials into a first database as survey input data;

(b) wrapping each element of said survey input data with markup language tags defined in a schema to provide a collection of data in a markup language-wrapped document;

(c) publishing said markup language-wrapped document, wherein said survey input data are in the form of a collection of markup language-wrapped data, by parsing said markup language-wrapped data against said schema;

(d) sending the parsed, markup language-wrapped data in output defined style sheets to a plurality of types of interface devices via suitable communications networks;

(e) receiving survey response data in a second database via suitable communications networks;

(f) publishing the received data by wrapping the received data in a desired markup language document.

7. The process of claim 6 further comprising accessing said received data using at least one type of electronic interface device.

8. The process of claim 6 further comprising accessing said user data using plurality of types of electronic interface devices.

9. The process of claim 6 wherein said step of accessing is performed by a creator of said survey.

10. The process of claim 6 further comprising analyzing the received data prior to publishing the received data.

11. The process of claim 6 wherein said markup language-wrapped data are further validated against a predefined schema.

12. The process of claim 6 wherein said received data is further parsed against a second schema to enable the analyzed data to be accessed by at least one interface device type specified by a creator of the survey.

13. The process of claim 6 wherein said markup language is extensible markup language.

14. Apparatus for conducting an electronic survey, said apparatus comprising:

a first database for storing an electronic survey comprised of survey input data;

a publishing engine for sending said survey to a plurality of users having different types of electronic interface devices via suitable communications networks; and

a second database for receiving survey response data from said electronic interface devices via suitable communications networks, wherein said publishing engine further publishes said survey response data for access by at least one type of electronic interface device.

15. The apparatus of claim 14 further comprising means for analyzing said response data prior to publishing by said publishing engine.

16. The apparatus of claim 15 wherein said analyzing means analyzes said user data in accordance with criteria established by a creator of said survey.

17. The apparatus of claim 14 wherein, prior to sending said survey to a plurality of users, said publishing engine wraps each element of said survey input data with markup language tags defined in a schema to provide a collection of data in a markup language-wrapped document.

18. The apparatus of claim 14 wherein, prior to publishing said survey response data, said publishing engine wraps each element of said survey response data with markup language tags defined in a schema to provide a collection of data in a markup language-wrapped document.

19. The apparatus of claim 17 wherein said markup language is extensible markup language.

20. The apparatus of claim 18 wherein said markup language is extensible markup language.

* * * * *

PA-H

US005442786A

United States Patent [19]

Bowen

[11] **Patent Number: 5,442,786**

[45] **Date of Patent: Aug. 15, 1995**

[54] **METHOD FOR RECORDING USER INTERACTION WITH A COMPUTER DATABASE TO GENERATE REPORTS**

[76] Inventor: **Robert E. Bowen**, 1515 S. 152nd Ave. Cir., Omaha, Nebr. 68144

[21] Appl. No.: **234,223**

[22] Filed: **Apr. 28, 1994**

[51] **Int. Cl.**^{6} **G06F 17/30**

[52] **U.S. Cl.** **395/600**; 364/943; 364/974; 364/963; 364/DIG. 2

[58] **Field of Search** 395/600

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,348,744	9/1982	White	395/200
4,954,699	9/1990	Coffey et al.	235/462
4,975,840	12/1990	DeTore et al.	364/401
5,025,374	6/1991	Roizen et al.	364/413.02
5,159,667	10/1992	Borrey et al.	395/148
5,326,270	7/1994	Ostby et al.	434/362
5,341,291	8/1994	Roizen et al.	364/413.02

OTHER PUBLICATIONS

D. Goodman "The Complete HyperCard 2.2 Handbook" Random House, 1987, pp. 44–61, 216–219.

N. Yankelovich et al "Reading and Writing the Electronic Book" IEEE Computer, Oct. 1985, pp. 15–30.

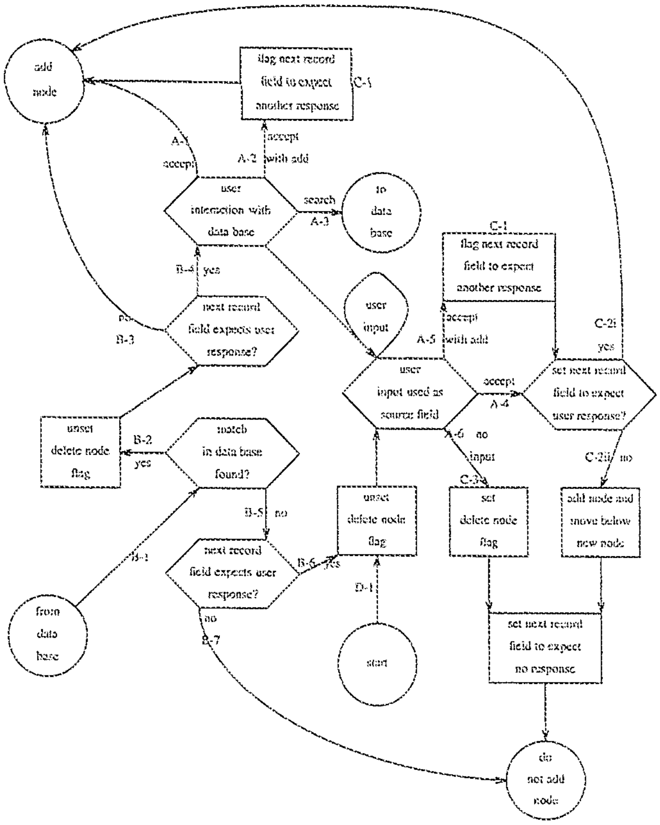
J. Conklin "Hypertext: An Introduction and Survey" IEEE Computer, Sep. 1987, pp. 17–41.

Primary Examiner—Thomas G. Black
Assistant Examiner—Peter Y. Wang
Attorney, Agent, or Firm—Zarley, McKee, Thomte Voorhees & Sease; Mark D. Frederiksen

[57] **ABSTRACT**

A method for creating a hierarchical data tree includes the initial step of providing a computer information processing system with a data base which interacts with user input to build the data tree based upon system responses. The user initially designates a source field and a next record field and the system searches the data base for records which have a target field matching the designated source field. The next record field of each record determines interaction with records, which have target fields matching the displayed record's source field. If a response is expected from the user, the user may either accept the record displayed or provide a separate user response as a selection. If no response is expected, according to the next record field, the system automatically selects the first matching record found. The system records the selection, whether from the system or from the user, utilizing a block of computer memory. The system then utilizes the recorded node as the basis for further search and further interaction. The system first creates a vertical leg of the data tree, before creating horizontal branches, according to the rules of interaction.

6 Claims, 12 Drawing Sheets


add node
flag next record field to expect another response
C-5
A-1
accept
A-2
accept with add
user interaction with data base
search
A-3
to data base
C-1
flag next record field to expect another response
B-4
yes
next record field expects user response?
no
B-3
user input
A-5
accept with add
C-2i
yes
user input used as source field
accept
A-4
set next record field to expect user response?
unset delete node flag
B-2
yes
match in data base found?
A-6
no input
C-3
C-2ii
no
B-5
no
unset delete node flag
set delete node flag
add node and move below new node
B-1
next record field expects user response?
B-6
yes
D-1
from data base
no
B-7
start
set next record field to expect no response
do not add node

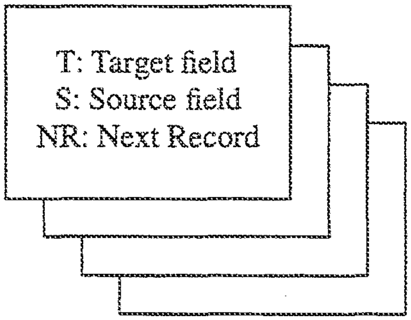


Figure 1

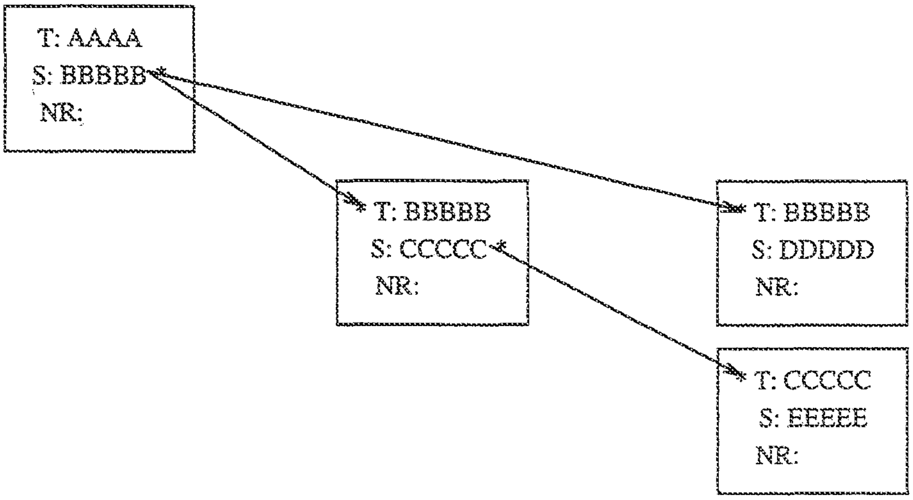


Figure 2

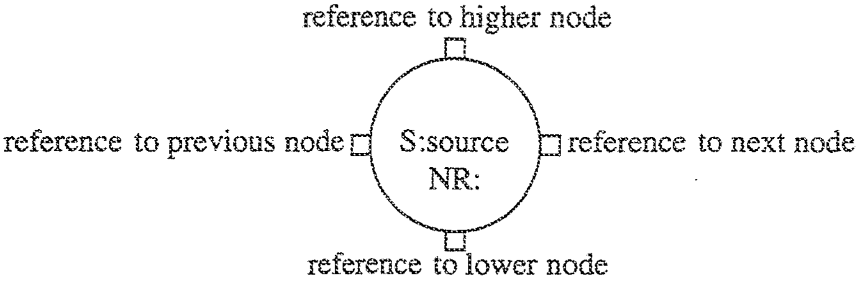


Figure 3

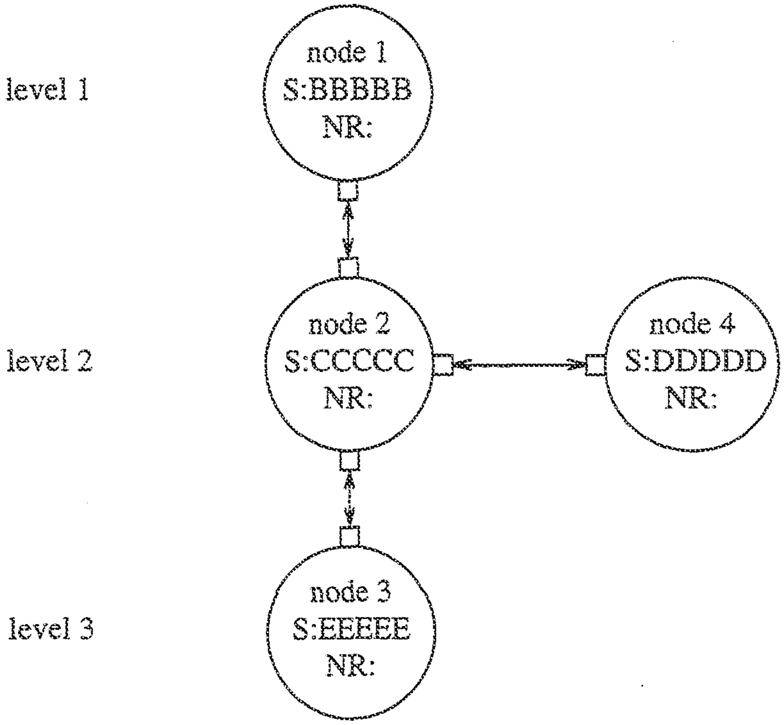


Figure 4

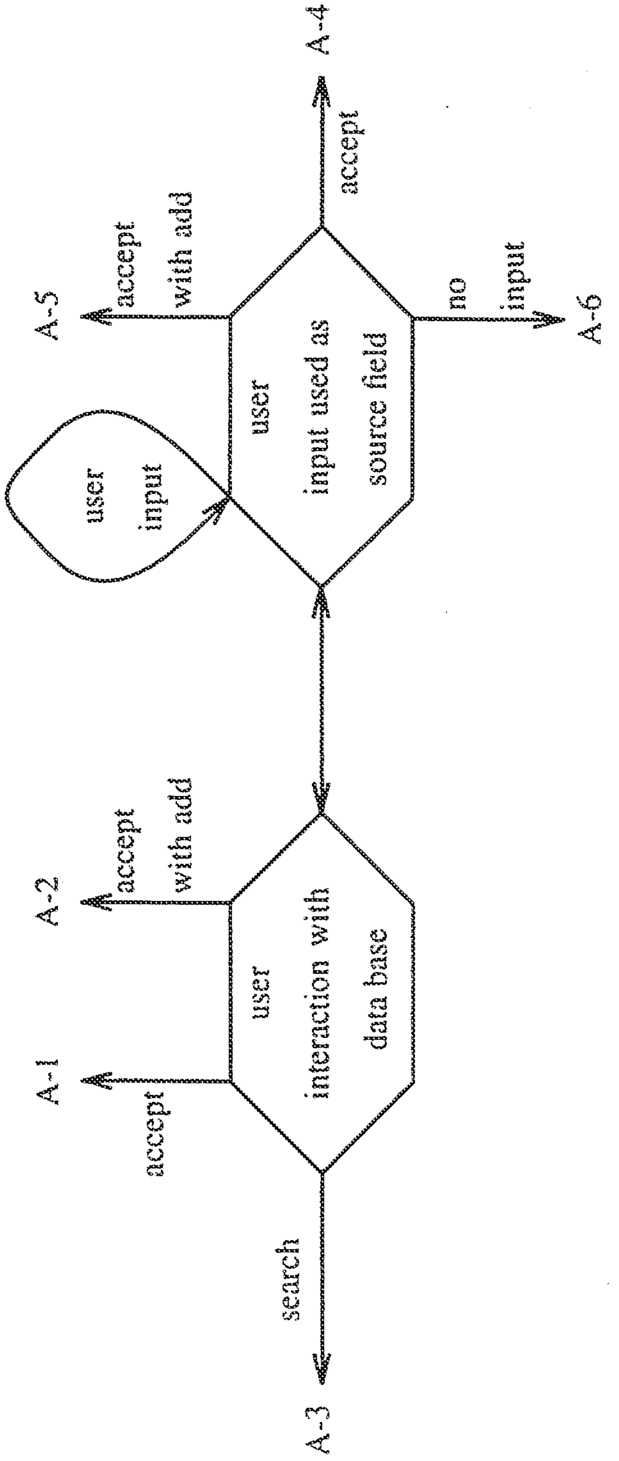

A-4
accept
A-5
accept with add
no input
A-6
user input used as source field
user input
A-2
accept with add
A-1
accept
user interaction with data base
search
A-3

Figure 5

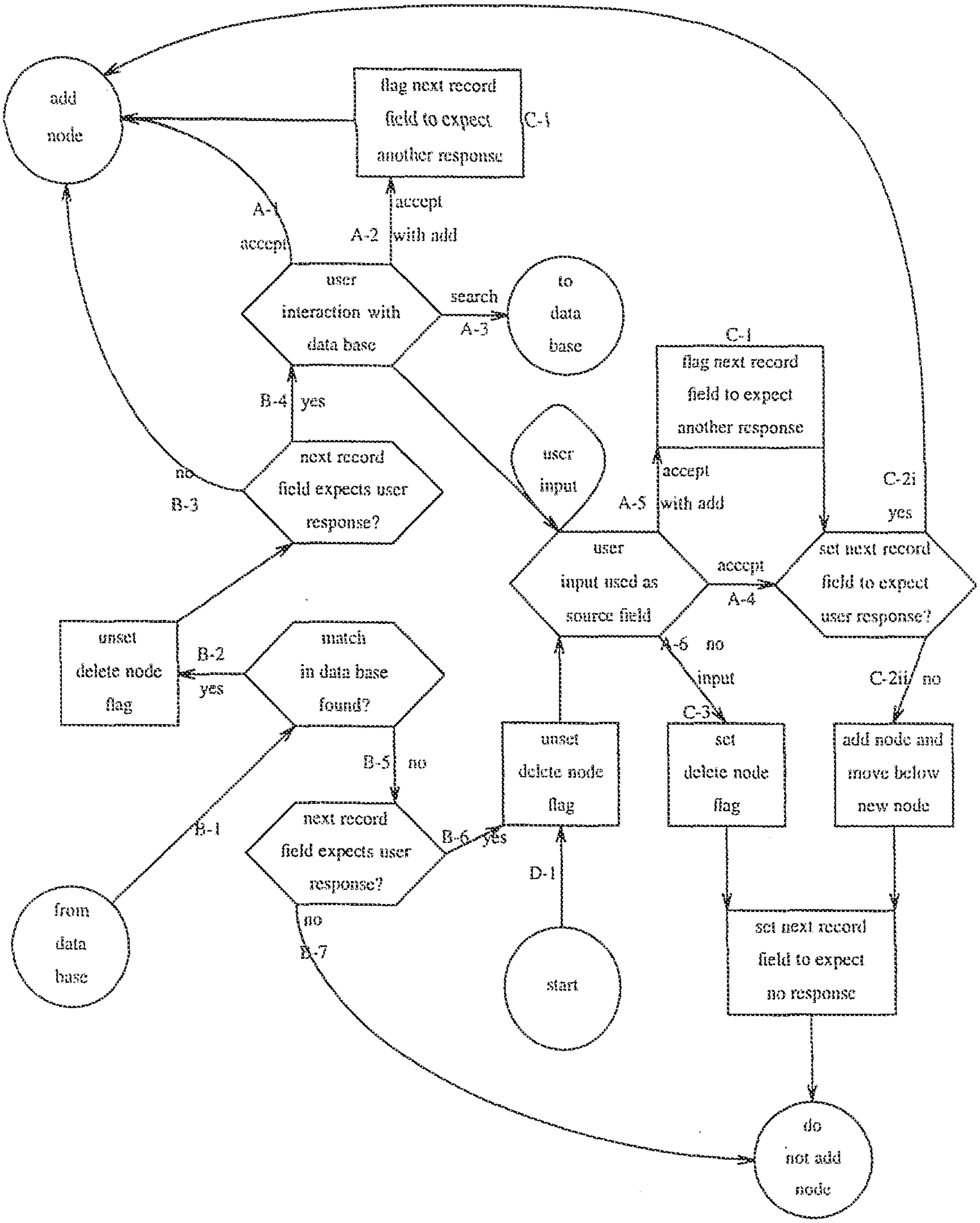

add node
flag next record field to expect another response
C-1
A-1
accept
A-2
accept with add
user interaction with data base
search
A-3
to data base
C-1
flag next record field to expect another response
B-4
yes
next record field expects user response?
no
B-3
user input
A-5
accept with add
C-2i
yes
user input used as source field
accept
A-4
set next record field to expect user response?
unset delete node flag
B-2
yes
match in data base found?
A-6
no input
C-2ii
no
C-3
B-5
no
unset delete node flag
set delete node flag
add node and move below new node
B-1
next record field expects user response?
B-6
yes
D-1
no
B-7
from data base
start
set next record field to expect no response
do not add node

Figure 6

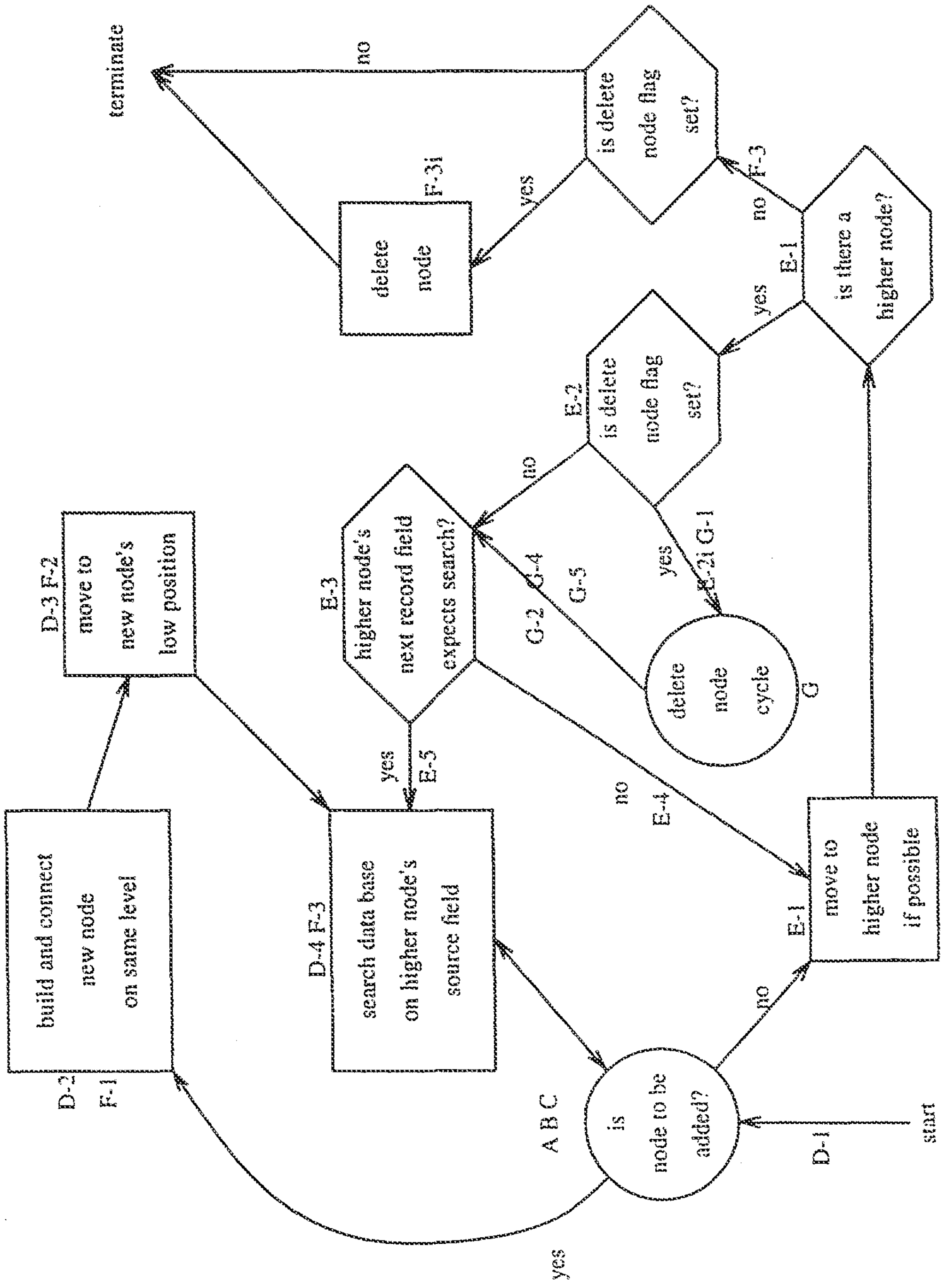

start
D-1
A B C
is node to be added?
yes
no
D-2
F-1
build and connect new node on same level
D-3 F-2
move to new node's low position
D-4 F-3
search data base on higher node's source field
E-1
move to higher node if possible
no
E-4
E-3
higher node's next record field expects search?
yes
E-5
E-1
is there a higher node?
yes
no
E-2
is delete node flag set?
no
yes
E-2i G-1
delete node cycle
G
G-2 G-4
G-5
F-3
is delete node flag set?
yes
no
F-3i
delete node
terminate

Figure 7

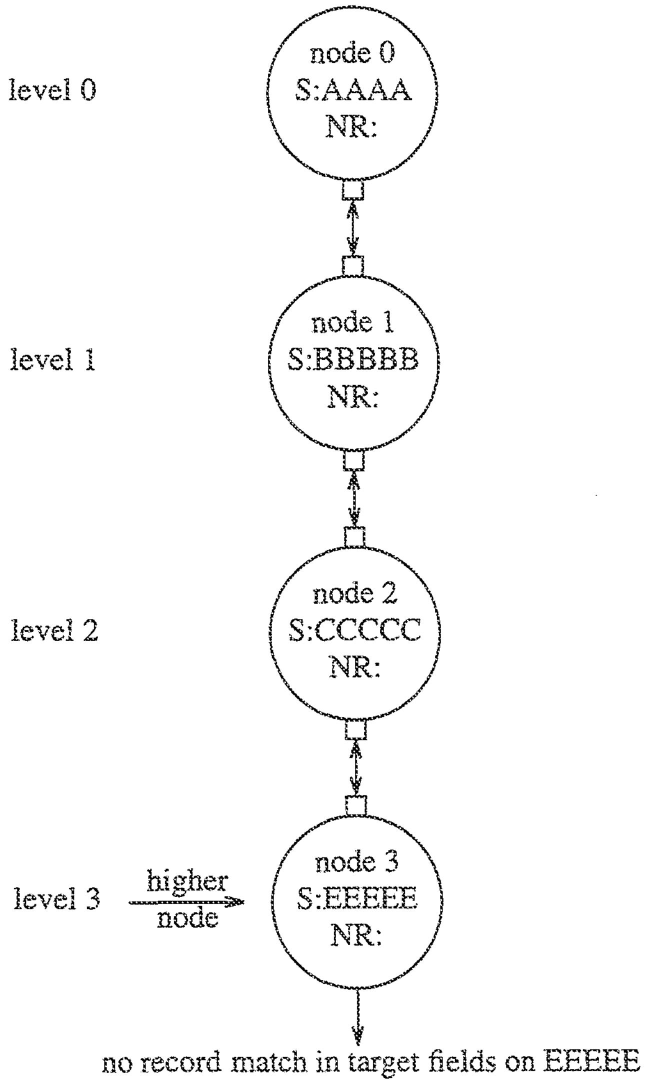

level 0
node 0
S:AAAA
NR:
level 1
node 1
S:BBBBB
NR:
level 2
node 2
S:CCCCC
NR:
level 3
higher
node
node 3
S:EEEEE
NR:
no record match in target fields on EEEEE

Figure 8

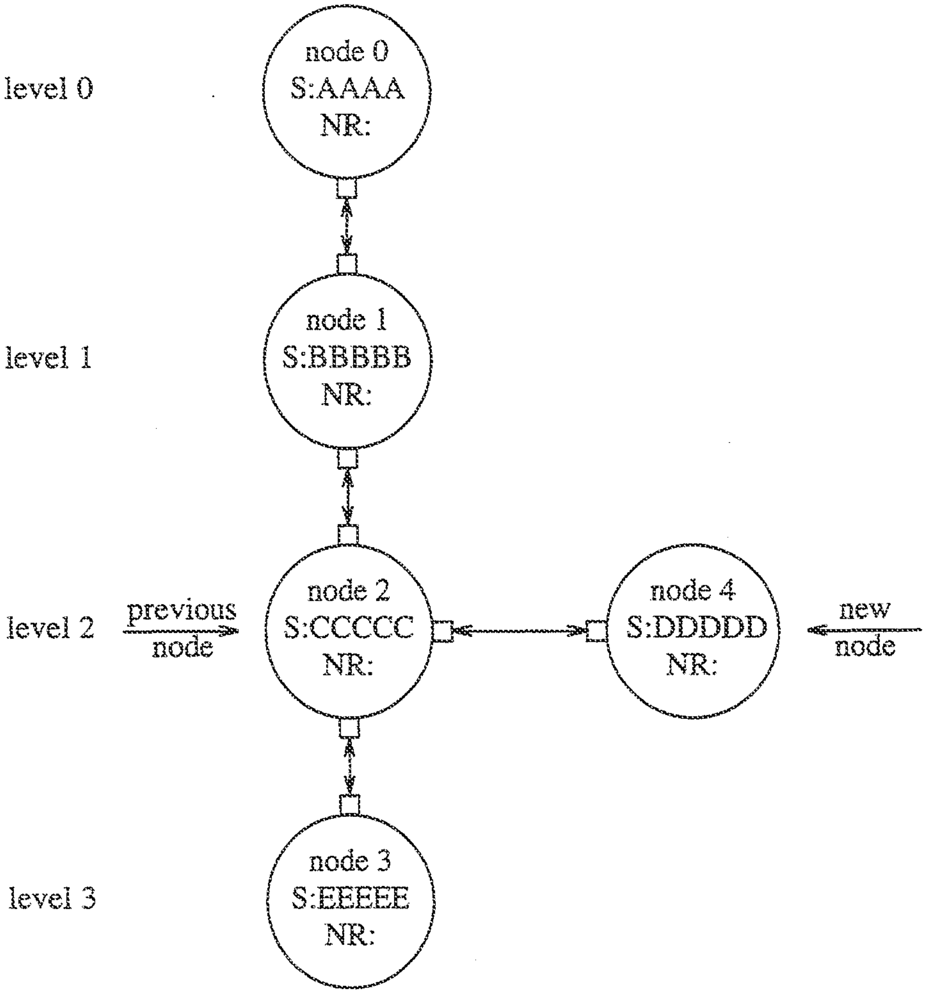

level 0
node 0
S:AAAA
NR:
level 1
node 1
S:BBBBB
NR:
level 2
previous
node
node 2
S:CCCCC
NR:
node 4
S:DDDDD
NR:
new
node
level 3
node 3
S:EEEEE
NR:

Figure 9

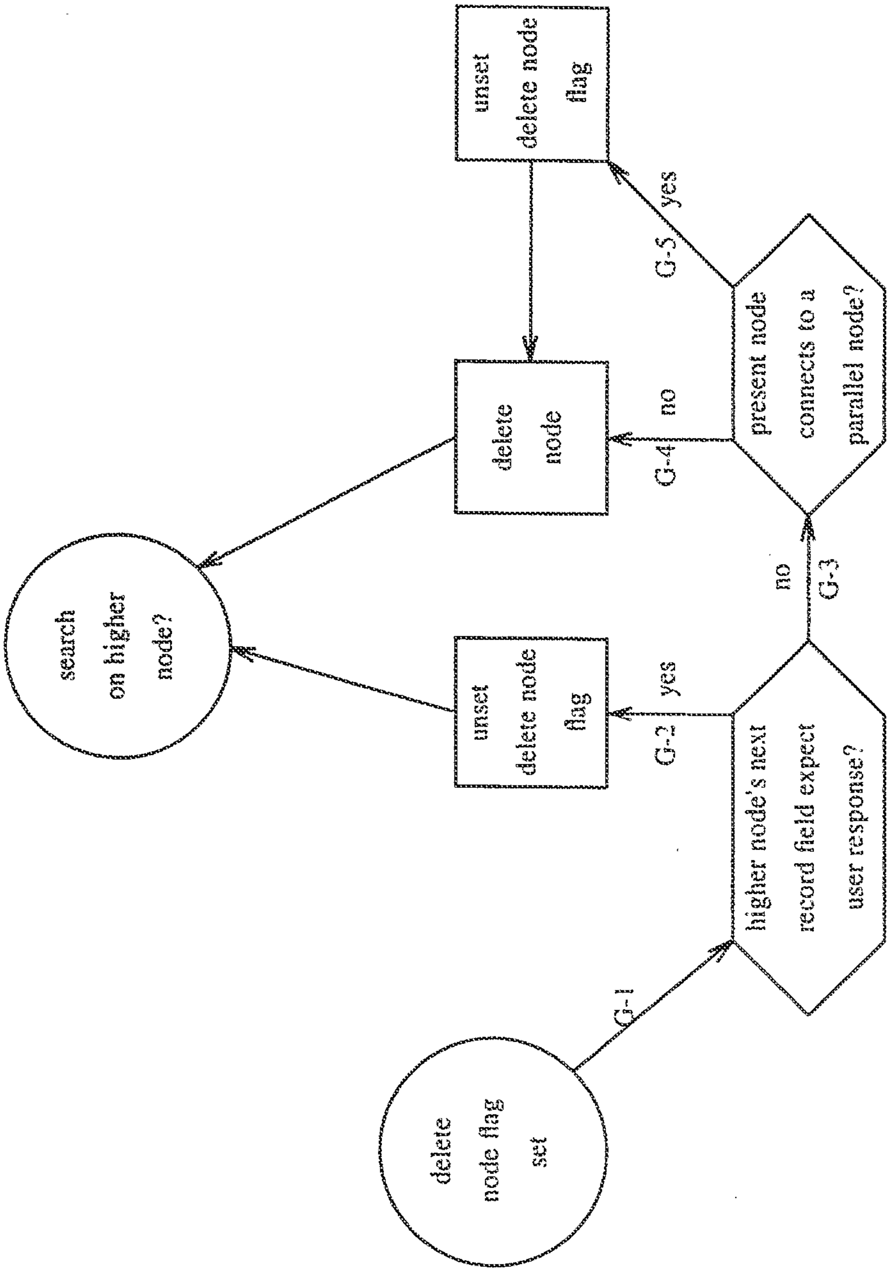

delete node flag set
G-1
higher node's next record field expect user response?
G-2
yes
unset delete node flag
G-3
no
present node connects to a parallel node?
G-4
no
delete node
G-5
yes
unset delete node flag
search on higher node?

Figure 10

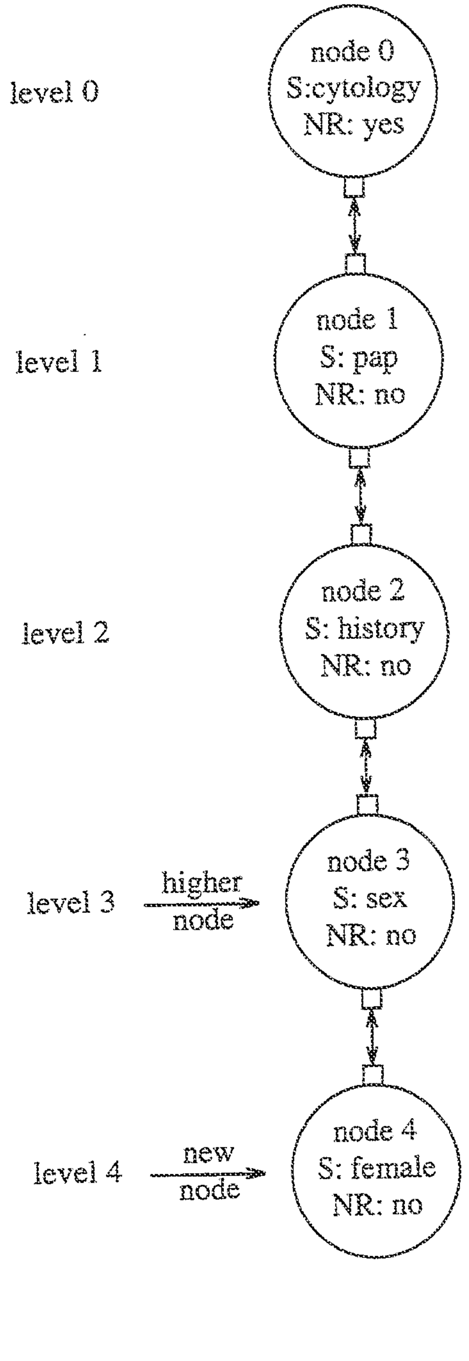

level 0
node 0
S:cytology
NR: yes
level 1
node 1
S: pap
NR: no
level 2
node 2
S: history
NR: no
level 3
higher
node
node 3
S: sex
NR: no
level 4
new
node
node 4
S: female
NR: no

Figure 11

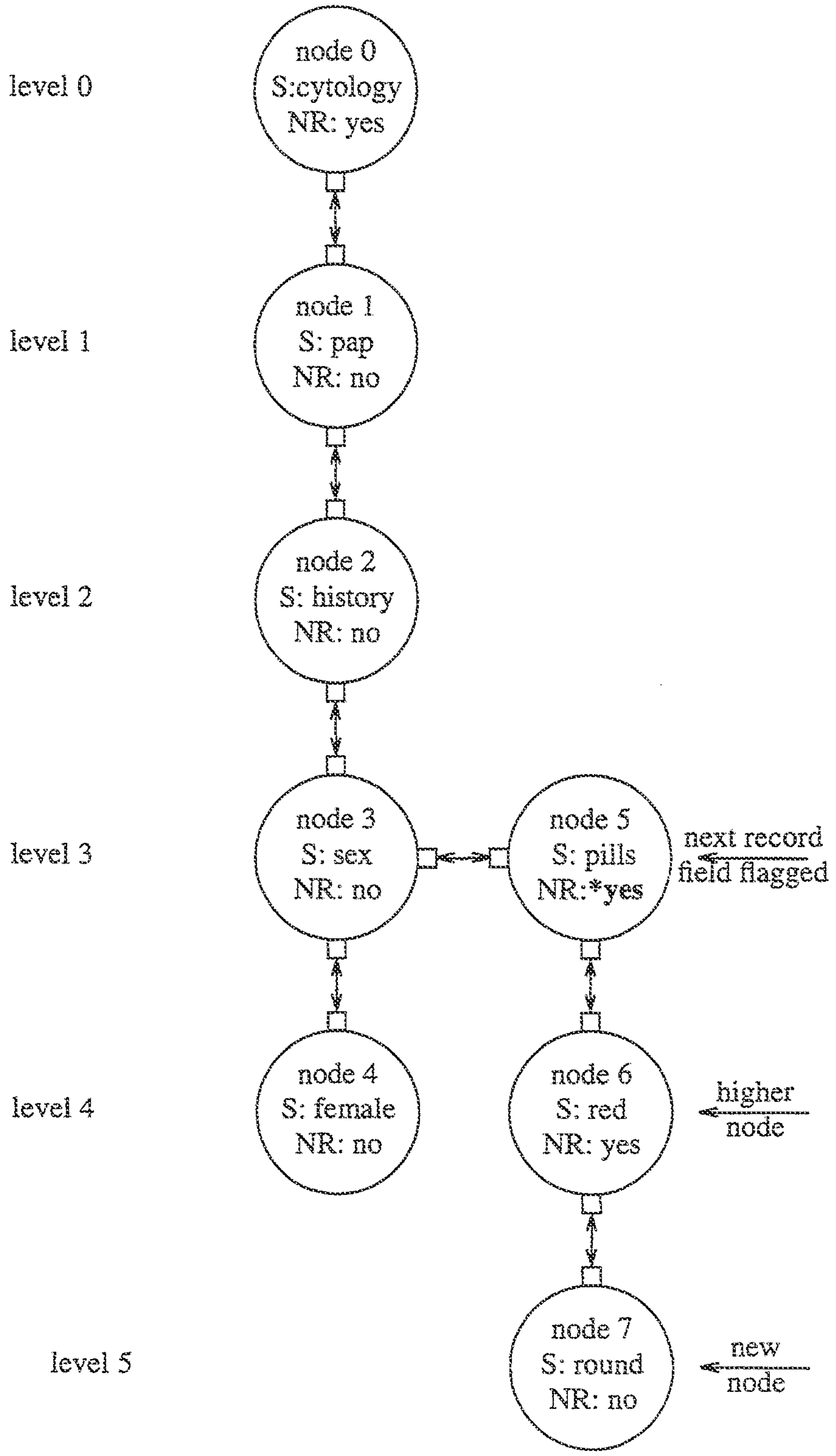

level 0
node 0
S:cytology
NR: yes
level 1
node 1
S: pap
NR: no
level 2
node 2
S: history
NR: no
level 3
node 3
S: sex
NR: no
node 5
S: pills
NR:*yes
next record
field flagged
level 4
node 4
S: female
NR: no
node 6
S: red
NR: yes
higher
node
level 5
node 7
S: round
NR: no
new
node

Figure 12

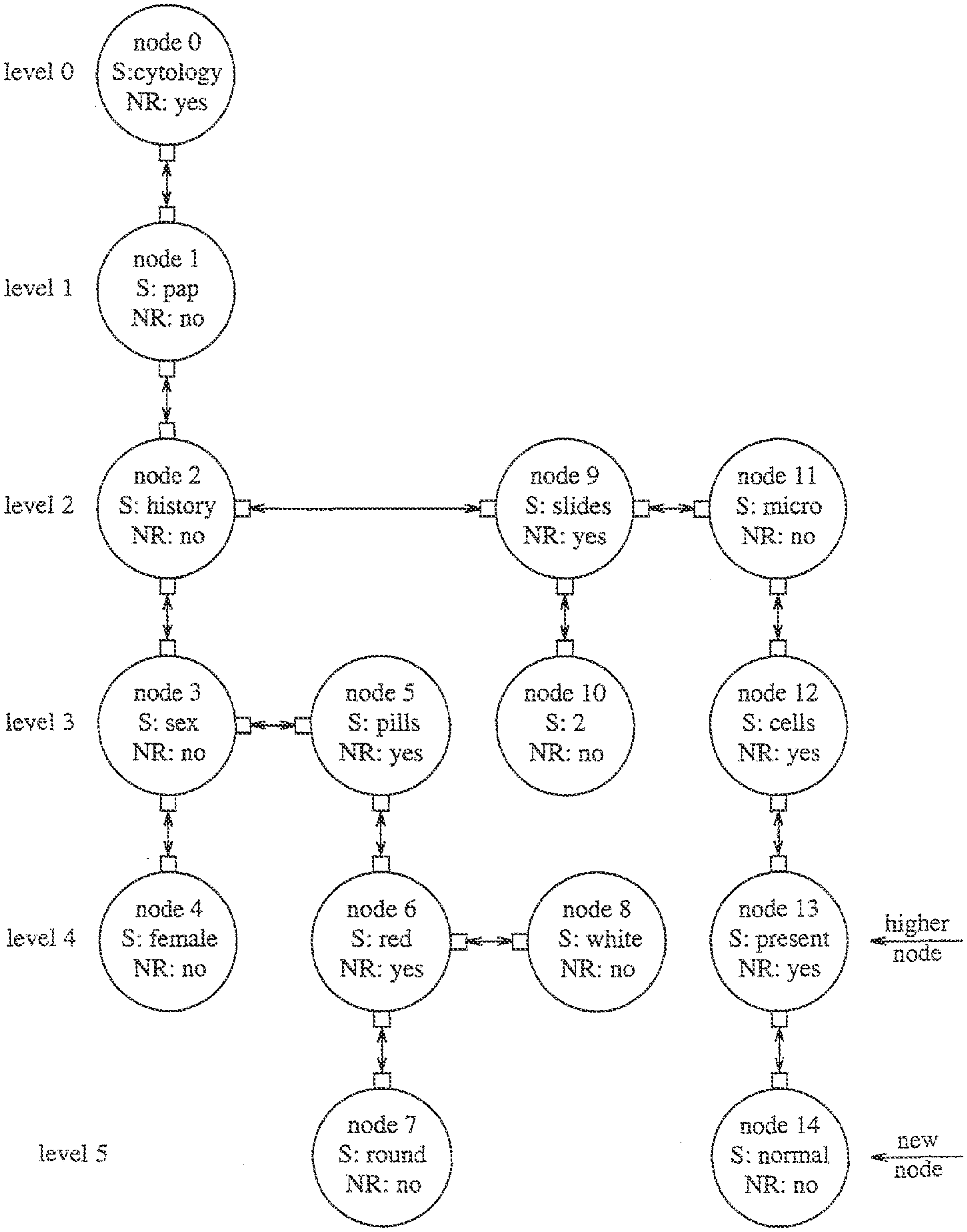

level 0
node 0
S:cytology
NR: yes
level 1
node 1
S: pap
NR: no
level 2
node 2
S: history
NR: no
node 9
S: slides
NR: yes
node 11
S: micro
NR: no
level 3
node 3
S: sex
NR: no
node 5
S: pills
NR: yes
node 10
S: 2
NR: no
node 12
S: cells
NR: yes
level 4
node 4
S: female
NR: no
node 6
S: red
NR: yes
node 8
S: white
NR: no
node 13
S: present
NR: yes
higher node
level 5
node 7
S: round
NR: no
node 14
S: normal
NR: no
new node

Figure 13

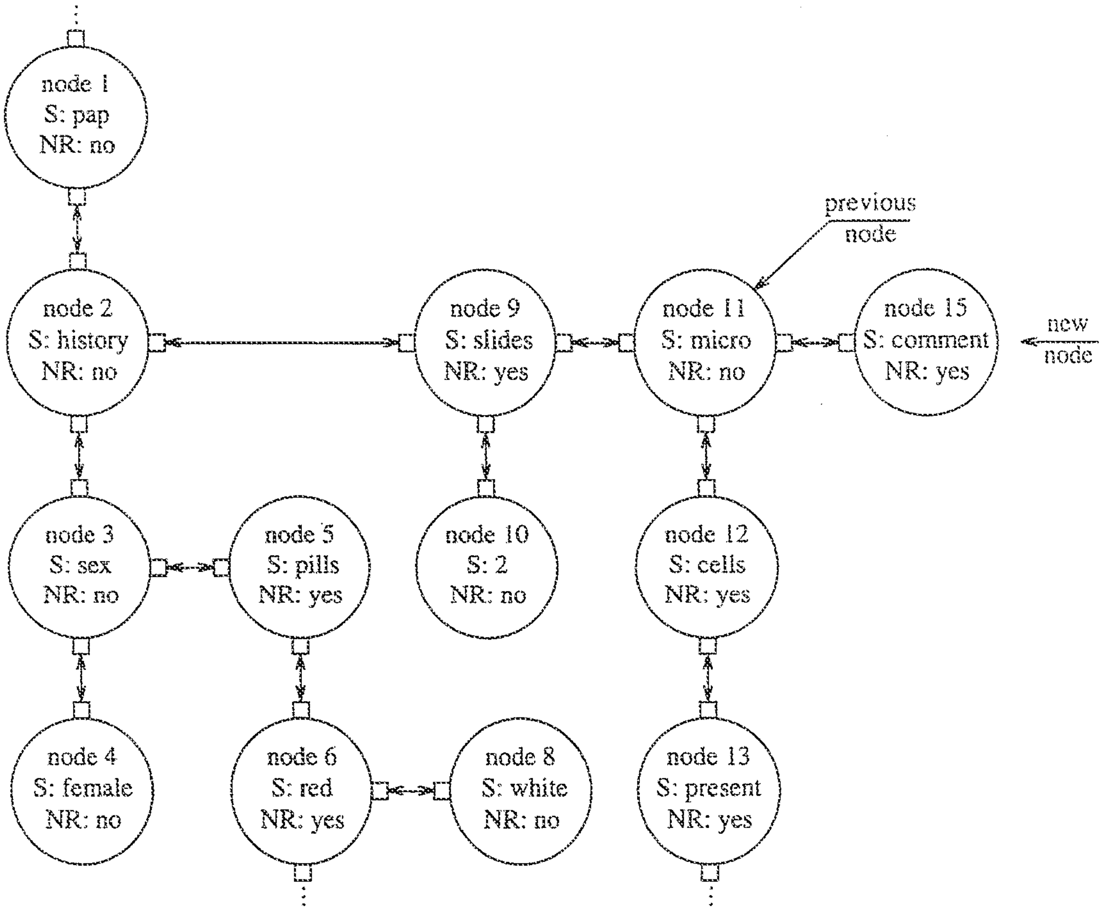

node 1
S: pap
NR: no
node 2
S: history
NR: no
node 9
S: slides
NR: yes
node 11
S: micro
NR: no
previous node
node 15
S: comment
NR: yes
new node
node 3
S: sex
NR: no
node 5
S: pills
NR: yes
node 10
S: 2
NR: no
node 12
S: cells
NR: yes
node 4
S: female
NR: no
node 6
S: red
NR: yes
node 8
S: white
NR: no
node 13
S: present
NR: yes

Figure 14

METHOD FOR RECORDING USER INTERACTION WITH A COMPUTER DATABASE TO GENERATE REPORTS

TECHNICAL FIELD

The present invention relates generally to systems for creating reports of complex data records, and more particularly to a system which utilizes a user interface interactive with a data base to build a hierarchical data tree.

BACKGROUND OF THE INVENTION

Computers have gone a long way towards increasing efficiency in many businesses. One particular area for which computers are especially well suited is the area of record keeping.

While there are a wide variety of data base record keeping systems currently on the market, they suffer several drawbacks. First, data base systems presently on the market are relatively inflexible in the format in which questions are presented to the user. The recording of data into a data base does not permit interaction between the user and the data base to customize the information stored by the data base. Rather, the user must either include all information from a particular field, or none of the information. Again, while conventional data bases permit recording of various collections of data from one or more different fields, they do not permit a user to create a report which permits user interaction in making choices, insertions, and the like.

SUMMARY OF THE INVENTION

It is therefore a general object of the present invention to provide an improved method for creating a data record in which greater flexibility is provided as to which questions are presented to the user and more control is allowed as to which responses are returned from the user.

Yet another object is to provide a method for building a hierarchical data tree based upon user interactivity with a data base.

These and other objects will be apparent to those skilled in the art.

The method for creating a hierarchical data tree of the present invention includes the initial step of providing a computer information processing system with a user interface, a data base and a system of rules which governs the interaction between the interface and the data base and determines a method of building a structured record of the interaction referred to as a hierarchical data tree. The user initially designates a target field and a next record field and the system searches the data base for records which have a source field matching the designated target field. The next record field of each record determines interaction with records matching the displayed record's target field. If a response is expected from the user, the user may either accept the record displayed or provide a separate user response as a selection. If no response is expected, according to the next record field, the system automatically selects the first matching record found. The system records the selection, whether from the system or from the user, utilizing a block of computer memory. The system then utilizes the recorded node as the basis for further search and further interaction. The system first creates a vertical leg of the data tree, before creating horizontal branches, according to the rules of interaction.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram of data base records utilized in the present invention;

FIG. 2 is a schematic diagram showing the hierarchical relationship of data base records;

FIG. 3 is a schematic diagram of a nodal structure for storage of interaction records;

FIG. 4 is a schematic view of a data tree constructed utilizing the nodal storage structure;

FIG. 5 is a flow chart showing rules of user interaction with the data base;

FIG. 6 is a flow chart showing interaction between the user and the data base;

FIG. 7 is a flow chart showing the rules provided for node construction and procedures;

FIG. 8 is a schematic diagram showing vertical node construction;

FIG. 9 is a schematic diagram showing horizontal node construction;

FIG. 10 is a flow chart showing the process of deleting a note;

FIG. 11 is a schematic diagram showing vertical node construction in a specific application;

FIG. 12 is a schematic diagram showing node construction beyond that of FIG. 12;

FIG. 13 is a schematic diagram showing nodal construction beyond that of FIG. 13; and

FIG. 14 is a schematic diagram showing the deletion of a node.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the drawings, the method of the present invention includes an interaction rule-based system consisting of a user interface which interacts with a data base to build a hierarchical data tree based upon the system responses.

The data base consists of multiple records each with a minimum of three fields: a target field "T", a source field "S", and a next record field "NR", as shown in FIG. 1.

The target field and source field are data buffers of identical type and size. Records are placed in the data base such that the source field in one record can be identical to a target field in another record. In this way, a search of the data base set based upon criteria placed in the source field of one record can find or match a different record, based upon information placed in that record's target field. The target and source fields can then act as a linked list between records such that the source field of one record points to one or more target fields in other records, which in turn have their own source fields which can point to other target fields, as shown in FIG. 2.

The next record field determines how the system will interact with the records matching the source field. In its simplest form, a user interaction or response is either "expected" or "not expected" in selecting the next record. If the next record field allows for a user response, the user may select one or more matching records or provide a separate user response to be used as the source field. If the next record field does not allow for a user response, all matching records will be automatically selected by the system in the order of their occurrence in the data base.

Information placed in the target and source fields determines the relationships between records. Every record may reference one or more additional records. The interface interacts between the data base and the user and constructs a record of the interaction.

A record of the interaction is stored as a series of nodes, as shown in FIG. 3. A node consists of a block of memory which references the source field and next record field of the selected record and contains references to other nodes selected before and after the present node. In this way a hierarchical pattern of responses emerges.

A record which contains a source field which matches another record's target field can be visualized as located on a higher vertical level to the record it refers to. Records which have a source field which match the target field of more than one record allow for branching in the nodal storage system. Records which have identical target fields are considered to be located on the same horizontal level. In this way nodes are "constructed" when records are selected, the nodes being located on levels which reflect the relationships of the records. In addition to referencing the source field and next record field, a two dimensional node contains references to the next higher level node, lower level node, previous node in the same level, and the next node in the same level.

The data tree can now be constructed as a branched linked list of nodes, as shown in FIG. 4. The nodal tree can be visualized as consisting of one or more vertical legs with horizontal branch points. Utilizing the system of the present invention, records can be generated describing complex data, guided by the entries in the data base, governed by the user responses, and an interface algorithm.

As mentioned above, a user response is either "expected" or "not expected". If a user response is not expected, then the system will automatically select all matching records in the order of their occurrence in the date base.

When the next record field indicates that a user response is "expected", the user may provide input in two different ways, as shown in FIG. 5. The user can either select input generated from the data base (A-1, A-2, and A-3) or provide its own input to be used by the system as though it were a source field provided by the data base (A-4, A-5, A-6).

Three responses deal with data base interactions. The user may accept the match provided by the search on the data base as the only response to the search (A-1). The user may accept the match, while informing the system to look for an additional match for consideration based on the same search criteria (A-2). Finally, the user may reject the match and search for a different match (A-3).

Three responses deal with input provided from the user to be used as the source field. The user may provide their own input to be used as though it were generated by a match in the data base and accept this input as the only response to the previous source field (A-4). The user may accept its own input to be used as the source field while is informing the system to look for additional responses under the previous source field (A-5). The user may provide an "empty" response (A-6).

In this way the user has the flexibility to use the data base to provide source fields for consideration, or input their own source field to be used by the system. This interaction between the user and the data base is central to how the system "constructs" nodes.

Referring now to FIG. 6, the initial step of the process is to conduct a search of the data base for target fields which match a chosen source field (B-1).

If the target field matches, the system unsets the "delete node" flag (described below) and looks at the next record field (B-2). If the target field matches and the next record field does not expect a user response, the system adds the node (B-3). If the target field matches and the next record field does expect a user response, a user response is prompted by the system (B-4), and the interaction rules of FIG. 5 are carried out. If no target fields match (B-1), the system looks at the next record field (B-5). If no target fields (B-1) match, and the next record field expects a user response, the system unsets a "delete node" flag (described below) and a user response is prompted by the system (B-6), and the interaction rules of FIG. 5 are carried out. If the target field does not match, and the next record field does not expect a user response, a node is not added (B-7).

The results of the user interaction of FIG. 5 are carried out as shown in FIG. 6.

When the user accepts a match from the data base or accepts the user's own input to be used as a source field, and informs the system that another match is needed (A-2, A-5), the system flags the next record field so that when examined, the system is aware that additional input is needed. When the user accepts their own response to be used as the source field , the value of the next record field to be recorded in the resulting node must also be determined. When the next record field is set to expect a user response, the system will add the node and search the data base using the provided source field as the search criteria for a match on the target fields in the data base (C-2(i)). When the next record field is set to not expect a response from the user, the node is added and the system moves to the "low" position of this node (C-2(ii)). The data base is not searched and this vertical leg of the node tree is terminated after adding this node. In cases where no response is generated when the next record field expects a response from the user, the system will not add a node (C-3). In addition, the system will set a "delete node" flag so that higher nodes in this vertical leg may also be deleted (as described in more detail hereinbelow).

The interface between the user and data base constructs the nodes in a linear fashion, completing each vertical leg before adding a horizontal branch. Once a vertical leg is completed, the system will climb to higher level nodes until it finds a branch point. If a branch is added, the vertical leg of this branch is completed before another branch is added.

Some general rules utilized in node construction are as follows (See FIG. 3 and 4).

The source field and next record field of a node always refer to nodes on the next lower level. When the system adds a node, and a node does not exist on that level, the added node is connected at its higher node register. If a node already exists on this level, the added node is connected at its previous node register, on the same horizontal level. Each node must be added to a higher node or previous node. A node cannot have both a higher node and a previous node connection.

For the initial selection in the construction of nodes (as shown in FIGS. 7 and 8), the system must be given a source field and next record field value (D-1). After input is provided for the source field, and the next record field is set to accept a single response, the system constructs a Node on level 0 as Node 0 (D-2). As there is no higher node, the Node 0's higher node reference register remains empty. The system then moves to the Node 0's low register and now considers Node 0 the higher node (D-3). If the next record field is set to expect a user response, the system will search the data base for a match in the target fields on the Node 0's source field (D-4). A node is either added or not added.

If the next record field is set to expect a user response, the user may determine the selection. If the next record field is set not to expect a user response, the selection is done automatically by the system. If a node is added, the system constructs a new Node on the next level below Node 0. The higher node (Node 0) references the new node (Node 1) as its lower node. The new node references the higher node as its higher node. This cycle continues until no more vertical nodes can be generated, as shown in FIG. 8.

When the system can no longer add nodes in a vertical fashion, the system will then climb to higher nodes looking for a branch point (E-1), as shown in FIG. 7.

The system moves to its higher node and then looks to see if a higher node exists. If a higher node exists, the system checks to see if a "delete node" flag is set (E-2). If the delete node flag is set, the system will consider deleting the node. If a higher node exists, the system determines whether an additional node may be added on this level. An additional node may be added on this level if the higher node's next record field does not allow for a user response or, if a user response is expected and when the selection was made, the user determined that an additional selection was needed by flagging the higher node's next record field (E-3). If an additional node is not to be added (E-4), the system climbs to its higher node and starts again at step E-1. If an additional node may be added and a node already exists on this level, the system moves to the present node's next node position. The system searches the data base on criteria of the higher nodes source field as in step D-4.

Referring now to FIGS. 7 and 9, horizontal node construction takes place only after vertical leg construction has terminated, and the system is searching for a branch point. As shown in FIG. 8, the system's search of the data base has found no record match in target fields selected in Node 3's source field, therefore the system moves upward a level to Node 2.

If a response is selected and a node already exists on this level (Node 2), the system builds a node on the same horizontal level. The Node 2 references the new Node 4 as its "next node", and is connected at the "next node" register of Node 2. Node 4 references Node 2 as its "preview node" register. The system moves to Node 4's lower node register and now searches the data base for a match in the target fields on Node's 4 source field and begins the vertical phase of node construction again at (F-2). When no more responses can be supplied, the system will have climbed to its highest node (Node 0) and terminates. If the "delete node" flag is set, the node is deleted before termination, as discussed hereinbelow.

When the system expects a user response and there is no response, some higher nodes may become unnecessary. Under these circumstances, the system will delete nodes as it climbs the node tree. In this way, unnecessary information is removed from the record.

As shown in FIGS. 7 and 10, if the delete node flag is set, the system checks the higher node's next record field (G-1). If the higher node's next record field expects a user response, the delete node flag is unset and the node is not deleted (G-2). The system checks the higher node's next record field to determine if a search on the data base on the higher node's source field is needed. If the higher node's next record field does not expect a user response, the system checks to see if the node is connected to a previous or next node on the same horizontal level (G-3). If the present note is not connected to a previous or next node, the system deletes the present node ;and then the system checks the higher nodes next record field to determine if a search on the data base on the higher node's source field is needed (G-4).

If the present node is connected to a previous or next node, the system unsets the delete node flag (G-5). The system deletes the present node and then the system checks the higher node's next record field to determine if a search on the data base on the higher node's source field is needed.

Utilizing the above-described rules of interaction, a practical application of the system is described hereinbelow.

Suppose a medical laboratory wishes to generate a series of reports concerning cytology examinations. In this laboratory they may examine cytology specimens from several anatomic sources, such a cervix (pap smear) or lung (sputum). With each pap smear examination the lab decides that specific information on each case should be reported. A short history consisting of the sex of the patient and what medication (pills, red or white) the patient is taking will be included. If the patient is taking red pills, the report will determine if they are square pills or round pills. The number of slides submitted will be recorded. A summary of microscopic findings will also be generated consisting of presence or absence of cells. When cells are present, these cells will be coded as either normal or abnormal. A comment will be placed at the end of the report if needed. The following entries should be made into the data base for this pap report:

TABLE 1

RECORD NUMBER	TARGET FIELD	SOURCE FIELD	NEXT RECORD FIELD
1	cytology	pap	no
2	cytology	sputum	no
3	pap	history	no
4	pap	slides	yes
5	pap	micro	no
6	history	sex	no
7	sex	female	no
8	history	pills	yes
9	pills	none	no
10	pilis	red	yes
11	red	round	no
12	red	square	no
13	pills	white	no
14	micro	cells	yes
15	cells	none	no
16	cells	present	yes
17	present	normal	no
18	present	abnormal	no
19	pap	comment	yes

An entry in the data base is made for a sputum report for the purpose of illustrating a choice of reports under the heading of cytology. To enter a sputum report, the contents of the report would need to be placed in the data base.

To begin the interaction, the user must select the criteria for an initial search. In this case, the user will provide the search criteria as "cytology". For the initial selection, a response is expected from the user.

As shown in FIG. 11, the system constructs a node in level 0, which we will refer to as node 0, which contains the user-generated source field "cytology" and the given next record field of "yes" denoting a user response is expected. The system moves to the lower node register (see FIG. 3) and considers this Node 0 to be the "higher" node.

The system now uses the source field of Node 0, "cytology", as the search criteria. A search of the data base on the target fields finds two matches, record #1 and record #2 (see Table 1). As a response is expected from the user, the user may choose between the two records' source fields, "pap" and "sputum".

If we select "pap", a new node is constructed on level 1, which we will refer to as node 1. The system records and references the source field, "pap", and next record field, "no" of the selected record. Node 1 places in its high node register the location of node 0. Node 0 places in its low node register the location of node 1.

The system now moves to the low node register of Node 1 and Node 1 becomes the higher node. The source field "pap" of Node 1 becomes the search criteria on target fields in the data base, and no response is expected from the user.

As no response is expected, the first match on a target field with "pap" as an entry is selected, record 3 (see Table 1). A new Node on level 2 is constructed and will be referred to as node 2. The source field of record #3, "history", and its next record field "no" are recorded and referenced by node 2. Node 2 places in its high node register the location of node 1. Node 1 places in its low node register the location of node 2.

The system now moves to the low node register of Node 2 and Node 2 becomes the higher node. The source field "history" becomes the search criteria on target fields in the data base, no response is expected from the user.

As node 2's next record field expects no response from the user, the first match in the data base in the target fields on the criteria "history" is selected, record #6 (see Table 1). Node 3 is constructed on level 3 and references the source field and next record field of record #6. Node 3 places in its high node register the location of node 2. Node 2 places in its low node register the location of node 3. Record #6 determines that no response is expected from the user on the next match.

Continuing to refer to FIG. 11, Node 3 becomes the higher node. The source field from Node 3 "sex", now becomes the search criteria. As no response is expected from the user, the first match on "sex" is automatically selected, record #7 (see Table 1). (As all pap smears must be obtained on females, there is no need for an alternate selection.) Node 4 is constructed and references the source field "female" and the next record field. Node 4 places in its high node register the location of node 3. Node 3 places in its low node register the location of node 4. Record #7 determines no response from the user is expected on the next match.

Node 4 becomes the higher node. The interface now searches for a match on the source field, "female", of node 4. Finding no match, the system looks at the next record field of its higher node, node 4. As the next record field of node 4 does not allow for a user response, the system moves to node 4's "next node" position and looks for a previously unused match in the target fields for "sex". Finding no matches, the system climbs to node 3 which becomes the "present" node. The system looks at the next record field of node 3's higher node, node 2. As the next record field of node 2 does not allow for a user response, the system moves to the "next node" position of node 3.

Referring now to FIG. 12, node 3 becomes the previous node. The system looks for a previously unused match in the target fields on node 2's source field "history". A match is found in record #8 (Table 1). Node 5 is now constructed on the same level, adjacent to node 3 and places in its "previous node" register the location of node 3. Node 3 now places in its "next node" register the location of node 5. Node 5 records record #8's source field "pills" and its next record field "yes".

Node 5 becomes the higher node. The system now searches for a match in the target fields on "pills" and finds the first match, record #9, "none" (Table 1). As the system expects a response from the user, the user may select this choice or look for another match. If the user rejects this choice, the next selection to be offered is record #10 "red". If we select record #10 while indicating to the system that another selection is needed, the system constructs node 6 in level 4 and references the source field and next record field of record #10. Node 6 places in its high node register the location of node 5. Node 5 places in its low node register the location of node 6. Record #10 determines that a response is expected from the user on the next match. The system will flag node 5's next record field so that the system will know that the user wishes to make another selection in addition to record #10 from the records in the data base which have target fields which match "pills".

Node 6 becomes the higher node. The system now searches for a match in the target fields of the data base for a match on "red". The source field of the first match, record #11, "round", is presented to the user. If the user selects this match, node 7 is constructed in level 5. Node 7 references record #11 's source field and next record field. Node 7 places in its high node register the location of node 6. Node 6 places in its low node register the location of node 7.

Node 7 becomes the higher node. The system now searches the data base in the target fields for a match on round. Finding no match, the system looks at the next record field of its higher node, node 6. As a response is expected from the user and an additional selection has not been indicated, the system climbs to node 6 and looks at the next record field of its higher node, node 5. As a response is expected from the user and an additional selection has been indicated, the system moves to the "next node" position of node 6 and node 6 becomes the previous node (see FIG. 13).

The system looks for the next match in the data base on node #5's source field, "pills". Record #11 follows record #10 and its source field "white" is offered to the user as a selection. If selected without wanting another match, node 8 is constructed in level 4. Node 8 places in its previous node register the location of node 6. Node 6 would place in its next node register the location of node 8. Record #11 determines that no response is expected from the user on the next match. The next record field of node 5 is no longer flagged, as the user does not need another selection.

Node 8 becomes the higher node. The interface now searches for a match on node 8's source field "white".

Finding no match, the system climbs to node 8 and looks at the next record field of its high node, node 5. As node 5's next record field allows for a user response, and is no longer flagged for an additional response, the system climbs to node 5 and looks at the next record field of its higher node, node 2. As node 2 does not allow for a user response, the system moves to the "next node" position of node 3.

Node 3 becomes the previous node. The system looks for a previously unused match in the target fields on node 2's source field "history". Finding no match, the system now climbs to node 2 and looks at the next record field of its higher node, node 1. As the next record field does not allow for a user response, the system moves to the next node position of node 2.

Node 2 becomes the "previous node". The system looks for a previously unused match in the target fields on node 1's source field "pap". A match is found in record #4 (Table 1). Node 9 is now constructed on the same level, adjacent to node 2 and places in its "previous node" register the location of node 2. Node 2 now places in its "next node" register the location of node 9. Node 9 records record 4's source field "slides" and its next record field "yes".

Node 9 becomes the higher node. The system now searches for a match in the target fields on "slides". Finding no match and as the system expects a response from the user, the system prompts the user for a response. The user may respond with its own input to be used as the source field. If the user inputs "2" and accepts this as the only response and assigns the value "no" to the next record field, the system constructs node 10 in level 3 and references "2" as the source field. Node 10 places in its high node register the location of node 9. Node 9 places in its low node register the location of node 10. As the user assigned the value of "no" to the next record field of node 10 the system will not search the data base on the response referenced by node 10.

Node 9 becomes the higher node. The system will look at the higher node's next record field to determine if another response is needed. As a response is expected from the user and the next record field was not flagged for an additional response, the system will climb to node 9 and look at the next record field of its higher node, node 1. As node 1 does not expect a response from the user, the system moves to the next record position of its present node, node 9.

Nose 9 becomes the previous node. The next unused match in the target fields in the data base on node 1's source field "pap" is selected, record #5. Node 11 is constructed in level 2 and references record #5's source field "micro" and next record field. Node 11 places in its "previous node" register the location of node 9. Node 9 places in its "next node" register the location of node 11.

Node 11 becomes the higher node. The system now searches for a match in the target fields on "micro" and finds a match, record #14 (Table 1). As the system does not expect a response from the user, the system constructs a node 12 on level 3 and references the source field of record #14, "cells" and its next record field "yes". Node 12 places in its high node register the location of node 11. Node 11 would place in its low node register the location of node 12. Record #14 determines that a response is expected from the user in selecting the next node.

Node 12 becomes the higher node. The system searches the target fields in the data base for a match on node 12's source field "cells". The first match, record #15 is presented to the user. If the user selects record #15, "none" is recorded and as there are no selections in the target fields which match "none", the system would begin to climb the nodes looking for another parallel match. If, however, the user selects record #16, "present", a match in the target fields of records in the data base can be found, and this leg of the node tree is continued. If we select record #16, "present" the system constructs node 13 in level 4 and references the source field and next record field of the record. Node 12 places in its low node register the location of node 13. Node 13 places in its high node register the location of node 12. Record #16 determines that a response is expected from the user on the next match.

Node 13 becomes the higher node. The source field from node 13 "present", now becomes the search criteria. There are two matches in the data base, record #17, "normal", and record #18, "abnormal". If the user selects "normal", Node 14 is constructed and references the source field "normal" and the next record field, "no". Node 14 places in its high node register the location of node 13. Node 13 places in its low node register the location of node 14. Record #17 determines no response from the user is expected on the next match.

Node 14 becomes the higher node. The interface now searches for a match on the source field of node 14, "normal". Finding no match, and as no response is expected, the system climbs to node 14 and looks at the next record field of node 13. As this node allows for a user response and the user has not requested an additional selection, the system now climbs to node 13 and looks at the next record field of node 12. As this node allows for a user response and the user has not requested an additional selection, the system now climbs to node 12 and looks at the next record field of node 11. As this node does not allow for a user response, the system moves to the next record position of node 11 and looks for a previously unused match in the target fields on "micro". Finding none, the system now climbs to node 11 and looks at the next record field of node 1. As this next record field does not allow for a user response, the system moves to the next record position of node 11. Node 11 becomes the previous node. The system then looks for a previously unused match in the target fields on "pap". A match is found in record #19 (Table 1). As shown in FIG. 14, node 15 is now constructed on the same level and adjacent to node 11 and places in its previous node register the location of node 11. Node 11 now places in its next node register the location of node 15. Node 15 records record #19's source field "comment" and its next record field "yes".

Node 15 becomes the higher node. The system searches for a match on node 15's source field "comment". Finding none and as the system expects a user response, the system prompts for a user response.

If the user provides an empty response, the system does not construct a node. The system sets the delete node flag and then climbs to its higher node, node 15. The system looks at the higher node's next record field.

As the higher node's next record field does not expect a user response, the system checks to see if the node is connected to a parallel node. As note #15 connects to a parallel node the delete node flag is unset.

The system deletes node 15 and node 11 becomes the previous node.

As the higher node's next record field does not expect a user response, the system looks for a previously unused match in the target fields on its higher node's source field "pap".

Finding none, the system now climbs to node 1 and looks at its higher node's next record field. As node 0 does allow for a user response and is not flagged for an additional selection, the system terminates.

The node tree of the final report no longer contains a reference to a comment. If the user wished to indicate that a comment was considered and not given, the user may have input "none" under comment and a node would have been constructed and this leg of the node tree would have been preserved.

Whereas the invention has been shown and described in connection with the preferred embodiment thereof, it will be understood that many modifications, substitutions and additions may be made which are within the intended broad scope of the appended claims.

I claim:

1. A method for recording user interaction with a computer data base to generate reports, comprising the steps of:

providing a computer information processing system, including a data base which interacts with user data input to build a hierarchical data tree based upon system responses;

a user initially designating a source and a "next record" field in said system; each said "next record" field comprising instructions to direct a response by the system upon identification of a record with the designated source field;

said system searching a plurality of records forming said data base for records which have a target field matching the inputted source field, and displaying an initial record found for viewing by the user;

the system reading the instructions of the initially designated "next record" field of the initial record and interacting with records matching the initial record's source field according to those instructions;

said system responding to a "response expected" instruction in a "next record" field by prompting the user for one of two responses; wherein the two responses consist of:

(1) requesting the user to select a record with a target field matching the displayed source field; and

(2) requesting the user to provide a separate response as a selection;

said system responding to a "no response expected" instruction in a "next record" field by finding all records with target fields matching the source field of the displayed record, and automatically selecting the first record found;

said system recording a selection from the user or system as a first node, utilizing a block of computer memory;

said computer system designating nodes in a predetermined hierarchical sequence, designating nodes with identical target fields as being on the same horizontal level, and designating a node with a source field matching a designated record's target field as being on a higher level than the designated record, each said node containing references to a previously selected node on the same level, the next node to be selected on the same level, the next higher level node and the next lower level node;

said system searching said data base for records which have a target field matching the source field of a second selected source field inputted by the user or system, and displaying a first matching record found for viewing by the user or system;

said system recording a selection of a displayed record by the user or the system, as a second node, on a lower vertical level than the first node;

continuing the steps of searching records, displaying a record, and recording selections, until no matching target fields are found for a selected record source field, at which time a first vertical leg of a hierarchical data tree is terminated; and

said system generating a report of all recorded nodes.

2. A method for recording user interaction with a computer data base to generate reports, comprising the steps of:

providing a computer information processing system, including a data base which interacts with user data input to build a hierarchical data tree based upon system responses;

a user initially designating a source and a "next record" field in said system;

each said "next record" field comprising instructions to direct a response by the system upon identification of a record with the designated a sought source field;

said system searching a plurality of records forming said data base for records which have a target field matching the inputted source field, and displaying an initial record found for viewing by the user;

the system reading the instructions of the initially designated "next record" field of the displayed record and interacting with records matching the displayed record's source field according to those instructions;

said system responding to a "response expected" instruction in a "next record" field by prompting the user for one of two responses; wherein the two responses consist of:

(1) requesting the user to select a record with a target field matching the displayed source field; and

(2) requesting the user to provide a separate user response as a selection;

said system responding to a "no response expected" instruction in a "next record" field by finding all records with target fields matching the source field of the displayed record, and automatically selecting the first record found;

said system recording a selection from the user or system as a first node, utilizing a block of computer memory;

said computer system designating nodes in a predetermined hierarchical sequence, designating nodes with identical target fields as being on the same horizontal level, and designating a node with a source field matching a designated record's target field as being on a higher level than the designated record, each said node containing references to a previously selected node on the same level, the next node to be selected on the same level, the next higher level node and the next lower level node;

said user selecting a first record displayed by the system, in response to a prompt from the computer system caused by a "response expected" instruction in the initially designated "next record" field;

said system searching said data base for records which have a target field matching the source field of a selected first record and displaying a first matching record found for viewing by the user;

said system recording a selection of a displayed record by the user or the system, as a second node, on a lower vertical level than the first node;

continuing the steps of searching records, displaying a record, and recording selections, until no matching target fields are found for a selected record source field, at which time a first vertical leg of a hierarchical data tree is terminated; and

said system generating a report of all recorded nodes.

3. A method for recording user interaction with a computer data base to generate reports, comprising the steps of:

providing a computer information processing system, including a data base which interacts with user data input to build a hierarchical data tree based upon system responses;

a user initially designating a source and a "next record" field in said system;

each said "next record" field comprising instructions to direct a response by the system upon identification of a record with the designated source field; said system searching a plurality of records forming said data base for records which have a target field matching the inputted source field, and displaying an initial record found for viewing by the user;

the system reading the instructions of the initially designated "next record" field of the displayed record and interacting with records matching the displayed record's source field according to those instructions;

said system responding to a "response expected" instruction in a "next record" field by prompting the user for one of two responses; wherein the two responses consist of:

(1) requesting a user to select a record with a target field matching the displayed source field and

(2) requesting a user to provide a separate user response as a selection;

said system responding to a "no response expected" instruction in a "next record" field by finding all records with target fields matching the source field of the displayed record, and automatically selecting the first record found;

said system recording a selection from the user or system as a first node, utilizing a block of computer memory;

said computer system designating nodes in a predetermined hierarchical sequence, designating nodes with identical target fields as being on the same horizontal level, and designating a node with a source field matching a designated record's target field as being on a higher level than the designated record, each said node containing references to a previously selected node on the same level, the next node to be selected on the same level, the next higher level node and the next lower level node;

said system searching said data base for records which have a target field matching the source field of a record automatically selected by the computer system caused by a "no response expected" instruction in the initially designated "next record" field, and displaying a first matching record found for viewing by the user;

said system recording a selection of a displayed record by the user or the system, as a second node, on a lower vertical level than the first node;

continuing the steps of searching records, displaying a record, and recording selections, until no matching source fields are found for a selected record source field, at which time a first vertical leg of a hierarchical data tree is terminated; and

said system generating a report of all recorded nodes.

4. The method of claim 1, 2 or 3, further comprising the steps of:

said system searching the next higher node above the last recorded node, after the step of terminating the first vertical leg of the data tree, and checking the "next record" field thereof;

said system searching the data base for a second record with a target field matching the source field of the next higher node, and said system moving vertically up a node level if no match is found;

continuing the steps of searching the next higher node, searching the data base for a second record, and moving vertically up a node level until all nodes in the vertical leg are searched.

5. The method of claims 1, 2 or 3, further comprising the steps of:

said system searching the next higher node, after the step of terminating the first vertical leg of the data tree, and checking the "next record" field thereof;

if the next higher node next record field calls for a user response, and an additional selection is needed, then said system searching the data base for a second record with a target field matching the source field of the next higher node and displaying a matching record to the user;

if the next higher node next record field calls for no user interaction, then said system searching the database for an unused record with a target field which matches the next higher node source field, and displaying a next match;

said user or system selecting the displayed record;

said system adding a node on the same horizontal level and designating the added node as horizontal node 1.

6. The method of claim 5, further comprising the steps of:

said system searching said data base for records which have a target field matching the source field of the horizontal node 1, and displaying a first matching record found for viewing by the user;

continuing the steps of recording selected records, searching records and displaying records until no matching source fields are found for a selected record source field at which time a vertical leg below the horizontal node is terminated.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 6993

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
90/012,829	04/03/2013 RULE	709	3992	20351.RX816

APPLICANTS

7822816, Residence Not Provided;
MACROSOLVE, INC.(OWNER), TULSA, OK;
NOVAK DRUCE CONNOLLY BOVE + QUIGG LLP(3RD.PTY.REQ.), HOUSTON, TX;
NDQ SPECIAL REEXAM GROUP, HOUSTON, TX

** **CONTINUING DATA** *************************

This application is a REX of 10/643,516 08/19/2003 PAT 7822816
which claims benefit of 60/404,491 08/19/2002

** **FOREIGN APPLICATIONS** ********************

		STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed	□ yes □ no			14	3
35 USC 119 (a-d) conditions met	□ yes □ no □ Met after Allowance				
Verified and Acknowledged	____________ Examiner's Signature Initials				

ADDRESS
22206

TITLE

SYSTEM AND METHOD FOR DATA MANAGEMENT

FILING FEE RECEIVED 12000	FEES: Authority has been given in Paper No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	□ All Fees □ 1.16 Fees (Filing) □ 1.17 Fees (Processing Ext. of time) □ 1.18 Fees (Issue) □ Other ______________ □ Credit