As described in further detail elsewhere, the specific application of the techniques comprising the present invention is, in one embodiment, for a treatment decision tree for a particular ailment. The decision tree is determined by a clinician. This tree is recorded in a computer database as a series of questions, which are asked to patients and their caregivers. Data is collected in much greater detail about each step of the treatment process than has ever been done before. Time based information is also collected. This gives not only the treatment, but also the exact sequence of events for the treatment.

The questionnaire and its answers are all stored. As the store becomes large, they may be analyzed for variance in treatment leading to more desired results (long term efficacy, reduced cost, etc.). The method for analysis chosen is a Genetic Algorithm (GA).

The mathematics of the Genetic Algorithm are based upon the study of complex non-linear systems. This field has come to be known as "Complexity Theory". Complexity theory discusses the phenomenon of "emergent systems" -- systems that are defined by very simple rules, but which exhibit extremely complex behavior because of the non-linear nature of the problem. Complexity theory itself arose out of the study of non-linear dynamical systems in a field, which has been called "chaos theory". There are several excellent references to these fields, including: "Chaos and Fractals, New Frontiers of Science" by Peitgen; "Applied Chaos Theory, A Paradigm for Complexity" by Cambel; "Complexity, Metaphors, Models, and Reality" by Cowan, among others.

Genetic programming has significant advantages over previous attacks on NP complete problems, yielding order(s) of magnitude faster convergence to local minima, with the added bonus that the system does not tend to settle on local minima but to continue to search the phase space of the problem for different solutions. Results of typical GA performance can be found in "Genetic Programming" and "Genetic Programming II" by Koza.

We begin by defining some terms:

Genotype – a string representation of the parameters that will define a solution to a particular problem. In this case, it is a representation of a Clinical Pathway along with its potential answers, which amounts to a single individual in the population of potential solutions.

Fitness Function – a method for evaluating genotypes against some criteria set for which the population is being optimized.

Mutation – random changes performed to the Genotype.

Population – a collection of genotypes.

Reproduction – the creation of new individuals by recombining the individual genes of different genotypes.

The problem of optimizing the performance of a clinical pathway versus different criteria has several unique features among genetic algorithms.

First, the exact set of criteria for which the pathway can be optimized is not known. This means that the fitness function has to be dynamic, as opposed to static as in standard GA approaches. The very dynamic nature of the fitness function suggests a two tiered GA approach to the solution. Not only are populations tested against specific criteria, they are tested against different criteria. The optimization not only looks at input data, but input data along with different fitness functions. The purpose is to search for existing but unsuspected correlations.

In a standard GA, the fitness function is fixed. Therefore, the system can only optimize towards those fixed criteria. In this system, not only can the subset of input (independent variables) be searched, but the subset of criteria (dependent variables) can

also be searched. This leads to a problem of representation: How is a genotype represented? One approach used in the present invention is as follows, as also described elsewhere:

Clinical pathways are stored as a series of questions. Each question can be thought of as a decision point within a decision tree, and each question is uniquely identified. Data is collected from the patient at each point within the decision tree that represents the care given that patient. Also, medical history is collected in the same way, as answers to pre-defined questions. Lastly, result information is collected in the same manner. This yields a huge phase space for the algorithm to search. The data must be represented in a manner such that simple manipulation of a string will yield results.

The representation chosen is one that consists of a string that represents the answers to questions. For example, if questions 1, 2 and 3 had possible answers of Y/N, A/B/C, D/E/F the genome would consist of 3 packets of information, the actual string (for instance YAD) and constraining information (Y/N, A/B/C, D/E/F). The string can then be mutated by random selection from available answers. This prevents mutation from producing meaningless answers.

This approach has an analog in genetics, where genes are constructed of simple proteins. Mutation can only select between the proteins to produce a new gene. The concepts are identical.

Note that dependent variables are represented in the same manner. The point of the search then is to find subsets of the problem data space that reflect a causal relationship to other subsets within the data space. Also note that this is a substantial departure from how traditional GAs are performed -- the fitness function is also represented in this manner. Questions that would be asked for fitness, i.e. Variable A > value Y, can also be represented in the same manner. In this case, the question would be a 3 part string, where the first part is the list of possible variables, the second is a list

of possible logical operators, the third is another list of possible variables. This allows the fitness function to be manipulated by the GA as well.

A summary of the differences between standard GA s and the present approach is provided below:

1) Dependent and independent variables are represented together in a single data space.
2) Fitness functions are represented within the same data space.
3) The object is to locate causal effects within the entire data space, as opposed to a narrowly defined fitness function.
4) There is no terminal condition that signals the end of the GA. The process continues indefinitely.

For this specific problem, the following methodology may be followed, described with respect to Figure 5. For purposes of the process described below, the reference numerals in [brackets] correspond to like-numbered reference numerals shown in the figure:

[5001] Create initial population – the population will initially be determined by "best guess" from human clinicians, along with the sets of independent, dependent, and fitness variables.
[5002] Initial population is evaluated.
[5003] Repeat the following steps:
[5004] Assign reproductive probabilities based upon individuals fitness
[5005] Form new genotypes with recombination and mutation
[5006] Evaluate the new population.
[5007] If high level of correlation is found report it.
[5008] Go to step 5003

There are several exceptions to the routine that must be allowed. For instance,

there may be factors that are always optimized. Cost would be a good example. If lower costs are desired, then those genes that define the fitness function that evaluate cost can be marked as being permanent. They will always be in the fitness function. This would be entered as "delta cost less than 0".

The second exception to the fitness rules are weighing. Fitness statements are simply taken as a point score on the basis of the number of true statements. For instance, if there are 10 fitness statements, if all 10 are true, then the genotype scores a 10. It is possible to weigh a statement. If there were a fitness statement that you wanted to defiantly select against, assign it a weight of negative infinity. An example of this would be increased mortality. The fitness statement, "delta mortality greater than 0" would be weighed as negative infinity, then if that case were true (mortality is higher) it wouldn't matter what the other fitness function values were, the genotype would be selected against.

As noted above, the data elements can represent a default clinical pathway and, thus, the system updates the default pathway to be presented to users in the future. Alternatively, the pathway can be the one currently being followed, where some input generates a correlation that causes a deviation or modification (or suggestion to take one) to the pathway.

In a further embodiment, the system includes means for storing the selected subsequent decision data element in the storage device. Suitable storage devices include diskettes, random access memory, or any other device capable of storing digital information.

In an enhanced version of the system, the clinical pathways in the database are each associated with a particular medical procedure. This can be done by offering a new database, or extending the clinical pathway database further to include a medical procedure data element corresponding to the initial and subsequent decision data elements for a particular medical procedure. As is evident, the system is readily suited

and optimal for tracking care which can be classified as stable acute care as that term is used herein.

In another embodiment, the processing means further comprises means for, prior to modifying the subsequent decision data elements within the clinical pathway database, querying the user for authorization to make the modification. For instance, if the user is at a computer terminal, a prompt can be given to the user describing the correlation that was found and requesting authorization to, *e.g.*, change the default clinical pathway presented to other users. Assuming the user is trusted or has valid access, the change can be made globally on behalf of all users of the system.

In a further embodiment, at least one of the subsequent decision data elements corresponds to an appropriate discharge of a patient. In this system, the processing means also includes means for comparing the selected subsequent decision data element with predetermined appropriate discharge criteria and, based upon the comparison, generating a signal corresponding to the appropriateness of discharging the patient. Such comparing means can be implemented, for example, using computer software in conjunction with the above described system. The signal can correspond to various states, including permissibility/advisability of discharge or lack thereof.

In another embodiment, the historical clinical pathway database further comprises a medical procedure data element corresponding to the initial and subsequent decision data elements for a particular medical procedure and wherein the processing means further comprises means for storing the medical procedure data element. In this system, the modified subsequent decision data elements are also stored within the historical clinical pathway database. The system includes means for correlating the modified subsequent decision data elements in the historical clinical pathway database with the medical procedure data element, means for querying the historical clinical pathway database and generating a signal corresponding to the subsequent decision data elements corresponding to a particular medical procedure, and 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.

In another embodiment, the clinical pathway database further comprises a patient identification data element corresponding to the initial and subsequent decision data elements for a particular patient. In this fashion, the clinical pathways can be associated with particular patients. The identification element can identify the patient directly, or can be an anonymous or random identifier if patient confidence is critical or desired.

In another embodiment, the clinical pathway database further comprises at least one patient visit data element corresponding or related to the patient identification data element. For settings such as stable acute care or home care, the clinical pathway events are generally in terms of "visits." Therefore, in this embodiment, the system allows correlations based upon patient visits and this data element becomes a part of the overall feedback system. In yet another embodiment, the clinical pathway database further comprises a time stamp data element corresponding to each of the at least one patient visit data elements, and the processing means further includes means for comparing the time stamp data element to predetermined criteria, and means for generating a signal corresponding to the result of the comparison and outputting the signal to a signal processing means. Thus, the system allows correlations based upon the visit duration (one example of a predetermined criteria generally of interest) or chronology.

In a further embodiment, the system includes a database for storing follow-up information and wherein the comparing means of the processing means is further responsive to the stored follow-up information in the follow-up information database. Where available, follow-up visit information is incorporated into the databases of the system and the system can use the follow-up information to determine correlations. For instance, if the patient's condition deteriorates months after the procedure, a correlation might be found in some treatment activity in the original clinical pathway. In essence, the pathway is continued forward an indefinite amount of time tracking the patient.

In addition, in one embodiment, the present invention provides a client / server system for manipulation and analysis of data related to clinical pathways. Referring now to Figure 4, one possible client / server configuration 400 useful for practicing the present invention is shown. The system accommodates an arbitrary number of physician or nurse clients 401. A major portion of the communication system of the invention is used to handle connections by physicians and nurses to the system. In one embodiment, data is converted internally to a more efficient format than the external standard HL7 protocol. All communications are coordinated by the server 402 of the system. When a physician or nurse requests information from a source 404 outside of the system, the information may be retrieved from an outside information server 403. Of course, both servers 402, 403 could be implemented on a single machine, if desired. The outside information server 403 contains a translation mechanism for handling translation to and from internal representations based on HL7. By interfacing with HL7, the system of the invention is capable of accessing patient information from existing HL7 external clients 404, as well as serving such information to HL7 systems requesting it.

Thus, the client / server system includes a communication network. In addition, the system includes at least one client workstation in communication with the communication network, where the client includes means (such as a modem or network adapter) for generating at least one signal corresponding to a clinical pathway decision and transmitting the at least one decision signal over the communication network. In addition, the client includes means (which can also be a modem or network adapter) for receiving at least one signal corresponding to a clinical pathway modification from the communication network. The client further includes means for outputting the at least one modification signal to a signal processing means (such as a monitor, printer, digital storage device, network connection, or further computing system). The system also includes a server on the communication network and the server includes (locally or remotely via appropriate connectivity) the 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. In addition, also associated with the server is a historical clinical pathway database (the two databases could, of course, exist on a single machine and, in fact, could have overlapping storage) for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element. Finally, the system includes processing means, in communication with the communication network, the client workstation, and the server, for performing various steps. The processing means is responsible for receiving the decision signal from the communication network and, based on the received decision signal, selecting one of the subsequent decision data elements. Then the processing means is responsible for 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 subsequent decision data elements within the clinical pathway database. Finally, the processing means is responsible for 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.

The present invention also provides a system for manipulation and analysis of 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 processing means, including a storage device, for performing the steps of selecting one of the 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 subsequent decision data elements within the clinical pathway database.

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. The system also includes 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. The system also includes processing means, optionally including a storage device, for selecting one of the subsequent decision data elements and comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database. Based upon predetermined correlation criteria, the system then modifies the subsequent decision data elements within the clinical pathway database. In addition, the system includes a statistical processing means, in communication with the clinical pathway database and the historical clinical pathway database, for accessing the historical clinical pathway database, computing a 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.

In an alternate embodiment, the system for determining optimal pathways is responsive to criteria determined to assess efficacy of the pathway. For instance, if the system is configured to optimize for three variables, *e.g.*, minimization of cost, time of stay, and post-op complications, then clinical pathways are simply ranked according to their overall efficacy under the stated criteria. Any clinical pathway that departs from a clinical pathway already resident in the database of the system can be annotated as

being untested relative to the criteria. As patient information is accumulated about the new criteria, it is evaluated along with all the other criteria already resident in the system. Because the system coordinates communications between a physician and attending nurse, in each case the information needed to track a clinical pathway's progress must necessarily pass through the system at which time the system can record and then or later analyze the path. The system is then able to determine all known pathways used to treat a particular problem and this data set can form one statistical database into which queries are made to determine efficacy.

The selection of criteria on which to base rankings may be entered along with the weightings of the criteria. For instance, using the three criteria above, these criteria can be manually ranked in the following order: least complications, shorter stay, lower cost. Any pathway which results in a higher incidence of complications will be ranked below any which have lower rates, even if they have lower cost because cost is not the primary criteria. In addition, the system itself can suggest ranking order, rather than relying on manual entry in all cases.

In addition, using the present system, as described above, it is possible to assign weights to the rankings as opposed to making them simple absolute rankings relative to each other. In this embodiment, it would be readily be possible for a weighted combination of least cost in conjunction with shorter stay to outweigh differences in complications.

In another embodiment, the invention provides a system for rating medical care based upon manipulation and analysis of data related to clinical pathways, including 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 elements corresponding to available subsequent decision points within the clinical pathway. The system also includes 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 rating value. In addition, the system includes processing means, including optionally a storage device, for selecting one of the 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 subsequent decision data elements within the clinical pathway database. The system further includes statistical processing means, in communication with the clinical pathway database and the historical clinical pathway database, for accessing the historical clinical pathway database, computing a pathway rating value based on the accessed rating values in the historical database, generating at least one signal corresponding to the pathway rating value, and outputting the at least one rating signal to a signal processing means.

With the present invention, the clinical pathway database of the system can, in one embodiment, be initialized with current proposed optimal clinical pathways. As a physician makes changes in the default pathway, a departure index is be created that is a measure of how far from the original pathway the physician is. In one embodiment, this index is calculated as a simple geometric distance from the current pathway. In such an embodiment, clinical pathways and their associated decision tree nodes are represented as points in n-space, thus making a distance function easy to compute. N-tuple representations of a particular clinical pathway decision tree also provide an exact identifier for that tree. No evaluation of the efficacy of the pathway can be made unless the particular departure is one that is already known to the system. Likewise, if a pathway is too different from existing pathways, then the system is configured to circumvent evaluations at all, even distance calculations, because the pathway is, in fact, in a different arbitrary space. In this case, new data for that clinical pathway decision tree is determined incrementally as new statistical data for the decision tree is collected over time.

For trees that are similar enough to be existing pathways known to the system, relative efficiency can, in one embodiment, be accomplished by a simple ranking

according to the efficacy criteria established for the system. For instance, this can be represented using a chart where column headings represent grading criteria, in order of importance, and relative efficiency in each category is reported relative to the optimal profile.

Similarly, in a further embodiment of the present invention, caregivers can be rated or ranked using, for instance, the final results of their contribution to tracking a patient through a particular clinical pathway. If a caregiver deviates from the optimal (default) pathway, yet still concludes with sufficiently successful outcome results, then the caregiver would be accorded a more positive rating. Again, ratings are based on the criteria as described above. In this embodiment, a simple ranking of patient data relative to the criteria is generated.

In a further embodiment, the historical database further stores the identity of the medical care provider determining the selection subsequent decision element and wherein the computing means is further responsive to the identity. In an alternate embodiment, the historical database further stores a rating for each of the historical clinical pathways in the database, and wherein the computing means is further responsive to the historical pathway ratings.

II. Data Inputting System Including Stable Acute Care System:

The improved input systems of the invention make possible a new type of care, stable acute care. The above data gathering and manipulation systems rely heavily upon access to a database of clinical pathways. Pursuant to the present invention, systems for inputting this information in a format suitable for the purposes described herein are also provided.

Using the present input systems, post-operative surgical procedures having any of following properties may be suitable candidates for stable acute care and prequalification for appropriate (early or late) post-operative hospital discharge. Such

patients include those who have been discharged very quickly and are in need of possible IV fluid, IV medication, or nursing care; patients having surgeries where there has been an abdominal incision with significant manipulation of the intestines which would have required a post-operative ileus in the past; and patients where it is desirable to discharge the patient sooner than is standard care (*e.g.*, as recited in Milliman & Robertson) in this country.

Stable acute care is appropriate for a variety of procedures. Stable acute care is made possible using the prequalification and communication systems of the present invention. For instance, using gynecological oncology as an example, radical hysterectomies, procedures with lymph node dissections, debulking procedures, procedures with gastrointestinal anastomoses, and even vaginal procedures may be suitable for stable acute care.

Regardless of the particular procedure being performed, in each case the initial history and physical are evaluated and entered into the system. Any conditions which would restrict a patient from receiving stable acute care are specifically determined for the patient. For instance, such conditions can include, but are not limited to, whether the patient has a 24 hour care giver, whether the patient's environment is inappropriate (for example, the patient has no electricity or no refrigerator), or whether the patient has particular medical problems or physical problems (absolute restrictive criteria would include patient with recent heart attack or a recent CVA). Once a patient is prequalified by the system as an appropriate candidate based on the procedure and the above noted restrictive criteria, the patient is then informed about the program. Assuming consent, the patient may then be seen by a nurse preferably capable of performing stable acute care who will educate them and give them other appropriate information.

After obtaining prequalification information and determining that the patient is suitable for receiving stable acute care, the system develops an appropriate set of orders for the specific surgery or procedure the patient is undergoing. The initial orders correspond to a default clinical pathway and, depending on the procedure, there may be

as few as only five or six different default orders per procedure such that the difference between post-operative and/or stable acute care for a radical hysterectomy and a mastectomy may be very minimal.

For use in the system, a questionnaire (again representing the default clinical pathway) is also developed specifically for each procedure. The questionnaire is later processed by the computer software to develop a SOAP note for the patient. When the patient is visited by the nurse, the SOAP note is generated by the computer for that visit and/or procedure. In a similar fashion, events such as IV hydration and IV meds and the use of H2 blockers such as TAGAMET® are also addressed for each procedure. Often, the decision elements or points which yield daily progress notes are similar from one procedure to another. Additionally, an order sheet may be created and may be modified by the physician or the nurse on a daily basis and the orders on the order sheet are determined from the initial order sheet that was created. In this fashion, the initial order sheet is essentially a clinical pathway printout, or at least a decision data element and subsequent data element printout.

In essence, the system creates a universal protocol template and, often, only minor changes are needed for each procedure for a specific plan. The data elements tracked in the model include, but are not required to include and are not limited to, appropriate procedures, restrictive criteria for patients, insurance information, pre-operative education, the clinical pathway, daily order sheets for subsequent orders, the questionnaire developed to determine specific notes, and daily progress sheets created to track the fluids, IV's, or medications given.

The patient information is collected in the doctors office during the first visit. Once the patient is known to be, for instance, an insurance candidate, the patient then receives appropriate information on a pre-operative visit. The patient is educated and given any appropriate information at that time. In the post-operative period in the hospital, once the patient appears to be in satisfactory condition to be released, the nursing care team is contacted and an arrangement is made for visits at the patient's

house or other stable acute care facility.

Criteria for discharge may be simple or complex. For instance, the patient for some procedures need only demonstrate adequate pain control and no significant medical or physical problems. As long as the patient is not demonstrating any nausea, vomiting or GI dysfunction, remains afebrile, and does not demonstrate any medical complications, the patient can be released to at least a stable acute care facility. Prior to the patient being released, the default clinical pathway is generated for the patient by the system and is used to instruct the nurse care giver as to how many times a day to see the patient and what kind care to administer to the patient. The nurse practitioner also supplies the appropriate information to the physician and communicates to the physician when a patient is ready to be released from the program.

Questionnaire Language grammar

The present invention includes data structures used to track clinical pathways. Because the pathways form a decision tree, the decisions at nodes can be analogized to questions. To assist in automating the present system, a special questionnaire language was developed for the syatem. The grammar and syntax of that system are described in the following paragraphs.

Items are entered into the database in the following form:

(text,{}) [(text, ,{ })...]

NOTATION:

Italics are literal strings.

Items enclosed in square brackets are optional.

Items enclosed in parenthesis are a selection (choose one of the list).

Ellipses indicate that you can repeat the previous grouped item.

Items enclosed in curly braces are format strings for field entry.

VALID OPERATOR LIST

Following is the list of valid operators:

Numeric

{INT, s [, [n], [m]] [:d] }

{FLOAT, x.y [, [n], [m]] [:d] }

Text

{CHAR, (s, "(x, 9, char)") [:d] }

{RADIO, (h, v) 9, "text")...[:d] }

{CHECK, (h, v) (, "test")...[:d] }

Date/Time

{TODAY}

{NOW}

{DATE, n[:d]}

{TIME, n[:d]}

Boolean

{BOOL,)n, { }) [:d{ }

{ERROR}

CONTROL

{IF (exp) , [{1}], [{2}] }

Storage

{MAKE varname AS vartype [stream]}

VALID OPERATOR DESCRIPTION

NUMERIC

{INT, s [, [n], [m]] [:d] }

Arguments:

s - size of input field

n - lower limit, n <= m

m - upper limit, m >= n

d - default value

Returns:

input integer value

Question Example:

"Enter Pulse : "{INT,4,0,} {IF{ERROR}, "Must be greater than 0"{REDO}, {NOOP} };

{FLOAT, X.Y], [n], [m]] [:d] }

Arguments:

x.y - format x - # of digits left of decimal, y - # of digits right of decimal

n - lower limit, n <= m

m - upper limit, m >= n

d - default value

Returns:

input floating point value

Question Example:

"Enter CBC count : "{FLOAT,1.2,0.0,5.0} {IF {ERROR}, "Must be between 0 and 5" {REDO} , {NOOP} };

TEXT

{CHAR, (s, "(x, 9, char)") [:d] }

Arguments:

s - size of input field

mask, x matches any character, 9 matches numbers, any other character is a literal.

d - default value

Returns:

string value of the input

Question Example:

"Enter SSN : "{CHAR,"999-99-9999"};

"Enter Name ; "{CHAR,40};

{RADIO, (h, v) (, "text")...[:d] }

Arguments:

h,v - horizontal or vertical presentation

"text" - items to be selected

d - default item number

Returns:

integer - corresponds to selected item, beginning with 0

Question Example:

"Select One : "{RADIO, h, "item 1", "item 2", "item 3":2};

This would display as:

Select One : item 1 ( ), item 2 ( ), item 3 (x)

This would return a 2, if item 1 were selected, item 3 would automatically de-select and the operation would return 0.

{CHECK, (h, v) (, "test")...[:d]}

Arguments:

h,v - horizontal or vertical presentation

"text" - items to be selected

d - default item number

Returns:

array of integers - corresponds to all selected items

Question Example:

"Which of the following do you have : "{CHECK, v, "diabetes", "hypertension", "hangnail"};

This would display as:

Which of the following do you have: **[ ] Diabetes**
[ ] Hypertension
[ ] Hangnail

DATE/TIME

{TODAY}

Returns:

integer - Julian number of today's date

{NOW}

Returns:

integer - integer time since midnight

{DATE, n[:d] }

Arguments:

n - number of format
1 - mm/dd/yy
2 - dd-mmm-yy
d - default

Returns:

integer - Julian number of the date

Question Example:

"Procedure Schedule time : "{TIME,2};

BOOLEAN

{BOOL, n["d] }

Arguments:

n - format

1 - yes/no

2 - true/false

d - default value,) or 1

Returns:

0 for false / negative

1 for true / positive

Question Example:

"All vital signs normal ? "{IF({BOOL, 1:1}) , {DEFAULT, {GROUP} }, {NOOP} }

{ERROR}

Returns:

0 for no errors

1 for errors exist

Question Example:

see INT example

CONTROL

{IF (exp) , [{1}], [{2}] }

Arguments:

(exp) - boolean expression, accepts <, >, <=, >=, ==, AND, OR NOT

Returns:

{NOOP}

Question Example:

If exp is true, 1 is performed. If exp is false, 2 is performed.

{DEFAULT, (screen id, tab id, group id, question id, field id) [, (screen id, tab id, group id, question id, field id] ...}

Arguments

5 tuple that specifies questions to trip default value on.

Returns:

{NOOP}

{ANSWER}

Returns:

Text of entire question after answer accepted

{NOOP}

Null operation.

{ERRMSG, TEXT}

Arguments:

test - Text to display in error status box.

Question Example:

See INT example

{SCREEN}

Returns:

Current screen id

{TAB}

Returns:

Current tab id

{GROUP}

Returns:

Current group id

{QUESTION}

Returns:

Current question id

{FIELD}

Returns:

Current field id

Storage

{MAKE (var) AS (vartype) [STREAM]}

Arguments:

Var – local variable name

Vartype – any language type listed above

STREAM – makes the variable storeable to disk

With reference to Figures 1A-1B, a sample daily routine for a caregiver using the system of the present invention is depicted. For purposes of Figure 1A, it will be assumed that the caregiver is a nurse or other medical professional who provides the care to a patient in the home, or other location remote from a primary care center (e.g., hospital, doctor's office, etc.). Of course, other suitable applications of the present invention may also be made, and the patient need not necessarily be in the home, but may also be in other locations (even in the hospital). Also, for purposes of the present discussion, it will be assumed that a client computer 401 (see Figure 4) may be used by either a nurse or by a physician. The mode of operation may be dictated by who logs into the particular computer 401. Of course, in another embodiment, two separate types

of client computers 401 could also be created, one for a physician and one for a nurse, etc.

In step 101, a nurse logs into a client computer 401. 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 modem and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link.

In steps 103-104, the nurse prepares for the visit to the patient (or a first patient, if more than one) by obtaining the necessary supplies, etc., and travels to the patient's location. 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.

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 non-compliance, the reasons are also stored. gain, all such stored data may later be transmitted back to the server 402.

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.

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 1B. 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. For purposes of the present discussion, a modem and telephone line connection will be assumed.

In step 124, if the modem connection was not successful, then in step 123 the user may be allowed to try the connection again, returning to step 122. If another attempt is not to be made, then step 128 is encountered, as described further below.

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.

In step 127, if the data has been correctly exchanged between the client computer 401 and server 402, then final step 129 is encountered. Otherwise, step 126 is encountered, where a decision is made whether to retry the transmission. If a retry is to

be attempted, then step 125 is performed again. Otherwise, step 128 is encountered. In step 128, an alert is set at the client computer 401, indicating that the transmission between the client computer 401 and server was unsuccessful, allowing the nurse to manually provide the data to the physician, or other personnel at the central location (e.g., via voice telephone, etc.).

Just as Figures 1A and 1B depict a remote caregiver's routine, Figures 2A-2C depict a sample routine performed by a physician. With reference to Figure 2A, in step 201, the physician logs into a client computer 401. In step 202, the physician may examine all patients for whom an alert has been generated (such as in certain of the steps of Figures 1A and 1B, described previously). The physician may thereafter selectively examine previously scheduled patients (step 203) and/or add new patients to be examined (step 204). At some point thereafter, the physician may log off of the client computer 401 in step 205.

Figure 2B depicts the patient examination step 203 of Figure 2A in further detail. In step 206, the examination process begins. Using the client computer 401, the physician may selectively review SOAP notes generated for the patient (step 207) and/or review the flow of care (FOC) generated for the patient (step 208). In step 209, the physician may review orders and questionnaire based on the SOAP notes and flow of care. Finally, in step 210, the physician is done with the present patient.

Step 209 of Figure 2B is described in further detail with respect to Figure 2C. In step 211, the review of order or questionnaire process begins. In step 212, the physician determines whether changes are required in the patient's flow of care, or in the questionnaire(s) to be used by the nurse. If changes are required, then the physician may modify the orders or the questionnaire as necessary in step 213. In step 214, such modifications are transmitted to the nurse's computer 401 using the process previously described with respect to Figure 1B. In step 215, if the transmission is successful, then step 216 is finally encountered. If not, then in step 217 a decision is made whether to retry the transmission. If so, step 214 is encountered again. Otherwise, an alert is set in

step 218, and step 216 is finally encountered.

Figures 3A and 3B depict in further detail the new patient addition step 204 of Figure 2A. In step 301, the process begins, and the patient is diagnosed with a particular ailment by a physician, etc. In step 302, a decision is made whether the patient is a candidate for a particular type of surgery, for example. If not, then step 303 is encountered, indicating that the process of Figure 3A and 3B is not necessarily applicable.

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.

In step 306, a decision is made whether the patient has any restrictive criteria, which would prevent the patient from being eligible for home health care. If so, then step 307 is encountered, and the process stops. Otherwise, in step 306 an initial visit is arranged. In step 309, an initial visit questionnaire is administered, and in step 310 a determination is made whether the initial visit is OK. Of not, then step 311 is encountered.

Otherwise, in step 312, a preop questionnaire is administered, and step 313 is encountered. In step 314, a determination is made whether the preop questionnaire is OK. If not, step 315 is encountered. Otherwise, in step 316 the patient is admitted to the hospital, and in step 317 the procedure is performed. In steps 318-319, a discharge questionnaire and orders are reviewed until OK, and in steps 320-321 the discharge questionnaire is administered until acceptable answers are obtained. In step 322, the home care provider is notified regarding the new patient, and in step 323 the patient becomes part of the daily routine of the physician and home care provider, previously

described.

The present invention provides for a very flexible data structure to be used for collecting data, as well as a relatively detailed amount of information to be collected about patients and their progress through a clinical pathway. This data format is required for purposes of optimizing the pathways and procedures, as previously described. Because of the flexible data structure allowed by the present invention, the present invention has the ability to produce custom reports. These reports can easily be tailored to present information in any format desired. Examples of this might be productions of the standard "Home Health Certification and Plan of Care" form HCFA-485 and the HCFA-487 form, which is an addendum to the plan of treatment or a medical update.

Both of these forms can be constructed from a subset of the data required by the present invention. Home health agencies, Medicare and many private insurance plans use the HCFA-485 form for reimbursement of services. Changes to orders and the patient's condition are reported on the HCFA-487 form. Both of these forms are used to track a patient's progress through some treatment plan. Within the present invention, this treatment plan corresponds to a clinical pathway. Progress may be noted in the present invention by recording visits to the patient, and the patient's actual condition. Changes made to orders are recorded as changes to the pathway. As a result of recording the information needed by the system of the present invention, it would be easy to produce reports, in whatever format needed, to demonstrate compliance with various regulatory or insurance requirements.

It will be apparent to those skilled in the art that various modifications and variations can be made in the present invention without departing from the scope or spirit of the invention. Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with the true scope and spirit of the invention being indicated by the

following claims.

What is claimed is:

1. A system for analyzing health care data, comprising:
   (a) a database comprising a plurality of records, each record including:
      (1) event data corresponding to a health-related event for a person,
      (2) type data corresponding to the health-related event type,
      (3) chronological data corresponding to a relative point in time that the health-related event occurred;
   (b) processing means for performing the steps of:
      (1) analyzing the database in order to identify correlations between the plurality of event data within the plurality of records, based upon the type data and chronological data in each record;
      (2) modifying a selected series of records within the database in order to maximize the likelihood of occurrence of a desired health-related event having chronological data corresponding to a point in time later than the chronological data associated with the selected series of records.

2. The system of claim 1, wherein the event data corresponds to a decision point among the plurality of records.

3. The system of claim 1, wherein the selected series of records corresponds to a clinical pathway.

4. The system of claim 3, wherein the clinical pathway corresponds to a medical procedure performed on the person.

5. The system of claim 1, wherein the analyzing step is performed using a genetic algorithm.

6. The system of claim 1, wherein the analyzing step is performed using actuarial analysis techniques.

7. A system for analyzing health care data, comprising:

(a) an output device for providing output to a user;

(b) a database comprising a plurality of records, each record including:

(1) event data corresponding to a health-related event for a person,

(2) type data corresponding to the health-related event type,

(3) chronological data corresponding to a relative point in time that the health-related event occurred;

(c) processing means for performing the steps of:

(1) analyzing the database in order to identify correlations between the plurality of event data within the plurality of records, based upon the type data and chronological data in each record;

(2) identifying a selected series of records within the database that may be modified in order to maximize the likelihood of occurrence of a desired health-related event having chronological data corresponding to a point in time later than the chronological data associated with the selected series of records; and

(3) outputting to the user an identifier associated with the selected series of records.

8. The system of claim 7, wherein the processing means further performs the steps of:

(4) prompting the user whether the selected series of records should be modified; and

(5) modifying the selected series of records.

9. A process for analyzing health care data, comprising the steps of:

(a) storing a plurality of records, each record including:

(1) event data corresponding to a health-related event for a person,

(2) type data corresponding to the health-related event type,

(3) chronological data corresponding to a relative point in time that the health-related event occurred;

(b) analyzing the plurality of records to identify correlations between the plurality of event data within the plurality of records, based upon the type data and chronological data in each record;

(c) modifying a selected series of records in order to maximize the likelihood of occurrence of a desired health-related event having chronological data corresponding to a point in time later than the chronological data associated with the selected series of records.

10. The process of claim 9, wherein the analyzing step is performed using actuarial analysis techniques.

11. A system for manipulation and analysis of data related to clinical pathways, comprising:

(a) a clinical pathway database for storing:

(i) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(ii) at least one subsequent decision data element, corresponding to an available subsequent decision point within the clinical pathway;

(b) a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(c) processing means, including a storage device, for performing the steps of:

(i) selecting one of the at least one subsequent decision data elements;

(ii) comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored

in the historical clinical pathway database; and

(iii) based upon predetermined correlation criteria, modifying the at least one subsequent decision data element within the clinical pathway database.

12. The system of claim 11, further comprising means for storing the selected subsequent decision data element in the storage device.

13. The system of claim 11, wherein the clinical pathway database subsequent decision data elements further comprise default subsequent decision data elements and wherein the processing means modifies the default subsequent decision data elements.

14. The system of claim 11, wherein the clinical pathway database further comprises a medical procedure data element corresponding to the initial and subsequent decision data elements for a particular medical procedure.

15. The system of claim 14, wherein the medical procedure corresponds to the stable acute care of a patient.

16. The system of claim 11, wherein the processing means further comprises means for, prior to modifying the at least one subsequent decision data element within the clinical pathway database, querying the user for authorization to make the modification.

17. The system of claim 11, wherein at least one of the at least one subsequent decision data elements corresponds to an appropriate discharge of a patient and wherein the processing means further comprises means for comparing the selected subsequent decision data element with predetermined appropriate discharge criteria and, based upon the comparison, generating a signal corresponding to the appropriateness of discharging the patient.

18. The system of claim 14, wherein the historical clinical pathway database further comprises a medical procedure data element corresponding to the initial and subsequent decision data elements for a particular medical procedure and wherein the processing means further comprises:

(iv) means for storing the medical procedure data element, and the modified at least one subsequent decision data element within the historical clinical pathway database;

(v) means for correlating the modified at least one subsequent decision data elements in the historical clinical pathway database with the medical procedure data element;

(vi) means for querying the historical clinical pathway database and generating a signal corresponding to the at least subsequent decision data element corresponding to a particular medical procedure; and

(vii) means for outputting the signal to a signal processing means.

19. The system of claim 11, wherein the clinical pathway database further comprises a patient identification data element corresponding to the initial and subsequent decision data elements for a particular patient.

20. The system of claim 19, wherein the clinical pathway database further comprises at least one patient visit data element corresponding to the patient identification data element.

21. The system of claim 20, wherein the clinical pathway database further comprises a time stamp data element corresponding to each of the a least one patient visit data elements, and wherein the processing means further comprises:

(ix) means for comparing the time stamp data element to predetermined criteria; and

(x) means for generating a signal corresponding to the result of the

comparison and outputting the signal to a signal processing means.

22. The system of claim 11, further comprising a database for storing follow-up information and wherein the comparing means of the processing means is further responsive to the stored follow-up information in the follow-up information database.

23. A client / server system for manipulation and analysis of data related to clinical pathways, comprising:

(a) a communication network;

(b) a client workstation in communication with the communication network, wherein the client workstation comprises:

(i) 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

(ii) means for receiving at least one signal corresponding to a clinical pathway modification from the communication network; and

(iii) means for outputting the at least one modification signal to a signal processing means;

(c) a server on the communication network, wherein the server comprises:

(i) a clinical pathway database for storing:

(1) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(2) at least one subsequent decision data element, corresponding to at least one available subsequent decision point within the clinical pathway; and

(ii) a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(d) processing means, in communication with the communication network,

the client workstation, and the server, for performing the steps of:

(1) receiving the at least one decision signal from the communication network;

(2) based on the received decision signal, selecting one of the at least one subsequent decision data elements;

(3) comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database;

(4) based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database;

(5) generating at least one signal corresponding to a clinical pathway modification of the subsequent decision data elements in the clinical pathway database; and

(6) transmitting the at least one clinical pathway modification signal over the communication network to the receiving means of the client workstation.

24. A system for manipulation and analysis of data related to clinical pathways, comprising:

(a) a clinical pathway database for storing:

(i) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(ii) at least one subsequent decision data element corresponding to at least one available subsequent decision point within the clinical pathway;

(b) a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(c) processing means, including a storage device, for performing the steps of:

(i) selecting one of the at least one subsequent decision data elements;

(ii) comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database; and

(iii) based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database.

25. A system for assessing utilization of medical resources based upon manipulation and analysis of statistical data related to clinical pathways, comprising:

(a) a clinical pathway database for storing:

(i) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(ii) at least one subsequent decision data element corresponding to at least one available subsequent decision points within the clinical pathway;

(b) a historical clinical pathway database for storing:

(i) previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(ii) for each of the previously selected subsequent decision data elements, a utilization value corresponding to the decision data element;

(c) processing means, including a storage device, for performing the steps of:

(i) selecting one of the at least one subsequent decision data elements;

(ii) comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database; and

(iii) based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database; and

(d) statistical processing means, in communication with the clinical pathway database and the historical clinical pathway database, for performing the steps of:

(i) accessing the historical clinical pathway database;

(ii) computing pathway utilization value based on the accessed utilization values in the database;

(iii) generating at least one signal corresponding to the pathway utilization value; and

(iv) outputting the at least one utilization value signal to a signal processing means.

26. A system for rating medical care based upon manipulation and analysis of data related to clinical pathways, comprising:

(a) a clinical pathway database for storing:

(i) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(ii) at least one subsequent decision data elements, corresponding to available subsequent decision points within the clinical pathway;

(b) a historical clinical pathway database for storing:

(i) previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(ii) for each of the previously selected subsequent decision data elements, a rating value;

(c) processing means, including a storage device, for performing the steps of:

(i) selecting one of the at least one subsequent decision data elements;

(ii) comparing the selected subsequent decision data element with

the previously selected subsequent decision data elements stored in the historical clinical pathway database; and

(iii) based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database; and

(d) statistical processing means, in communication with the clinical pathway database and the historical clinical pathway database, for performing the steps of:

(i) accessing the historical clinical pathway database;

(ii) computing a pathway rating value based on the accessed rating values in the historical database;

(iii) generating at least one signal corresponding to the pathway rating value; and

(iv) outputting the at least one rating signal to a signal processing means.

27. The system of claim 26, wherein the historical database further stores the identity of the medical care provider determining the selection subsequent decision element and wherein the computing means is further responsive to the identity.

28. The system of claim 26, wherein the historical database further stores a rating for each of the historical clinical pathways in the database, and wherein the computing means is further responsive to the historical pathway ratings.

29. A method for manipulation and analysis of data related to clinical pathways, the method comprising the steps of:

(a) providing a clinical pathway database for storing:

(i) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(ii) at least one subsequent decision data element, corresponding to

an available subsequent decision point within the clinical pathway;

(b) providing a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element;

(c) selecting one of the at least one subsequent decision data elements;

(d) comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database; and

(e) based upon predetermined correlation criteria, modifying the at least one subsequent decision data element within the clinical pathway database.

30. The method of claim 29, further comprising storing the selected subsequent decision data element in a storage device.

31. The method of claim 29, wherein the clinical pathway database subsequent decision data elements further comprise default subsequent decision data elements and wherein the method further comprises modifying the default subsequent decision data elements.

32. The method of claim 29, wherein the clinical pathway database further comprises a medical procedure data element corresponding to the initial and subsequent decision data elements for a particular medical procedure.

33. The method of claim 32, wherein the medical procedure corresponds to the stable acute care of a patient.

34. The method of claim 29, further comprising, prior to modifying the at least one subsequent decision data element within the clinical pathway database, querying the user for authorization to make the modification.

35. The method of claim 29, wherein at least one of the at least one subsequent decision data elements corresponds to an appropriate discharge of a patient and wherein the method further comprises comparing the selected subsequent decision data element with predetermined appropriate discharge criteria and, based upon the comparison, generating a signal corresponding to the appropriateness of discharging the patient.

36. The method of claim 32, wherein the historical clinical pathway database further comprises a medical procedure data element corresponding to the initial and subsequent decision data elements for a particular medical procedure and wherein the method further comprises:

(f) storing the medical procedure data element, and the modified at least one subsequent decision data element within the historical clinical pathway database;

(g) correlating the modified at least one subsequent decision data elements in the historical clinical pathway database with the medical procedure data element;

(h) querying the historical clinical pathway database and generating a signal corresponding to the at least subsequent decision data element corresponding to a particular medical procedure; and

(i) outputting the signal to a signal processing means.

37. The method of claim 29, wherein the clinical pathway database further comprises a patient identification data element corresponding to the initial and subsequent decision data elements for a particular patient.

38. The method of claim 37, wherein the clinical pathway database further comprises at least one patient visit data element corresponding to the patient identification data element.

39. The method of claim 38, wherein the clinical pathway database further

comprises a time stamp data element corresponding to each of the a least one patient visit data elements, and wherein the method further comprises:

(f) comparing the time stamp data element to predetermined criteria; and

(g) generating a signal corresponding to the result of the comparison and outputting the signal to a signal processing means.

40. The method of claim 29, further comprising a database for storing follow-up information, wherein the comparing step is further responsive to the stored follow-up information in the follow-up information database.

41. A client / server method for manipulation and analysis of data related to clinical pathways, the method comprising the steps of:

(a) providing a communication network;

(b) providing a client workstation in communication with the communication network, wherein the client workstation comprises:

(i) 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

(ii) means for receiving at least one signal corresponding to a clinical pathway modification from the communication network; and

(iii) means for outputting the at least one modification signal to a signal processing means;

(c) providing a server on the communication network, wherein the server comprises:

(i) a clinical pathway database for storing:

(1) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(2) at least one subsequent decision data element, corresponding to at least one available subsequent decision point within the clinical pathway; and

(ii) a historical clinical pathway database for storing previously

selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(d) receiving the at least one decision signal from the communication network;

(e) based on the received decision signal, selecting one of the at least one subsequent decision data elements;

(f) comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database;

(g) based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database;

(h) generating at least one signal corresponding to a clinical pathway modification of the subsequent decision data elements in the clinical pathway database; and

(i) transmitting the at least one clinical pathway modification signal over the communication network to the receiving means of the client workstation.

42. A method for manipulation and analysis of data related to clinical pathways, the method comprising the steps of:

(a) providing a clinical pathway database for storing:

(i) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(ii) at least one subsequent decision data element corresponding to at least one available subsequent decision point within the clinical pathway;

(b) providing a historical clinical pathway database for storing previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(c) selecting one of the at least one subsequent decision data elements;

(d) comparing the selected subsequent decision data element with the

previously selected subsequent decision data elements stored in the historical clinical pathway database; and

(e) based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database.

43. A method for assessing utilization of medical resources based upon manipulation and analysis of statistical data related to clinical pathways, the method comprising the steps of:

(a) providing a clinical pathway database for storing:

(i) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(ii) at least one subsequent decision data element corresponding to at least one available subsequent decision points within the clinical pathway;

(b) providing a historical clinical pathway database for storing:

(i) previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(ii) for each of the previously selected subsequent decision data elements, a utilization value corresponding to the decision data element;

(c) selecting one of the at least one subsequent decision data elements;

(d) comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database;

(e) based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database;

(f) accessing the historical clinical pathway database;

(g) computing pathway utilization value based on the accessed utilization values in the database;

(h) generating at least one signal corresponding to the pathway utilization value; and

(i) outputting the at least one utilization value signal to a signal processing means.

44. A method for rating medical care based upon manipulation and analysis of data related to clinical pathways, the method comprising the steps of:

(a) providing a clinical pathway database for storing:

(i) an initial procedure decision data element, corresponding to a decision point within the clinical pathway; and

(ii) at least one subsequent decision data elements, corresponding to available subsequent decision points within the clinical pathway;

(b) providing a historical clinical pathway database for storing:

(i) previously selected subsequent decision data elements, selected corresponding to the initial procedure decision data element; and

(ii) for each of the previously selected subsequent decision data elements, a rating value;

(c) selecting one of the at least one subsequent decision data elements;

(d) comparing the selected subsequent decision data element with the previously selected subsequent decision data elements stored in the historical clinical pathway database;

(e) based upon predetermined correlation criteria, modifying the at least one subsequent decision data elements within the clinical pathway database;

(f) accessing the historical clinical pathway database;

(g) computing a pathway rating value based on the accessed rating values in the historical database;

(h) generating at least one signal corresponding to the pathway rating value; and

(i) outputting the at least one rating signal to a signal processing means.

45. The method of claim 44, wherein the historical database further stores the identity of the medical care provider determining the selection subsequent decision element and wherein the comparing step is further responsive to the

identity.

46. The method of claim 44, wherein the historical database further stores a rating for each of the historical clinical pathways in the database, and wherein the comparing step is further responsive to the historical pathway ratings.

47. A system for collecting health care data for a person, comprising:

(a) a database for storing a clinical pathway, wherein the clinical pathway includes at least one decision point corresponding to at least one health-related event data and event type data to be collected with respect to the person;

(b) collection means for performing the steps of:

(i) collecting event data and event type data, as well as associated chronological data corresponding to a point in time that the event data occurred for the person, and

(ii) storing the event data, event type data and chronological data in the database.

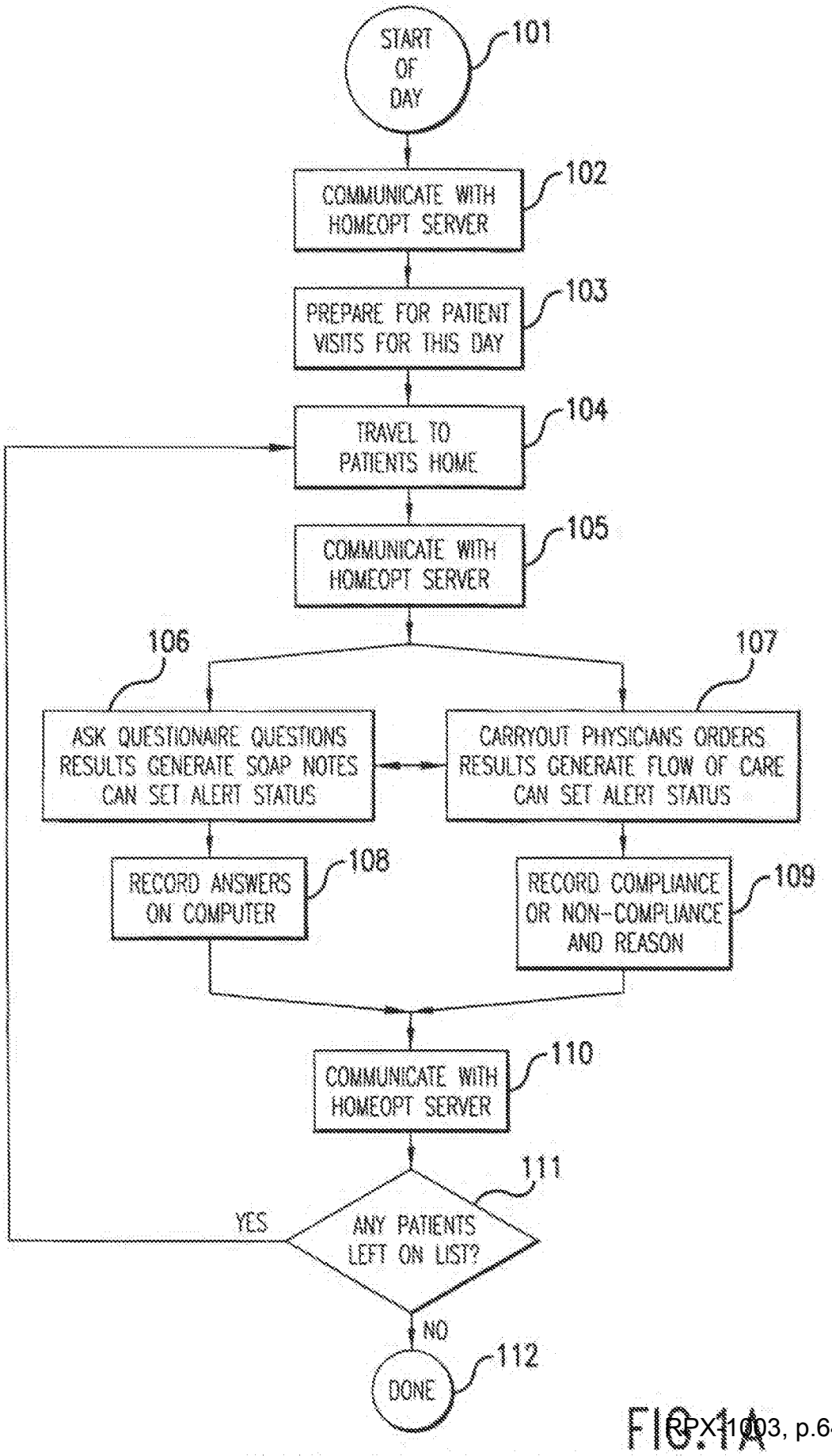


FIG.1A

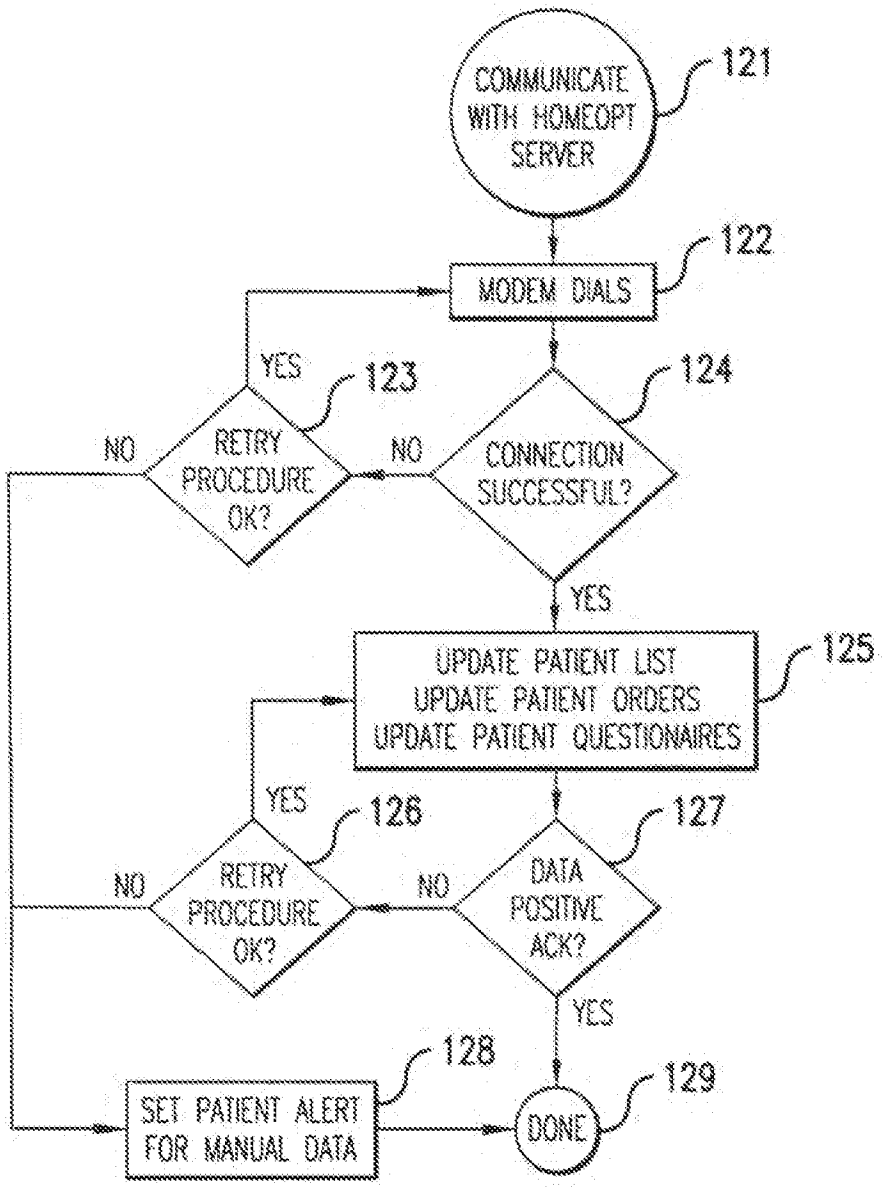

COMMUNICATE WITH HOMEOPT SERVER
121
MODEM DIALS
122
YES
123
RETRY PROCEDURE OK?
NO
NO
CONNECTION SUCCESSFUL?
124
YES
UPDATE PATIENT LIST
UPDATE PATIENT ORDERS
UPDATE PATIENT QUESTIONAIRES
125
YES
126
RETRY PROCEDURE OK?
NO
NO
DATA POSITIVE ACK?
127
YES
128
SET PATIENT ALERT FOR MANUAL DATA
DONE
129

FIG.1B

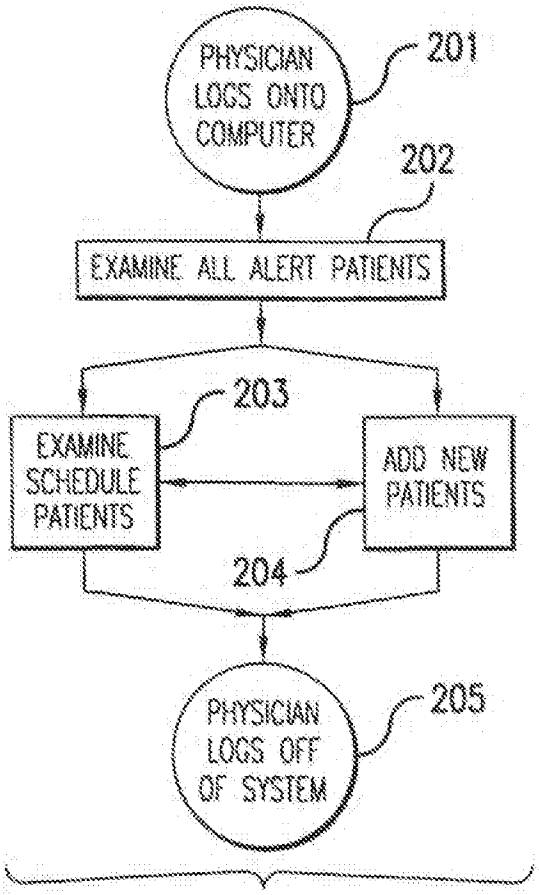

PHYSICIAN LOGS ONTO COMPUTER
201
202
EXAMINE ALL ALERT PATIENTS
203
EXAMINE SCHEDULE PATIENTS
ADD NEW PATIENTS
204
PHYSICIAN LOGS OFF OF SYSTEM
205

FIG.2A

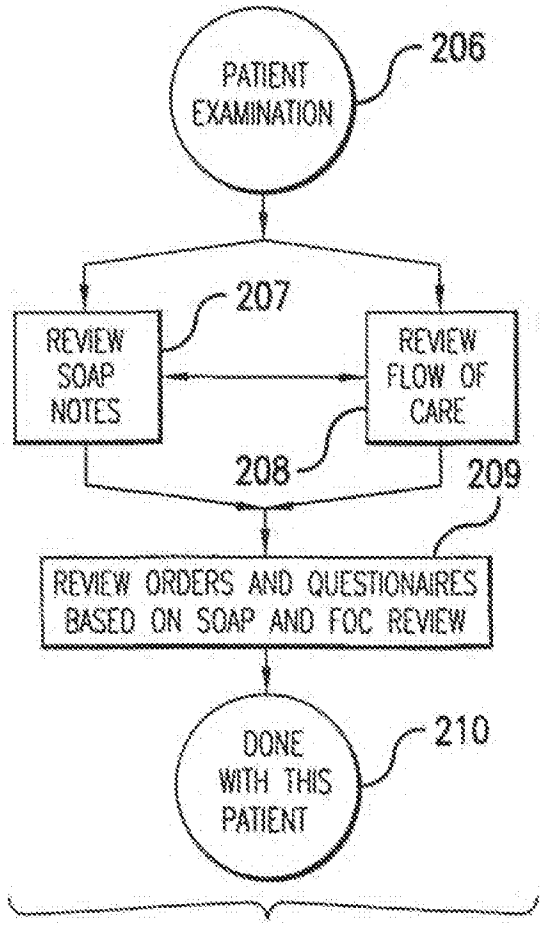

PATIENT EXAMINATION
206
207
REVIEW SOAP NOTES
REVIEW FLOW OF CARE
208
209
REVIEW ORDERS AND QUESTIONAIRES BASED ON SOAP AND FOC REVIEW
DONE WITH THIS PATIENT
210

FIG.2B

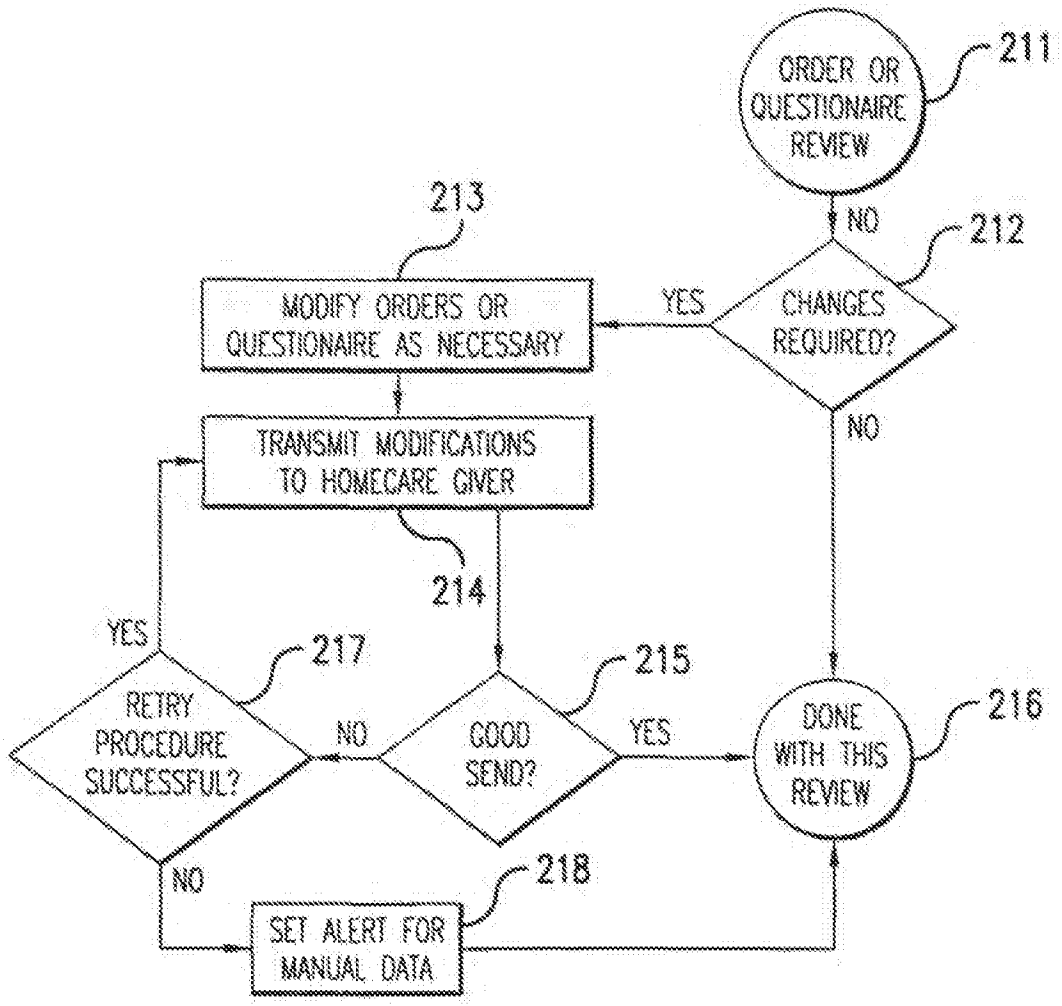

ORDER OR QUESTIONAIRE REVIEW
211
NO
CHANGES REQUIRED?
212
YES
MODIFY ORDERS OR QUESTIONAIRE AS NECESSARY
213
NO
TRANSMIT MODIFICATIONS TO HOMECARE GIVER
214
GOOD SEND?
215
YES
DONE WITH THIS REVIEW
216
NO
RETRY PROCEDURE SUCCESSFUL?
217
YES
NO
SET ALERT FOR MANUAL DATA
218

FIG.2C

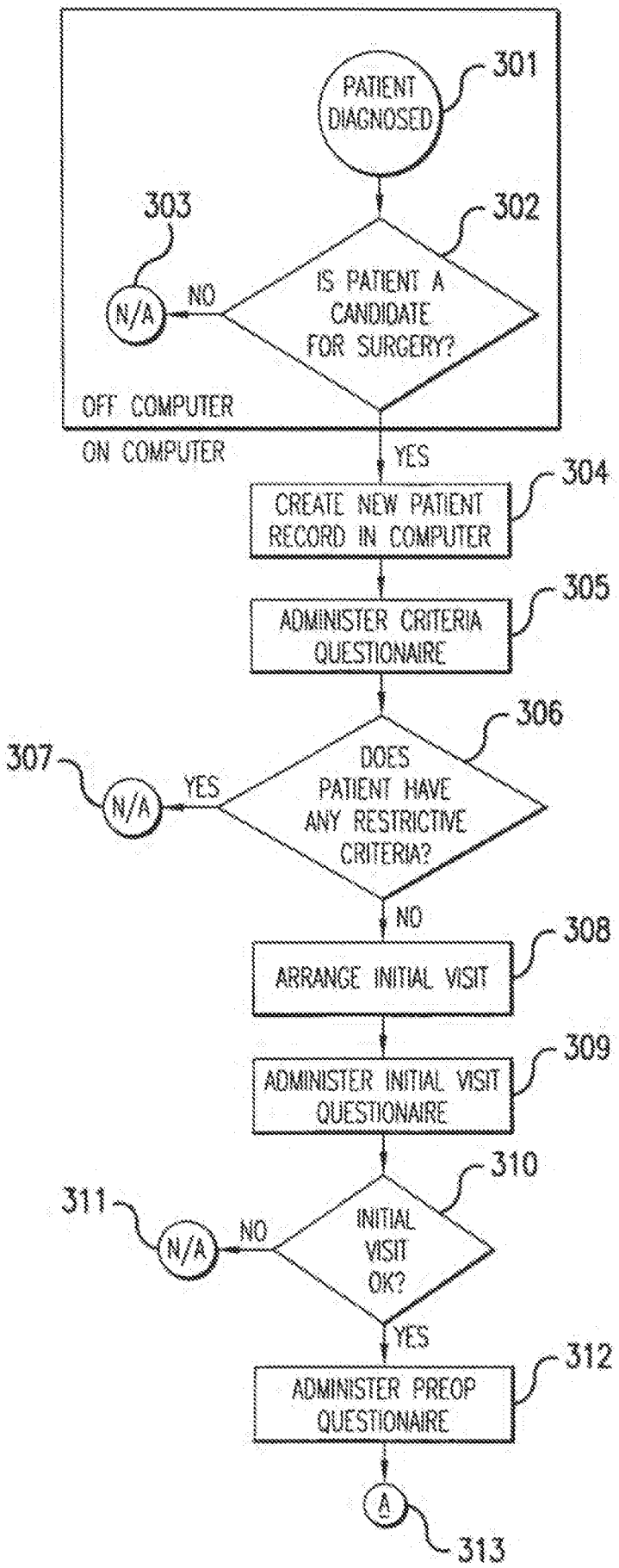


FIG.3A

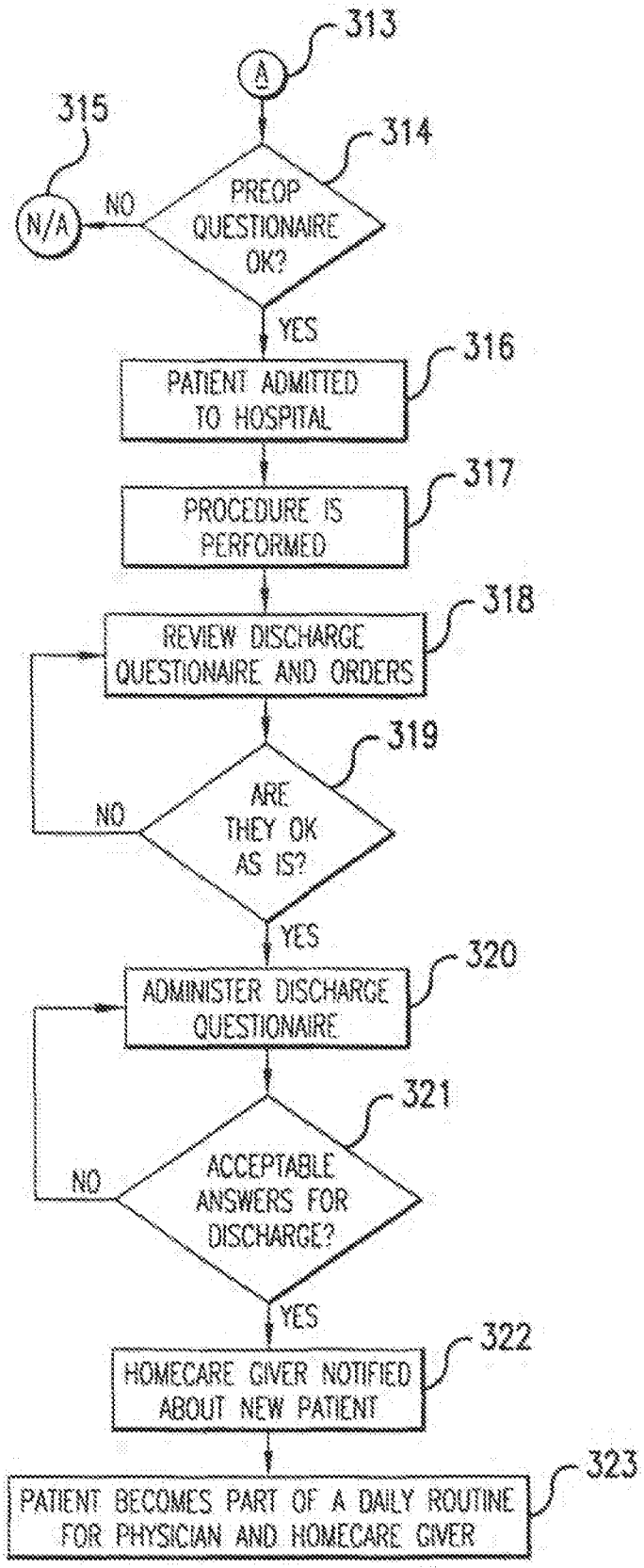


FIG.3B

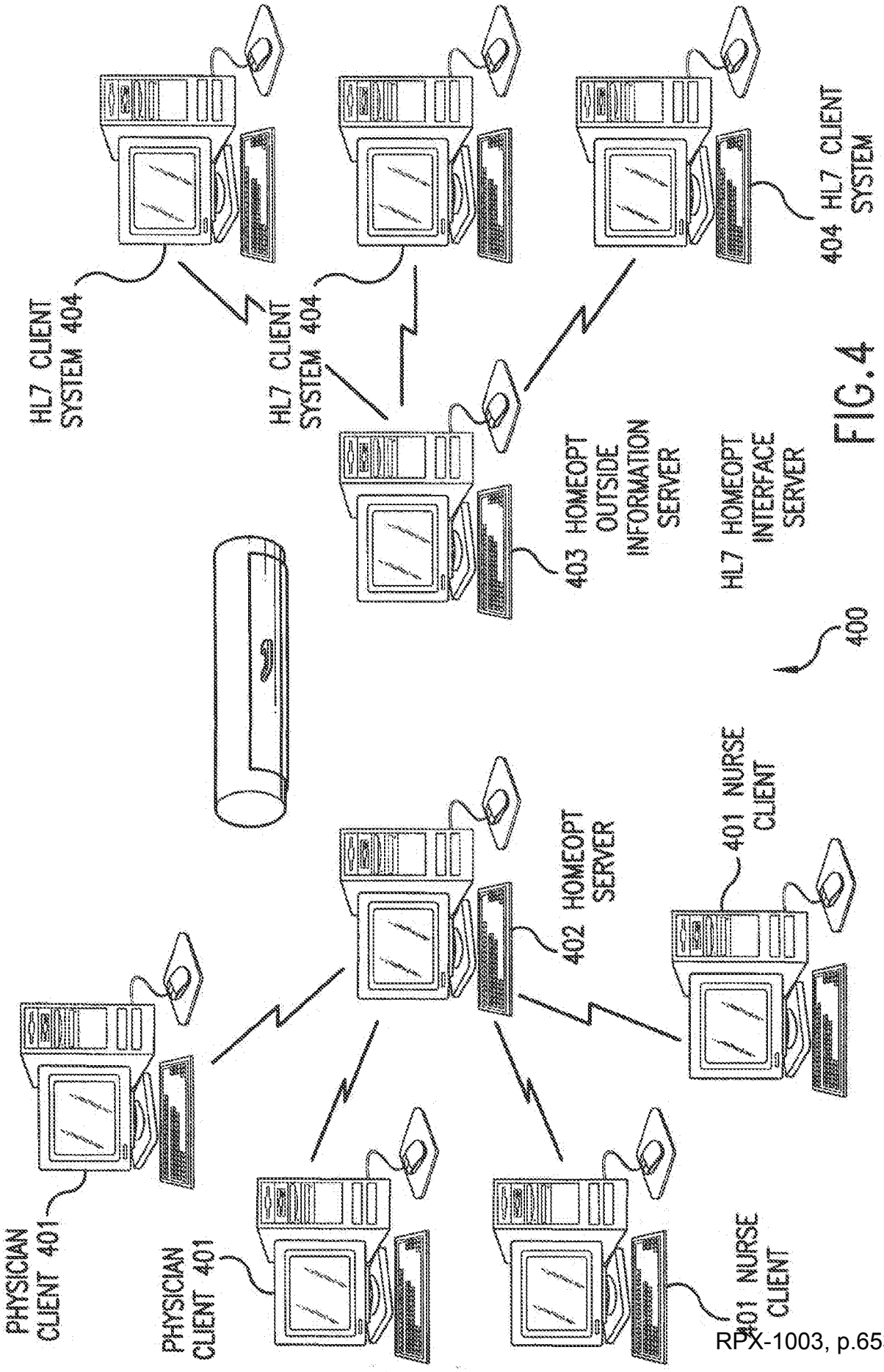


FIG.4

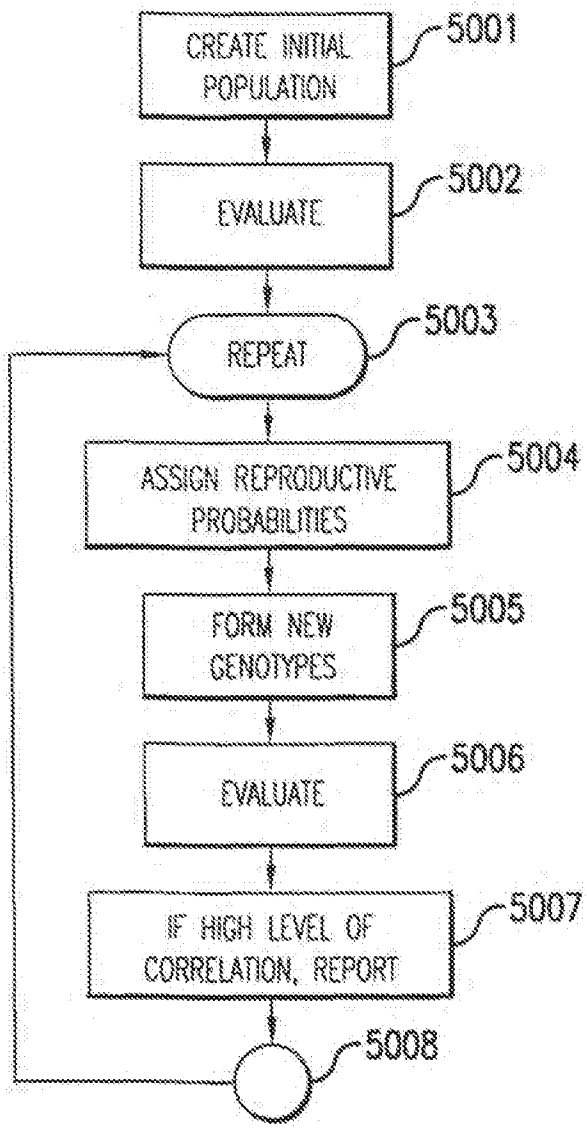

CREATE INITIAL POPULATION
5001
EVALUATE
5002
REPEAT
5003
ASSIGN REPRODUCTIVE PROBABILITIES
5004
FORM NEW GENOTYPES
5005
EVALUATE
5006
IF HIGH LEVEL OF CORRELATION, REPORT
5007
5008

FIG.5

PA-G

US005991771A

United States Patent [19]

Falls et al.

[11] Patent Number: 5,991,771

[45] Date of Patent: Nov. 23, 1999

[54] **TRANSACTION SYNCHRONIZATION IN A DISCONNECTABLE COMPUTER AND NETWORK**

[75] Inventors: **Patrick T. Falls**, Newbury; **Brian J. Collins**, New Malden; **Stephen P. W. Draper**, Basingstoke, all of United Kingdom

[73] Assignee: **Novell, Inc.**, Provo, Utah

[21] Appl. No.: **08/700,487**

[22] PCT Filed: **Jul. 18, 1996**

[86] PCT No.: **PCT/US96/11901**

§ 371 Date: **Jul. 3, 1997**

§ 102(e) Date: **Jul. 3, 1997**

[87] PCT Pub. No.: **WO97/04389**

PCT Pub. Date: **Feb. 6, 1997**

Related U.S. Application Data

[60] Provisional application No. 60/001,261, Jul. 20, 1995.

[51] **Int. Cl.**[6] **G06F 17/00**

[52] **U.S. Cl.** **707/202**; 707/201; 395/182.1; 395/182.13

[58] **Field of Search** 707/8, 10, 201, 707/203, 200, 202; 395/180, 182.1, 182.13

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,575,793	3/1986	Morel et al.	364/200
4,622,631	11/1986	Frank et al.	364/200
4,774,655	9/1988	Kaiila et al.	364/200
4,774,661	9/1988	Kumpati	364/300
4,827,399	5/1989	Shibayama	364/200
4,878,167	10/1989	Kapulka et al.	364/200
4,941,845	7/1990	Eppley et al.	439/505
5,001,628	3/1991	Johnson et al.	364/200
5,008,814	4/1991	Mathur	364/200
5,019,963	5/1991	Alderson et al.	364/200
5,043,876	8/1991	Terry	364/200

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

87107475	1/1988	European Pat. Off.	G06F 11/14
92308730	8/1993	European Pat. Off.	G06F 15/16
0508809	3/1995	European Pat. Off.	G05F 17/30
95100255	7/1995	European Pat. Off.	G06F 17/30

OTHER PUBLICATIONS

Advance Program—Second Workshop on the Management of Replicated Data (WMRD–II), Nov. 12–13, 1992, pp. 1–2.

"Application–Aware Adaptation for Mobile Computing", M. Satyanarayanan et al., *ACM SIGOS Operating Systems Review 29.1*, 1995, pp. 52–55.

"Architecture of the Ficus Scalable Replicated File System", T. Page, Jr., *Computer Science Department Technical Report University Of California At Los Angeles*, Mar. 1991, pp. 1–18.

"Coda: A Highly Available file System for a Distributed Workstation Environment", M. Satyanarayanan et al., *IEEE Transactions On Computers*, vol. 39 No. 4 Apr. 1990, pp. 447–459.

"Coding for Compression in Full–Text Retrieval Systems", A. Moffat et al., *IEEE DCC Data Compression Conference*, 1992, pp. 72–81.

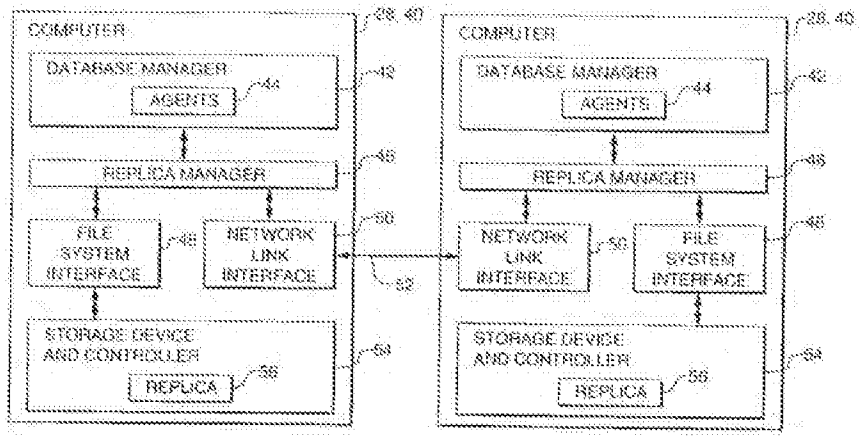
(List continued on next page.)

Primary Examiner—Paul V. Kulik
Attorney, Agent, or Firm—Computer Law

[57] **ABSTRACT**

A method and apparatus are disclosed for synchronizing transactions in a disconnectable network. Each transaction includes operations that were performed on a database replica on one computer while that computer was disconnected from another computer and hence from that other computer's replica. Transaction synchronization, which occurs after the computers are reconnected, transfers information from each computer to the other computer and applies updates to both replicas as appropriate. Transaction logs and clash handling tools may be used with the invention.

34 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

5,113,519	5/1992	Johnson et al.	395/600
5,142,680	8/1992	Ottman et al.	395/700
5,146,561	9/1992	Carey et al.	395/200
5,151,989	9/1992	Johnson et al.	395/600
5,155,847	10/1992	Kinoune et al.	395/600
5,159,669	10/1992	Trigg et al.	395/159
5,170,480	12/1992	Mohan et al.	395/600
5,185,857	2/1993	Rozmanith et al.	395/148
5,212,789	5/1993	Rago	395/600
5,229,768	7/1993	Thomas	341/51
5,237,680	8/1993	Adams et al.	395/600
5,247,683	9/1993	Holmes et al.	395/700
5,274,803	12/1993	Dubin et al.	395/600
5,276,868	1/1994	Poole	395/600
5,276,871	1/1994	Howarth	395/600
5,276,876	1/1994	Coleman et al.	395/650
5,278,979	1/1994	Foster et al.	395/600
5,278,982	1/1994	Daniels et al.	395/600
5,291,591	3/1994	Kawano et al.	395/600
5,297,278	3/1994	Wang et al.	395/600
5,313,646	5/1994	Hendricks et al.	395/600
5,317,728	5/1994	Tevis et al.	395/600
5,321,832	6/1994	Tanaka et al.	395/600
5,325,524	6/1994	Black et al.	395/600
5,333,315	7/1994	Saether et al.	395/600
5,347,653	9/1994	Flynn et al.	395/600
5,355,476	10/1994	Fukumaru	395/600
5,375,207	12/1994	Blakely et al.	395/200
5,377,326	12/1994	Murata et al.	395/200
5,388,256	2/1995	Herbert	395/600
5,390,335	2/1995	Stephan et al.	395/800
5,403,639	4/1995	Belsan et al.	395/600
5,408,619	4/1995	Oran	395/325
5,410,543	4/1995	Seitz et al.	370/85.13
5,410,684	4/1995	Ainsworth et al.	395/575
5,412,801	5/1995	de Remer et al.	395/575
5,418,957	5/1995	Narayan	395/700
5,423,034	6/1995	Cohen-Levy et al.	395/600
5,430,871	7/1995	Jamoussi et al.	395/600
5,434,994	7/1995	Shaheen et al.	395/500
5,452,450	9/1995	Delory	395/600
5,553,279	9/1996	Goldring	395/600
5,588,147	12/1996	Neeman et al.	395/601
5,613,113	3/1997	Goldring	395/618
5,666,530	9/1997	Clark et al.	395/617
5,684,984	11/1997	Jones et al.	395/610
5,692,129	11/1997	Sonderegger et al.	395/200.11
5,710,922	1/1998	Alley et al.	395/617
5,737,600	4/1998	Geiner et al.	395/616
5,737,601	4/1998	Jain et al.	395/617
5,740,433	4/1998	Carr et al.	395/618
5,761,660	6/1998	Josten et al.	707/8
5,774,717	6/1998	Porcaro	707/202
5,778,390	7/1998	Nelson et al.	707/204
5,806,075	9/1998	Jain et al.	707/201
5,832,518	11/1998	Masters	707/202
5,876,434	3/1999	Draper et al.	707/202

OTHER PUBLICATIONS

"A compact representation for file versions: a preliminary report", A. Black et al., *5th IEEE Conference On Data Engineering*, 1989, pp. 321–329.

"Concurrency Control and Consistency of Multiple Copies of Data in Distributed INGRES", M. Stonebraker, *IEEE Transactions On Software Engineering*, vol. SE–3, No. 3, May 1979, pp. 188–194.

"Conflict Detection Tradeoffs for Replicated Data", M. Carey et al., *ACM Transactions on Database Systems*, vol. 16, No. 4, Dec. 1991, pp. 703–746.

"Countdown to Mobile Blast–Off", I. Brodsky, *Network World*, Feb. 19, 1996, pp. 44–46,52.

"Data Management for Mobile Computing", T. Imielinski et al., *ACM SIGMOD Record*, vol. 22, No. 1, Mar. 1993, pp. 34–39.

"Data Replicas in Distributed Information Services", B. Gladney, *ACM Transactions on Database Systems*, vol. 14, No. 1, Mar. 1989, pp. 75–97.

"Database System Issues in Nomadic Computing", R. Alonso et al., *ACM SIGMOD Record*, 22 2, 1993, pp. 388–392.

"DGDBM: Programming Support for Distributed Transactions Over Replicated Files", M. Franky, *ACM SIGOS Operating Systems Review*, 29 3, Jul. 1995, pp. 64–74.

"Disconnected Operation for AFS", L. Huston et al., Mobile and Location—Independent Computing Symposium, USENIX Association, 1994, pp. 1–10.

"Disconnected Operation in the Coda File System", J. Kistler et al., *ACM Operating Systems Review*, 25 5, 1991, pp. 213–225.

"Disconnected Operation in a Distributed File System", J. Kistler, Ph.D. thesis, Department of Computer Science, Carnegie Mellon University, May 1993, pp. 1–186.

"Discord in hardwareland", T. Schmidt, *Network World*, Feb. 19, 1996, p. 47.

"Distributed Logging for Transaction Processing", D. Daniels et al., *ACM*, 1987, pp. 82–96.

"Experience with Disconnected Operation in a Mobile Computing Environment", M. Satyanarayanan et al., *Mobile and Location–Independent Computing Symposium*, 1994, pp. 11–28.

"Fixed Length Semiorder Preserving Code for Field Level Data File Compression", M. Inyama et al., *IEEE—First International Conference on Data Engineering*, 1984, pp. 244–253.

"Flexible and Safe Resolution of File Conflicts", P. Kumar et al., *1995 USENIX Technical Conference*, Jan. 16–20, 1995, pp. 95–106.

"The Generalized Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data", D. Agrawal et al., *ACM Transactions on Database Systems*, vol. 17, No. 4, Dec. 1992, pp. 689–717.

"A Generic Multicast Transport Service to Support Disconnected Operation", S. Mafficis et al., *Mobile and Location–Independent Computing Symposium*, 1995, pp. 79–89.

"Getting Your Byte's Worth", S. Vaughan–Nichols, *Byte*, Nov. 1990, pp. 331–336.

"Grapevine: An Exercise in Distributed Computing—Abstract", A. Birrell et al., *Communications of the ACM*, vol. 25, No. 4, Apr. 1982, pp. 260–261.

"Going Mobile", S. Biagi, *Network Var*, Apr. 1996, p. 14.

"Impact of Mobility on Distributed Computations", R. Badrinath et al., *ACM SIGOS Operating Systems Review*, 27 2, 1993, pp. 15–20.

"An Introduction to Database Systems vol. II", C. Date, Addison–Wesley Publishing Company, 1993, pp. 1–33, 291–340.

"Isolation–Only Transactions for Mobile Computing", Q. Lu et al., *ACM SIGOS Operating Systems Review*, 28 2, 1994, pp. 81–87.

"Log–Based Directory Resolution in the Coda File System", P. Kumar et al., *IEEE*, 1993, pp. 202–213.

"The Lotus Notes™ Storage System", K. Moore, *ACM SIGMOD Record*, 24 2, 1995, pp. 427–428.

"Low Cost Management of Replicated Data in Fault-Tolerant Distributed Systems", T. Joseph et al., *ACM Transactions on Computer Systems*, vol. 4, No. 1, Feb. 1986, pp. 54–70.

"Maintaining Availability in Partitioned Replicated Databases", A. Abbadi et al., *ACM Transactions on Computer Systems*, vol. 14, No. 2, Jun. 1989, pp. 264–290.

"Model Based Concordance Compression", A. Bookstein et al., *IEEE DCC Data Compression Conference*, 1992, pp. 82–91.

"The Multicast Policy and its Relationship to Replicated Data Placement", O. Wolfson et al., *ACM Transactions on Database Systems*, vol. 16, No. 1, Mar. 1991, pp. 181–205.

"A Multi-Group Technique for Data Compression", E. Hazboun et al., *ACM SIGMOD Conference*, 1982, pp. 284–292.

"NetWare 4 for Professionals", D. Bierer et al., *New Riders Publishing*, 1993, pp. 359–374.

"A Non-Blocking Transaction Data Flow Graph Based Approach For Replicated Data", P. Krishna Reddy et al., *Operating Systems Review (SIGOPS) 27 No. 3*, Jul. 1993, pp. 46–54.

"Partially Connected Operation", L. Huston et al., *Mobile and Location-Independent Computing Symposium*, 1995, pp. 91–97.

"Peephole Log Optimization", L. Huston et al., *IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1994, pp. 1–8.

"Performing Remote Operations Efficiently on a Local Computer Network", A. Spector, *Communications of the ACM*, vol. 25, No. 4, Apr. 1982, pp. 246–259.

"Primarily Disconnected Operation: Experiences with Ficus", J. Heidemann et al., *IEEE*, 1992, pp. 2–5.

"Replicated Data in a Distributed Environment", M. Colton, *ACM SIGMOD Record*, 22 2, 1993, pp. 464–466.

"Remote access can't slow down", H. Allard, *Network World*, Feb. 19, 1996, p. 53.

"A Replicated UNIX File System (Extended Abstract)", B. Liskov et al., *ACM SIGOS Operating Systems Review*, 25 1, 1991, pp. 60–64.

"Replication in the Harp File System", B. Liskov, *ACM Operating Systems Review*, 25 5, 1991, pp. 226–238.

"Resolving File Conflicts In The Ficus File System", P. Reiher et al., *1994 Summer Usenix*, Jun. 6–10, 1994, pp. 183–195.

"RFS Architectural Overview", A. Rifkin et al., Jun. 1986, pp. 248–259.

"Scalable, Secure, and Highly Available Distributed File Access",M. Satyanarayanan, *Computer 23 No.5*, May 1990, pp. 9–20.

"A Snapshot Differential Refresh Algorithm", B. Lindsay et al., *ACM SIGMOD Record*, 15 2, 1986, pp. 53–60.

"Software spins wheels in niche markets", K. Scherberger, *Network World*, Feb. 19, 1996, p. 49.

Space and Time Savings Through Large Data Base Compression and Dynamic Restructuring, F. Alsberg, *Proceedings of the IEEE*, vol. 63, No. 8, Aug. 1975, pp. 1114–1122.

"Sun-3 Architecture" Aaron., Aug. 1986, pp. 8–9, 49–57.

"Supporting Application-Specific Resolution in an Optimistically Replicated File System", P. Kumar et al., *IEEE*, 1993, pp. 66–70.

"System Isolation and Network Fast-Fail Capability in Solaris", G. Menenegro et al., *Mobile and Location-Independent Computing Symposium*, 1995, pp. 67–78.

"Transaction Support in a Log-Structured File System", M. Seltzer, *IEEE—Ninth International Conference on Data Engineering*, 1993, pp. 503–510.

"The Transparent Remote File System", R. Hughes, Date Unknown.

"Two Levels of Filesystem Hierarchy on One Disk", V. Cate, *Department of Computer Science, Carnegie Mellon University*, May 1990, pp. 1–20.

"Using Prospero to Support Integrated Location-Independent Computing", B. Neuman et al., *Mobile and Location-Independent Computing Symposium*, 1994, pp. 29–34.

"Wireless IR lets mobile devices get personal" (partial article),J. Edney, *Electronic Engineering Times*, Feb. 19, 1996, p. 44.

"Wireless LANs roaming for standards" (partial article), unknown, *Electronic Engineering Times*, Feb. 19, 1996, p. 65.

"Wireless nets come of age", I. Gillon, *Network World*, Feb. 19, 1996, p p. 50, 52.

Summary of Fitler et al. Invention, 1992.

Mobile NetWare Lite Specification, Version 1.0, Aug. 20, 1992 (best available copy).

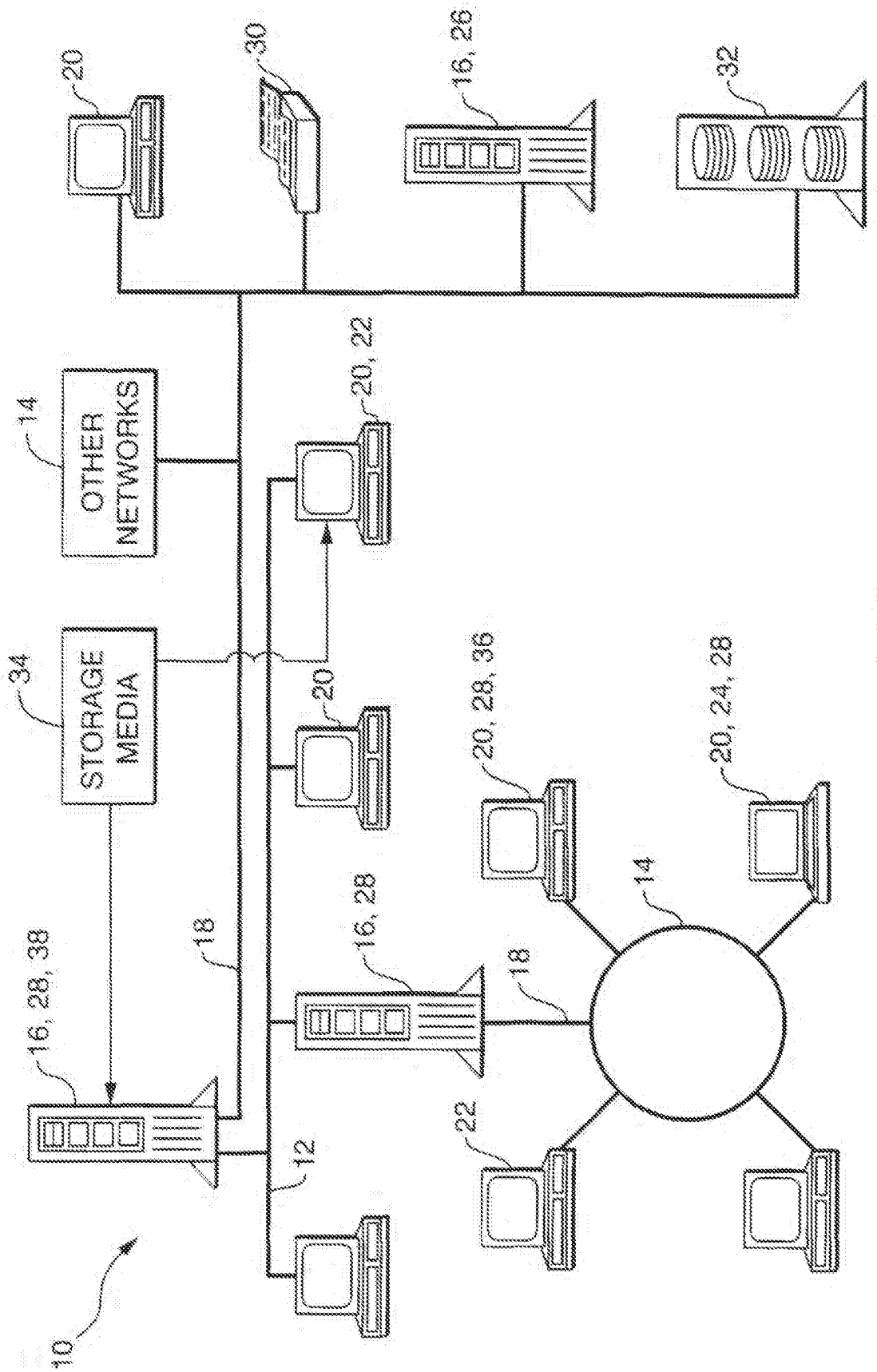

20
30
16, 26
32
OTHER NETWORKS
14
STORAGE MEDIA
34
20, 22
20
20, 28, 36
20, 24, 28
16, 28
18
16, 28, 38
14
18
22
12
10

FIG. 1

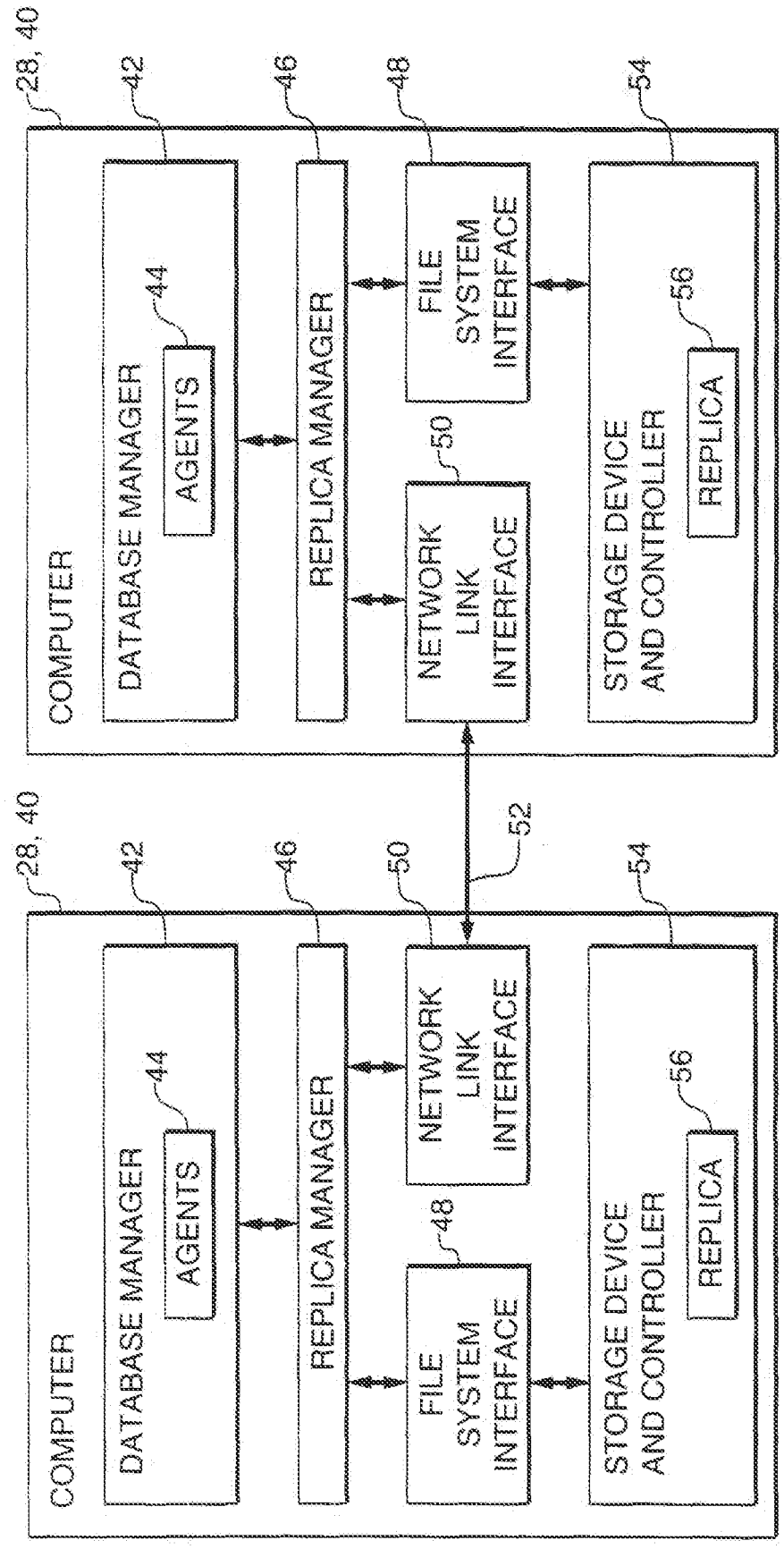

28, 40
42
46
48
54
COMPUTER
DATABASE MANAGER
AGENTS
44
REPLICA MANAGER
FILE SYSTEM INTERFACE
NETWORK LINK INTERFACE
50
STORAGE DEVICE AND CONTROLLER
REPLICA
56
52

FIG. 2

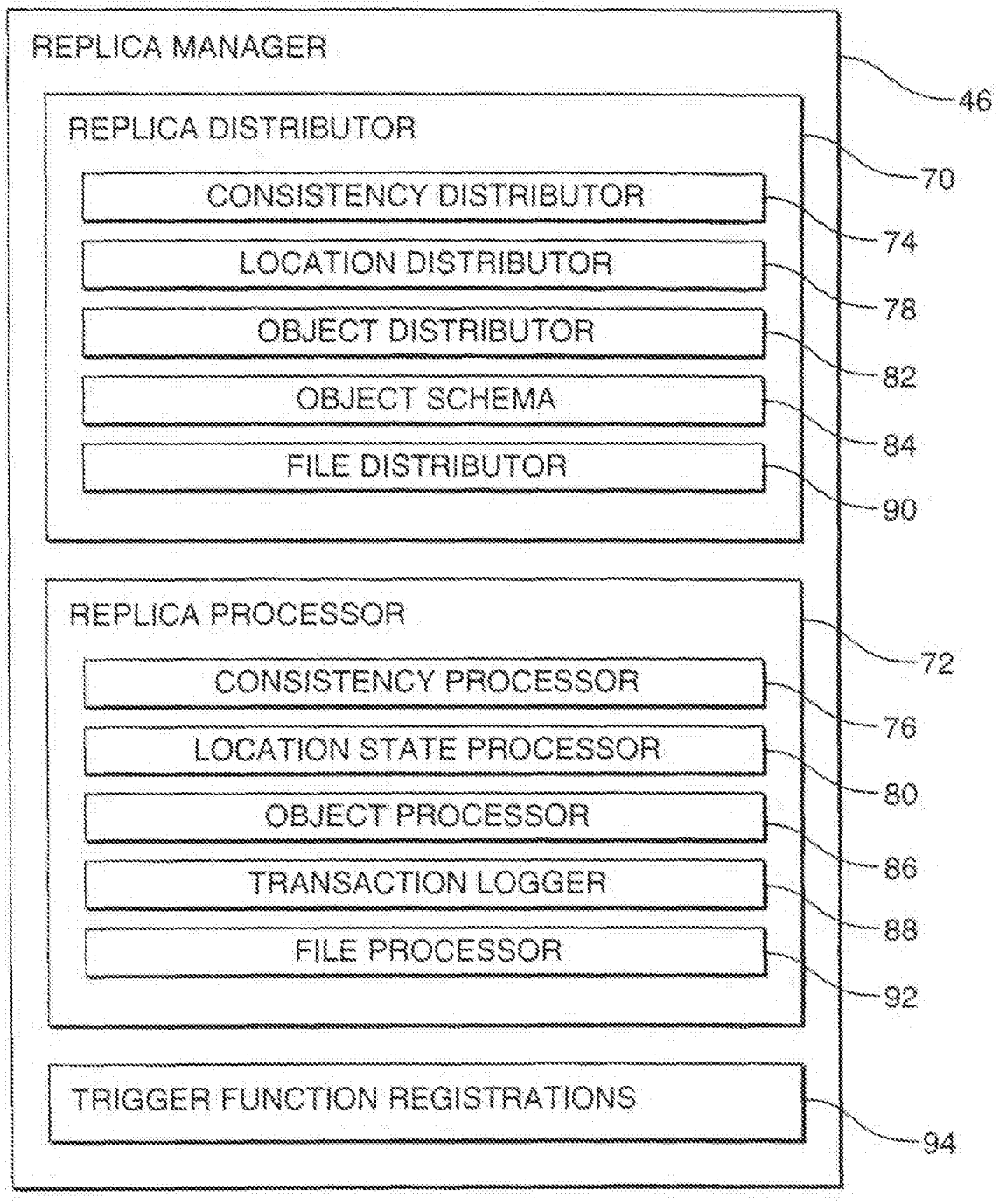

REPLICA MANAGER
46
REPLICA DISTRIBUTOR
70
CONSISTENCY DISTRIBUTOR
74
LOCATION DISTRIBUTOR
78
OBJECT DISTRIBUTOR
82
OBJECT SCHEMA
84
FILE DISTRIBUTOR
90
REPLICA PROCESSOR
72
CONSISTENCY PROCESSOR
76
LOCATION STATE PROCESSOR
80
OBJECT PROCESSOR
86
TRANSACTION LOGGER
88
FILE PROCESSOR
92
TRIGGER FUNCTION REGISTRATIONS
94

FIG. 3

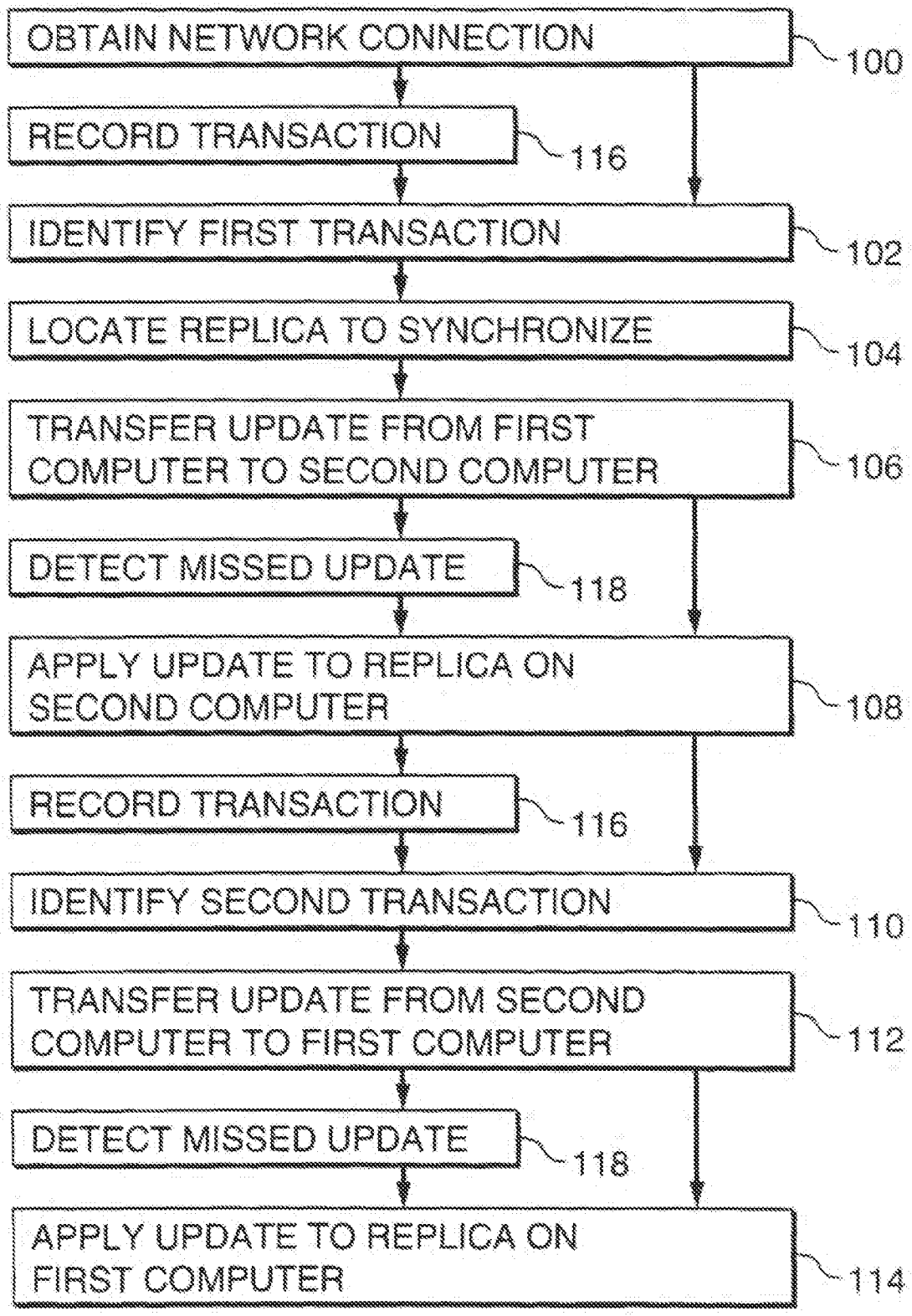

OBTAIN NETWORK CONNECTION
100
RECORD TRANSACTION
116
IDENTIFY FIRST TRANSACTION
102
LOCATE REPLICA TO SYNCHRONIZE
104
TRANSFER UPDATE FROM FIRST COMPUTER TO SECOND COMPUTER
106
DETECT MISSED UPDATE
118
APPLY UPDATE TO REPLICA ON SECOND COMPUTER
108
RECORD TRANSACTION
116
IDENTIFY SECOND TRANSACTION
110
TRANSFER UPDATE FROM SECOND COMPUTER TO FIRST COMPUTER
112
DETECT MISSED UPDATE
118
APPLY UPDATE TO REPLICA ON FIRST COMPUTER
114

FIG. 4

TRANSACTION SYNCHRONIZATION IN A DISCONNECTABLE COMPUTER AND NETWORK

This application is a 371 of PCT/US96/11901 filed Jul. 18, 1996. This case claims benefit of provisional application Ser. No. 60/001201 filed Jul. 20, 1995.

FIELD OF THE INVENTION

The present invention relates to the synchronization of transactions performed on separated disconnectable computers, such as transactions performed on a mobile computer and on a computer network while the mobile computer and the network are disconnected, or transactions performed on separate server computers in a network. More particularly, the present invention relates to the synchronization of transactions when the separate computers are reconnected.

TECHNICAL BACKGROUND OF THE INVENTION

It is often convenient, and sometimes essential, to carry a computer and selected data while traveling. It may also be convenient or essential to access a computer network using a "mobile computer" such as a laptop, palmtop, notebook, or personal digital assistant. However, different types of mobile computing make very different assumptions about the use and availability of computer networks.

Some mobile computers are not ordinarily connected to a computer network. Like their non-traveling "stand-alone" counterparts, such "walk-alone" computers cannot be connected to a network unless significant hardware or software modifications are made to them or to the network.

"Mobile-link" portable computers are typically connected to a computer network and attempt (with varying degrees of success) to maintain that network connection during mobile use through a wireless link. Typical wireless links use radio waves or infrared light as carriers. Mobile-link computers can be used in a walk-alone mode if the network connection is lost. However, mobile-link systems provide few or no automatic facilities to synchronize the mobile-link computer with the network when the connection is re-established.

"Disconnectable" computers include portable computers that operate in either a walk-alone or a mobile-link mode and provide significant automated facilities for synchronizing operations performed on the mobile computer with operations performed on the network. Disconnectable computers need not be portable. For instance, separate server computers in a wide-area network (WAN) or other network that are connected to one another only sporadically or at intervals may be disconnectable computers.

Unfortunately, conventional disconnectable computers still rely routinely on manually directed file copying to select the data that will be used in the field. Moreover, conventional disconnectable computer systems are not easily extended to support a variety of database formats, and they do not properly handle the situation in which changes to the "same" data are made on both the portable computer and on a network computer during disconnected operation.

For instance, the Coda File System ("Coda") is a client-server system that provides limited support for disconnectable operation. To prepare for disconnection, a user may hoard data in a client cache by providing a prioritized list of files. On disconnection, two copies of each cached file exist: the original stored on the server, and a duplicate stored in the disconnected client's cache. The user may alter the duplicate file, making it inconsistent with the server copy. Upon reconnection, this inconsistency may be detected by comparing timestamps.

However, the inconsistency is detected only if an attempt is made to access one of the copies of the file. The Coda system also assumes that the version stored in the client's cache is the correct version, so situations in which both the original and the duplicate were altered are not properly handled. Moreover, the Coda synchronization mechanism is specifically tailored, not merely to file systems, but to a particular file system (a descendant of the Andrew File System). Coda provides no solution to the more general problem of synchronizing transactions in a distributed database that can include objects other than file and directory descriptors.

Some approaches to distributed database replication are not directed to mobile computing per se but do attempt to ensure consistency between widely separated replicas that collectively form the database. Examples include, without limitation, the replication subsystem in Lotus Notes and the partition synchronization subsystem in Novell NetWare® 4.1 (LOTUS NOTES is a trademark of International Business Machines, Inc. and NETWARE is a registered trademark of Novell, Inc.).

However, some of these approaches to replication are not transactional. A transaction is a sequence of one or more operations which are applied to a replica on an all-or-nothing basis. Non-transactional approaches may allow partially completed update operations to create inconsistent internal states in network nodes. Non-transactional approaches may also require a synchronization time period that depends directly on the total number of files, directories, or other objects in the replica. This seriously degrades the performance of such approaches when the network connection used for synchronization is relatively slow, as many modem or WAN links are.

Moreover, in some conventional approaches potentially conflicting changes to a given set of data are handled by simply applying the most recent change and discarding the others. Another drawback of several conventional approaches to replication is the requirement they impose that either or both computer systems be locked out of use while the replicas are being synchronized.

Another drawback of conventional disconnected computing approaches is that the location of data on the mobile computer does not always correspond to its location on the network computer. Files may be located in one subdirectory or on one drive during connected operation and in another subdirectory or on another drive during disconnected operation. Thus, the mobile computer does not present the same view of the network when it is disconnected as it does when connected to the network. In addition to creating a risk of confusion and conflicting file versions, these conventional approaches require users to repeatedly reconfigure application programs to look for data in different locations.

Thus, it would be an advancement in the art to provide a system and method for properly synchronizing transactions when a disconnectable computer is reconnected to a network.

It would be an additional advancement to provide such a system and method which identify potentially conflicting database changes and allow their resolution by either automatic or manual means.

It would also be an advancement to provide such a system and method which are not limited to file system operations but can instead be extended to support a variety of database objects.

It would be an additional advancement to provide such a system and method which do not require a synchronization time period that depends directly on the total number of files, directories, or other objects in the replicas.

It would be a further advancement to provide such a system and method which do not lock either the mobile computer or the network computers during synchronization.

It would be an additional advancement to provide such a system and method which present consistent file locations regardless of whether the mobile computer is connected to the network.

Such a system and method are disclosed and claimed herein.

BRIEF SUMMARY OF THE INVENTION

The present invention provides a system and method which facilitate disconnected mobile computing in several ways. Prior to disconnection, the invention allows network administrators or users to readily select data that should be copied from a network to a mobile computer by simply identifying one or more target database subtrees. During disconnected operation of the mobile computer, the invention presents the user with a "virtual network" environment that is consistent in use and appearance with the selected portion of the actual network.

Finally, upon reconnection of the mobile computer to the network, the invention synchronizes operations performed on the mobile computer during the disconnected interval with operations performed on the network during that interval. Synchronization is both substantially automatic and transactional, so minimal user intervention is needed and inconsistent internal states are avoided. Moreover, synchronization does not routinely discard any of the changes made on either the network or the mobile computer.

One embodiment of a system according to the present invention includes at least two computers capable of being connected by a network link. One computer will act as the mobile computer, while the other acts as the network. Of course, the network may also include more than one computer after the mobile computer is disconnected. Suitable mobile computers include laptops, palmtops, notebooks, and personal digital assistants. Suitable network computers include additional mobile computers as well as desktop, tower, workstation, micro-, mini-, and mainframe computers. Suitable network links include packet-based, serial, internet-compatible, local area, metropolitan area, wide area, and wireless network links.

Each of the computers includes a non-volatile storage device such as a magnetic or optical disk or disk array. Initially, the storage device on the network computer contains at least a portion of a target database. The target database includes file descriptors, directory descriptors, directory services objects, printer jobs, or other objects. The target database is a distributed database whose entries may be kept in one or more replicas on different computers.

Each replica of the target database contains at least some of the same variables or records as the other replicas, but different replicas may temporarily contain different values for the same variable or record. Such inconsistencies are temporary because changes in value are propagated throughout the replicas by the invention. Thus, if the changes to a particular variable or record are infrequent relative to the propagation delay, then all replicas will converge until they contain the same value for that variable or record.

Selected portions of the database may be copied from the network computer to the mobile computer's storage device prior to disconnection as a basis for the mobile computer's virtual network. Copying is accomplished using a device controller in each computer, a replica manager on each computer, and the network link. The device controller on each computer communicates with that computer's storage device to control data transfers.

Each computer's replica manager communicates with the device controller of that computer and with the network link. Each replica manager also communicates with a database manager on its computer. The database manager can send database transactions to the device controller only through the replica manager, allowing the replica managers to log transactions and to synchronize the transactions after the network connection is re-established.

Each replica manager includes a replica distributor and a replica processor. The replica distributor insulates the database manager from the complexities caused by having target database entries stored in replicas on multiple computers, while still allowing the database manager to efficiently access and manipulate individual target database objects, variables, and/or records. The replica processor maintains information about the location and status of each replica and ensures that the replicas tend to converge.

The network link supports a remote procedure call ("RPC"), distributed memory, or similar mechanism to allow replica distributors to call procedures in the replica processors on one or more network computers. The network link also tracks connectivity information such as which network computers are currently accessible and what state those computers are in.

Each replica distributor includes at least a consistency distributor and a location distributor, and each replica processor includes at least a consistency processor and a location state processor. The consistency distributors and consistency processors maintain convergent consistency of the target database replicas. The location distributors and the location state processors are used to determine the storage locations of database entries.

The replica distributor may also include an object distributor and an object schema, in which case the corresponding replica processor includes an object processor. The object distributor provides an interface to target database objects, making operations such as "add object", "modify object", and "read object" available. The objects are defined using a compile-time schema definition language. The database manager and various subsystems of the replica manager can all query the object schema to obtain information regarding the format and storage requirements of objects, but semantic interpretation of object values is generally restricted to the database manager.

One embodiment of the replica processor also includes a transaction logger which maintains a log of recent updates for each object in the target database. This log allows recovery of local transactions after power losses or other unexpected interruptions. The transaction log at a given location also provides an efficient source of the updates needed to bring other locations up to date. Transaction logs are further described in a commonly owned copending application Ser. No. 08/700,490, entitled TRANSACTION LOG MANAGEMENT IN A DISCONNECTABLE COMPUTER AND NETWORK, filed the same day and having the same inventors as the present application.

In one embodiment, the replica distributor and replica processor contain a file distributor and a file processor, respectively. These file subsystems provide access to file contents for operations such as "read" and "write" on file

objects. The file distributor and processor insulate the database manager from complexities caused by the distributed nature of the target database files. More generally, the replica managers intercept any file system or operating system call that directly accesses replicated files or database entries, so that consistent convergent replicas are maintained.

One embodiment of the replica manager contains trigger function registrations. Each registration associates a trigger function with a target database operation such that the registered trigger function will be invoked on each replica, once the computers are connected, if the associated operation is requested of the database manager. The trigger function is invoked on each replica after the associated operation request is transmitted from the database manager to the replica manager. Trigger functions can be used to handle tasks such as file replication, where the file contents are not directly accessed by the database manager, while ensuring that files converge in a manner consistent with the database operation.

In operation, the replica managers synchronize transactions upon reconnection in the following manner. Using the network link, a network connection is created between the mobile computer and a network computer. The network computer need not be the network computer from which the mobile computer was disconnected. The replica manager on the mobile computer identifies a transaction that targets an object in a replica on the mobile computer, and locates a corresponding replica that resides on the network computer. The mobile computer then transfers an update based on the transaction over the network connection to the network computer.

Meanwhile, the replica manager on the network computer performs similar steps, determining whether another transaction targeted an entry in the network computer replica and transferring an update based on any such transaction to the mobile computer's replica manager over the same network connection. The respective replica managers then apply the transaction updates to their respective replicas. The process is repeated for any other replicas in the network, with pairs of replica managers propagating the updates from the mobile computer throughout the network. To prevent inconsistencies, access to each replica is by way of a target database object lock that serializes updates to the replica, and the updates are applied atomically.

Each completed transaction has a corresponding transaction sequence number, and the transaction sequence numbers are consecutive and monotonic for all completed transactions. The update transferred by the replica manager includes both the transaction sequence number of the transaction in question and a location identifier specifying the computer on which the transaction was first requested. A missed update is indicated by a gap in the sequence of transferred transaction numbers.

During synchronization the replica managers detect mutually inconsistent updates to a given entry, and attempt to resolve such "clashes" automatically or with user assistance. Clash handling is further described in a commonly owned copending application entitled TRANSACTION CLASH MANAGEMENT IN A DISCONNECTABLE COMPUTER AND NETWORK, filed the same day and having the same inventors as the present application, now U.S. Pat. No. 5,878,434.

The features and advantages of the present invention will become more fully apparent through the following description and appended claims taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

To illustrate the manner in which the advantages and features of the invention are obtained, a more particular description of the invention summarized above will be rendered by reference to the appended drawings. Understanding that these drawings only provide selected embodiments of the invention and are not therefore to be considered limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

FIG. **1** is a schematic illustration of a computer network suitable for use with the present invention.

FIG. **2** is a diagram illustrating two computers in a network, each configured with a database manager, replica manager, network link manager, and other components according to the present invention.

FIG. **3** is a diagram further illustrating the replica managers shown in FIG. **2**.

FIG. **4** is a flowchart illustrating transaction synchronization methods of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Reference is now made to the Figures wherein like parts are referred to by like numerals. The present invention relates to a system and method which facilitate disconnected computing with a computer network. One of the many computer networks suited for use with the present invention is indicated generally at **10** in FIG. **1**.

In one embodiment, the network **10** includes Novell NetWare® network operating system software, version 4.x (NETWARE is a registered trademark of Novell, Inc.). In alternative embodiments, the network includes Personal NetWare, NetWare Mobile, VINES, Windows NT, LAN Manager, or LANtastic network operating system software (VINES is a trademark of Banyan Systems, NT and LAN Manager are trademarks of Microsoft Corporation; LANtastic is a trademark of Artisoft). The network **10** may include a local area network **12** which is connectable to other networks **14**, including other LANs, wide area networks, or portions of the Internet, through a gateway or similar mechanism.

The network **10** includes several servers **16** that are connected by network signal lines **18** to one or more network clients **20**. The servers **16** may be file servers, print servers, database servers, Novell Directory Services servers, or a combination thereof. The servers **16** and the network clients **20** may be configured by those of skill in the art in a wide variety of ways to operate according to the present invention.

The network clients **20** include personal computers **22**, laptops **24**, and workstations **26**. The servers **16** and the network clients **20** are collectively denoted herein as computers **28**. Suitable computers **28** also include palmtops, notebooks, personal digital assistants, desktop, tower, micro-, mini-, and mainframe computers. The signal lines **18** may include twisted pair, coaxial, or optical fiber cables, telephone lines, satellites, microwave relays, modulated AC power lines, and other data transmission means known to those of skill in the art.

In addition to the computers **28**, a printer **30** and an array of disks **32** are also attached to the illustrated network **10**. Although particular individual and network computer systems and components are shown, those of skill in the art will appreciate that the present invention also works with a variety of other networks and computers.

At least some of the computers **28** are capable of using floppy drives, tape drives, optical drives or other means to read a storage medium **34**. A suitable storage medium **34** includes a magnetic, optical, or other computer-readable storage device having a specific physical substrate configuration. Suitable storage devices include floppy disks, hard disks, tape, CD-ROMs, PROMs, RAM, and other computer system storage devices. The substrate configuration represents data and instructions which cause the computer system to operate in a specific and predefined manner as described herein. Thus, the medium **34** tangibly embodies a program, functions, and/or instructions that are executable by at least two of the computers **28** to perform transaction synchronization steps of the present invention substantially as described herein.

With reference to FIG. **2**, at least two of the computers **28** are disconnectable computers **40** configured according to the present invention. Each disconnectable computer **40** includes a database manager **42** which provides a location-independent interface to a distributed hierarchical target database embedded in convergently consistent replicas **56**. Suitable databases include Novell directory services databases supported by NetWare 4.x.

A database is a collection of related objects. Each object has associated attributes, and each attribute assumes one or more values at any given time. Special values are used internally to represent NULL, NIL, EMPTY, UNKNOWN, and similar values. Each object is identified by at least one "key." Some keys are "global" in that they are normally unique within the entire database; other keys are "local" and are unique only within a proper subset of the database. A database is "hierarchical" if the objects are related by their relative position in a hierarchy, such as a file system hierarchy. Hierarchies are often represented by tree structures.

The target database includes file descriptor objects, directory descriptor objects, directory services objects, printer job objects, or other objects. The target database is distributed in that entries are kept in the replicas **56** on different computers **40**. Each replica **56** in the target database contains at least some of the same variables or records as the other replicas **56**. The values stored in different replicas **56** for a given attribute are called "corresponding values." In general, corresponding values will be equal.

However, replicas **56** at different locations (namely, on separate computers **40**) may temporarily contain different values for the same variable or record. Such inconsistencies are temporary because changes in value are propagated throughout the replicas **56** by the invention. Thus, if the changes to a particular variable or record are infrequent relative to the propagation delay, then all replicas **56** will converge until they contain the same value for that variable or record.

More generally, the present invention provides a basis for a family of distributed software applications utilizing the target database by providing capabilities which support replication, distribution, and disconnectability. In one embodiment, the database manager **42** includes one or more agents **44**, such as a File Agent, a Queue Agent, or a Hierarchy Agent. The database manager **42** hides the complexity of distribution of data from the application programs. Distributed programs make requests of the database manager **42**, which dispatches each request to an appropriate agent **44**.

Each agent **44** embodies semantic knowledge of an aspect or set of objects in the distributed target database. Under this modular approach, new agents **44** can be added to support new distributed services. For instance, assumptions and optimizations based on the semantics of the hierarchy of the NetWare File System are embedded in a Hierarchy Agent, while corresponding information about file semantics are embedded in a File Agent. In one embodiment, such semantic information is captured in files defining a schema **84** (FIG. **3**) for use by agents **44**.

The schema **84** includes a set of "attribute syntax" definitions, a set of "attribute" definitions, and a set of "object class" (also known as "class") definitions. Each attribute syntax in the schema **84** is specified by an attribute syntax name and the kind and/or range of values that can be assigned to attributes of the given attribute syntax type. Attribute syntaxes thus correspond roughly to data types such as integer, float, string, or Boolean in conventional programming languages.

Each attribute in the schema **84** has certain information associated with it. Each attribute has an attribute name and an attribute syntax type. The attribute name identifies the attribute, while the attribute syntax limits the values that are assumed by the attribute.

Each object class in the schema **84** also has certain information associated with it. Each class has a name which identifies this class, a set of super classes that identifies the other classes from which this class inherits attributes, and a set of containment classes that identifies the classes permitted to contain instances of this class.

An object is an instance of an object class. The target database contains objects that are defined according to the schema **84** and the particulars of the network **10**. Some of these objects may represent resources of the network **10**. The target database is a "hierarchical" database because the objects in the database are connected in a hierarchical tree structure. Objects in the tree that can contain other objects are called "container objects" and must be instances of a container object class.

A specific schema for the Hierarchy Agent will now be described; other agents may be defined similarly. The ndr_ddb_server class is the top level of the HA-specific database hierarchy. Since a database may contain many servers, the name is treated as a unique key for HA servers within a database.

```
CLASS  ha_server
{
  SUPERCLASS  ndr_ddb_object_header;
  PARENT      ndr_ddb_database;
  PROPERTY    NDR_OS_CLASS_FLAG_FULLY_REPLICATED;
  ATTRIBUTE
  {
    ha_server_name  server_name
    PROPERTY        NDR_OS_ATTR_FLAG_SIBLING_KEY;
  }
```

-continued

```
}
CONSTANT  HA_VOLUME_NAME_MAX = 32;
DATATYPE  ha_volume_name    STRING HA_VOLUME_NAME_MAX;
DATATYPE  ha_volume_id      BYTE;
  A volume has a name, which must be unique within the
server and can be used as the root component of a path name:
CLASS  ha_volume
{
  SUPERCLASS  ndr_dosfs_object_header;
  PARENT      ha_server;
  PROPERTY    NDR_OS_CLASS_FLAG_NAMESPACE_ROOT;
  ATTRIBUTE
  {
    ha_volume_name   volume_name
    PROPERTY         NDR_OS_ATTR_FLAG_SIBLING_KEY |
                     NDR_OS_ATTR_FLAG_IS_DOS_FILENAME;
    ha_volume_id     volume_id;
  }
}
```

In order to allocate unique volume identifiers this object holds the next free volume ID. Initially this is set to 1, so that the SYS volume can be given ID 0 when it is added to the database, in case any applications make assumptions about SYS:

```
CLASS  ha_next_volume
{
  SUPERCLASS  ndr_dosfs_object_header;
  PARENT      ha_server;
  PROPERTY    NDR_OS_CLASS_FLAG_UNREPLICATED;
  ATTRIBUTE
  {
    ndr_dosfs_dummy_key  dummy_key
    PROPERTY             NDR_OS_ATTR_FLAG_SIBLING_KEY
    COMPARISON           ndr_dosfs_dummy_key_compare
    VALIDATION           ndr_dosfs_dummy_key_validate
    ha_volume_id         next_free_volume_id;
  }
}
```

A file or directory name can be 12 (2-byte) characters long:

```
  CONSTANT  HA_FILENAME_MAX = 24;
  DATATYPE  ha_filename  STRING HA_FILENAME_MAX;
```

The ha_file_or_dir_id is a compound unique key embracing the file or directory ID that is allocated by the server, as well as the server-generated volume number. The latter is passed as a byte from class 87 NetWare Core Protocols from which it is read directly into vol (declared as a byte below). Elsewhere in the code the type ndr_host_volume_id (a UINT16) is used for the same value.

```
  DATATYPE  ha_file_or_dir_id
  {
    ULONG       file_or_dir;
    ha_volume_id  vol;
  }
```

Files and directories have many shared attributes, the most important being the file name. This must be unique for any parent directory.

```
CLASS  ha_file_or_dir
{
  PARENT       ha_directory;
  SUPERCLASS   ndr_dosfs_object_header;
  ATTRIBUTE
  {
    ha_filename          filename
    PROPERTY             NDR_OS_ATTR_FLAG_SIBLING_KEY |
                         NDR_OS_ATTR_FLAG_IS_DOS_FILENAME;
    ha_file_or_dir_id    id
    PROPERTY             NDR_OS_ATTR_FLAG_GLOBAL_KEY |
                         NDR_OS_ATTR_FLAG_UNREPLICATED
    GROUP                file_or_dir_id_group;
    ULONG                attributes;
```

-continued

```
    SHORT               creation_date;
    SHORT               creation_time;
    ndr_dcelb_auth_id   creation_id;
    SHORT               access_date;
    SHORT               archive_date;
    SHORT               archive_time;
    ndr_dcelb_auth_id   archive_id;
  }
}
```

A file has some additional attributes not present in a directory, and may contain a contents fork which can be accessed via a file distributor **90** (FIG. **3**):

```
CLASS  fa_file
{
  SUPERCLASS  fa_file_or_dir;
  PROPERTY    NDR_OS_CLASS_FLAG_DEFINE_REPLICAS |
              NDR_OS_CLASS_FLAG_HAS_PARTIALLY_REPLICATED_FILE |
              NDR_OS_CLASS_FLAG_HAS_FILE_PATH_NAME |
              NDR_OS_CLASS_FLAG_PARENT_HAS_RSC;
  ATTRIBUTE
  {
    BYTE              content_type;
    SHORT             update_date;
    property          NDR_OS_ATTR_FLAG_UNREPLICATED;
    SHORT             update_time;
    property          NDR_OS_ATTR_FLAG_UNREPLICATED;
    ndr_dcelb_auth_id update_id;
    property          NDR_OS_ATTR_FLAG_UNREPLICATED;
    ULONG             length;
    property          NDR OS ATTR FLAG_UNREPLICATED;
  }
}
```

A directory does not possess a contents fork for file distributor **90** access. The access rights mask is inherited and should be managed by like access control lists ("ACLs"):

```
CLASS  fa_directory
{
  SUPERCLASS  fa_file_or_dir;
  PROPERTY    NDR_OS_CLASS_FLAG_DEFINE_REPLICAS |
              NDR_OS_CLASS_FLAG_HAS_FILE_PATH_NAME |
              NDR_OS_CLASS_FLAG_HAS_RSC;
              //replication support object
  ATTRIBUTE
  {
    BYTE              access_rights_mask;
    SHORT             update_date;
    SHORT             update_time;
    ndr_dcelb_auth_id update_id;
    SHORT             rsc
    PROPERTY          NDR_OS_ATTR_FLAG_IS_RSC |
                      NDR_OS_ATTR_FLAG_UNREPLICATED;
  }
}
```

The root directory must appear at the top of the hierarchy below the volume. Its name is not used; the volume name is used instead. This is the top of the replication hierarchy and therefore is the top level RSC in this hierarchy:

```
CLASS  fa_root_directory
{
  SUPERCLASS  fa_directory
```

-continued

```
  PARENT      fa_volume;
  PROPERTY    NDR_OS_CLASS_FLAG_DEFINE_REPLICAS |
              NDR_OS_CLASS_FLAG_HAS_RSC;
}
```

In one embodiment, schemas such as the schema **84** are defined in a source code format and then compiled to generate C language header files and tables. The named

source file is read as a stream of lexical tokens and parsed using a recursive descent parser for a simple LL(1) syntax. Parsing an INCLUDE statement causes the included file to be read at that point. Once a full parse tree has been built (using binary nodes), the tree is walked to check for naming completeness. The tree is next walked in three passes to generate C header (.H) files for each included schema file. The header generation passes also compute information (sizes, offsets, and so forth) about the schema which is stored in Id nodes in the tree. Finally, the complete tree is walked in multiple passes to generate the schema table C source file, which is then ready for compiling and linking into an agent's executable program.

Each disconnectable computer **40** also includes a replica manager **46** which initiates and tracks location-specific updates as necessary in response to database manager **42** requests. The replica manager is discussed in detail in connection with later Figures.

A file system interface **48** on each computer **40** mediates between the replica manager **46** and a storage device and controller **54**. Suitable file system interfaces **48** include well-known interfaces **48** such as the File Allocation Table ("FAT") interfaces of various versions of the MS-DOS® operating system (MS-DOS is a registered trademark of Microsoft Corporation), the XENIX® file system (registered trademark of Microsoft Corporation), the various NOVELL file systems (trademark of Novell, Inc.), the various UNIX file systems (trademark of Santa Cruz Operations), the PCIX file system, the High Performance File System ("HPFS") used by the OS/2 operating system (OS/2 is a mark of International Business Machines Corporation), and other conventional file systems.

Suitable storage devices and respective controllers **54** include devices and controllers for the media disclosed above in connection with the storage medium **34** (FIG. 1) and other conventional devices and controllers, including non-volatile storage devices. It is understood, however, that the database replicas **56** stored on these media are not necessarily conventional even though the associated devices and controllers **54** may themselves be known in the art.

Each computer **40** also has a network link manager **50** that is capable of establishing a network connection **52** with another disconnectable computer **40**. Suitable network link managers **50** include those capable of providing remote procedure calls or an equivalent communications and control capability. One embodiment utilizes "DataTalk" remote procedure call software with extended NetWare Core Protocol calls and provides functionality according to the following interface:

rpc_init()	Initialize RPC subsystem
rpc_shutdown()	Shutdown RPC subsystem
rpc_execute()	Execute request at single location
rpc_ping()	Ping a location (testing)
rpc_claim_next_execute()	Wait until the next rpc_execute() is guaranteed to be used by this thread
rpc_free_next execute()	Allow others to use rpc_execute()

Those of skill in the art will appreciate that other remote procedure call mechanisms may also be employed according to the present invention. Suitable network connections **52** may be established using packet-based, serial, internet-compatible, local area, metropolitan area, wide area, and wireless network transmission systems and methods.

FIGS. **2** and **3** illustrate one embodiment of the replica manager **46** of the present invention. A replica distributor **70** insulates the database manager **42** from the complexities caused by having database entries stored in replicas **56** on multiple computers **40** while still allowing the database manager **42** to efficiently access and manipulate individual database objects, variables, and/or records. A replica processor **72** maintains information about the location and status of each replica **56** and ensures that the replicas **56** tend to converge.

A consistency distributor **74** and a consistency processor **76** cooperate to maintain convergent and transactional consistency of the database replicas **56**. The major processes used include an update process which determines how transaction updates are applied, an asynchronous synchronization process that asynchronously synchronizes other locations in a location set, a synchronous synchronization process that synchronously forces two locations into sync with each other, an optional concurrency process that controls distributed locking, and a merge process that adds new locations to a location set. In one embodiment, processes for synchronization and merging are implemented using background software processes with threads or similar means. The concurrency process may be replaced by a combination of retries and clash handling to reduce implementation cost and complexity.

Each location is identified by a unique location identifier. A "location sync group" is the group of all locations that a specific location synchronizes with. The location sync group for a database replica **56** on a client **20** is the client and the server **16** or other computer **28** that holds a master replica **56**; the computer **28** holding the master replica **56** is the "storage location" of the target database. The location sync group for the computer **28** that holds the master replica **56** is all computers **28** connectable to the network that hold a replica **56**. A "location set" is a set of presently connected locations in a location sync group. Locations in an "active location set" have substantially converged, while those in a "merge location set" are currently being merged into the active location set. Objects are read at a "reference location" and updated at an "update location," both of which are local when possible for performance reasons. To support concurrency control, objects require a "lock location" where they are locked for read or update; the local location is the same for all processes in a given location set.

According to one update process of the present invention, the updates for a single transaction are all executed at one update location. Each group of updates associated with a single transaction have a processor transaction identifier ("PTID") containing the location identifier of the update location and a transaction sequence number. The transaction sequence number is preferably monotonically consecutively increasing for all completed transactions at a given location, even across computer **28** restarts, so that other locations receiving updates can detect missed updates.

The PTID is included in update details written to an update log by an object processor **86**. An update log (sometimes called an "update stream") is a chronological record of operations on the database replica **56**. Although it may be prudent to keep a copy of an update log on a non-volatile storage device, this is not required. The operations will vary according to the nature of the database, but typical operations include adding objects, removing objects, modifying the values associated with an object attribute, modifying the attributes associated with an object, and moving objects relative to one another.

The PTID is also included as an attribute of each target database object to reflect the latest modification of the

object. In one embodiment, the PTID is also used to create a unique (within the target database) unique object identifier ("UOID") when a target database object is first created.

A target database object may contain attributes that can be independently updated. For instance, one user may set an archive attribute on a file while a second user independently renames the file. In such situations, an object schema **84** may define attribute groups. A separate PTID is maintained in the object for each attribute group, thereby allowing independent updates affecting different attribute groups of an object to be automatically merged without the updates being treated as a clash.

The consistency distributor **74** gathers all of the updates for a single transaction and sends them, at close transaction time, to the update location for the transaction. The consistency processor **76** on the update location writes the updates to a transaction logger **88**. In one embodiment, the transaction logger **88** buffers the updates in memory (e.g. RAM). If the update location is not local then the updates are committed to the transaction log and the PTID for the transaction is returned, so that the same updates can be buffered locally; this allows all updates to be applied in order locally. In this manner the transaction updates are applied to the update location.

An objective of one asynchronous synchronization process of the present invention is to keep the rest of the locations in the location set in sync without unacceptable impact on foreground software process performance. This is achieved by minimizing network transfers.

A process of the consistency processor **76** (such as a background software process) either periodically or on demand requests the transaction logger **88** to force write all pending transactions to the log and (eventually) to the target database. The consistency processor **76** also causes the batch of updates executed at an update location to be transmitted to all other locations in the current location set as a "SyncUpdate" request. These updates are force written to the log before they are transmitted to other locations, thereby avoiding use of the same transaction sequence number for different transactions in the event of a crash.

The SyncUpdate requests are received by other locations in the same location set and applied to their in-memory transaction logs by their respective consistency processors **76**. Each consistency processor **76** only applies SyncUpdate transactions which have sequence numbers that correspond to the next sequence number for the specified location.

The consistency processor **76** can determine if it has missed updates or received them out of order by examining the PTID. If updates are missed, the PTID of the last transaction properly received is sent to the consistency distributor **74** that sent out the updates, which then arranges to send the missing updates to whichever consistency processors **76** need them.

Acknowledged requests using threads or a similar mechanism can be used in place of unacknowledged requests sent by non-central locations. Non-central locations (those not holding a master replica **56**) only need to synchronize with one location and thus only require a small number of threads. To promote scalability, however, central locations preferably use unacknowledged broadcasts to efficiently transmit their SyncUpdate requests.

The asynchronous synchronization process causes SyncUpdate requests to be batched to minimize network transfers. However, the cost paid is timeliness. Accordingly, a synchronous synchronization process according to the present invention may be utilized to selectively speed up synchronization. The synchronous synchronization process provides a SyncUptoPTID request and response mechanism.

In one embodiment, the SyncUptoPTID mechanism utilizes a SyncState structure which is maintained as part of a location state structure or location list that is managed by a location state processor **80** in the memory of each computer **28**. The SyncState structure for a given location contains a location identifier and corresponding transaction sequence number for the most recent successful transaction applied from that location. The SyncState structure is initialized from the update log at startup time and updated in memory as new transactions are applied.

A SyncUptoPTID request asks a destination to bring itself up to date with a source location according to a PTID. The destination sends a copy of the SyncState structure for the source location to that source location. The source location then sends SyncUpdate requests to the destination location, as previously described, up to an including the request with the PTID that was specified in the SyncUptoPTID request. In a preferred embodiment, the central server is a NetWare server and the SyncUptoPTID requirements are approximately 100 bytes per location, so scalability is not a significant problem for most systems.

A merge process according to the present invention includes merging location sets when disconnected disconnectable computers are first connected or reconnected. For instance, merging location sets normally occurs when a computer new to the network starts up and merges into an existing location set. Merging can also happen when two sets of computers become connected, such as when a router starts. Merging can also occur when requested by a user, when the network load drops below a predetermined threshold for a predetermined period of time, or on a scheduled basis, such as every night at 1 AM.

Merging occurs when two replicas **56** are resynchronized after the computers **28** on which the replicas **56** reside are reconnected following a period of disconnection. Either or both of the computers **28** may have been shut down during the disconnection. A set of updates are "merged atomically" if they are merged transactionally on an all-or-nothing basis. A distributed database is "centrally synchronized" if one computer **28**, sometimes denoted the "central server," carries a "master replica" with which all merges are performed.

Portions of the master replica or portions of another replica **56** may be "shadowed" during a merge. A shadow replica, sometimes called a "shadow database", is a temporary copy of at least a portion of the database. The shadow database is used as a workspace until it can be determined whether changes made in the workspace are consistent and thus can all be made in the shadowed replica, or are inconsistent and so must all be discarded. The shadow database uses an "orthogonal name space." That is, names used in the shadow database follow a naming convention which guarantees that they will never be confused with names in the shadowed database.

A "state-based" approach to merging compares the final state of two replicas **56** and modifies one or both replicas **56** to make corresponding values equal. A "log-based" or "transaction-based" approach to merging incrementally applies successive updates made on a first computer **28** to the replica **56** stored on a second computer **28**, and then repeats the process with the first computer's replica **56** and the second computer's update log. A hybrid approach uses state comparison to generate an update stream that is then applied incrementally. The present invention preferably utilizes transaction-based merging rather than state-based or hybrid merging.

As an illustration, consider the process of merging a single new location A with a location set containing locations B and C. In one embodiment, the following performance goals are satisfied:

(a) Use of locations B and C is not substantially interrupted by synchronization of the out-of-date location A with B and C; and

(b) Users connected to location A (possibly including multiple users if location B is a gateway) are able to see the contents of the other locations in the set within a reasonable period of time.

Merging typically occurs in three phases. During a "merging out" phase location A sends newer updates to location B. For instance, if A's location list contains PTID 50:14 (location identifier:transaction sequence number) and B's location list contains PTID 50:10, then the newer updates sent would correspond to PTID values 50:11 through 50:14.

During a "merging in" phase new updates in the merge location B are merged into A's location. For instance, suppose A's location list contains PTIDs 100:12 and 150:13 and B's location list contains PTIDs 100:18 and 150:13. Then the new updates would correspond to PTID values 100:13 through 100:18. If updates are in progress when merging is attempted, the initial attempt to merge will not fully succeed, and additional iterations of the merging in and merging out steps are performed.

In one embodiment, merging does not include file contents synchronization. Instead file contents are merged later, either by a background process or on demand triggered by file access. This reduces the time required for merging and promotes satisfaction of the two performance goals identified above. In embodiments tailored to "slow" links, merging is preferably on-going to take advantage of whatever bandwidth is available without substantially degrading the perceived performance of other processes running on the disconnectable computers.

In embodiments employing an update log, the log is preferably compressed prior to merging. Compression reduces the number of operations stored in the log. Compression may involve removing updates from the log, altering the parameters associated with an operation in a given update, and/or changing the order in which updates are stored in the log.

In one embodiment, all Object Database calls come through the consistency distributor **74**, which manages distributed transaction processing and maintains consistency between locations. Almost all calls from a location distributor **78** are made via the consistency distributor **74** because the consistency distributor **74** supports a consistent view of the locations and the database replicas **56** on them.

The consistency distributor **74** and an object distributor **82** support multiple concurrent transactions. This is needed internally to allow background threads to be concurrently executing synchronization updates. It could also be used to support multiple concurrent gateway users. In an alternative embodiment, multiple concurrent transactions on the same session is supported through the consistency distributor **74**.

In one embodiment, the consistency distributor **74** and the consistency processor **76** are implemented in the C programming language as a set of files which provide the functionality described here. Files CD.H and CD.C implement part of the consistency distributor **74**. A separate module having files CD_BG.H and CD_BG.C is responsible for background processes associated with merging and synchronization. A module having files CDI.H and CDI.C contains functions used by both the CD and CD_BG modules. These modules provide functionality according to the following interface:

cd_init	Init CD
cd_shutdown	Shutdown CD
cd_create_replica	Create a replica of a specified database
cd_remove_replica	Remove a replica of a specified database
cd_load_db	Load an existing database
cd_unload_db	Unload an existing database
cd_merge_start	Start merge of active and merge location sets
cd_merge_stop	Stop merge
cd_start_txn	Start a CD transaction
cd_set_txn_ref_loc	Set reference/update list (location identifier) for txn (transaction)
cd_get_txn_desc	Get a txn descriptor given a txn id
cd_abort_txn	Abort a CD transaction
cd_end_txn	End a CD transaction
cd_commit	Commit all previously closed txns to disk
cd_execute_txn	Execute locks and updates for a txn
cd_read	Do read or lookup request
cd_readn	Do readn
cd_lookup_by_uoid	Do lookup using UOID
cd_add_lock	Add an object or agent lock
cd_remove_lock	Remove an object or agent lock
cd_modify_attribute	Modify a single attribute in a previously read object
cd_init_new_doid	Setup all fields in a new doid
cd_add	Add a new object
cd_remove	Remove an object
cd_move	Move an object
cd_set_marker	Add marker point in txn
cd_revert_to_marker	Revert txn state to last marker
cd_get_effective_access_right	Get the effective access rights for the current session and object
cd_convert_uoid2doid	Convert UOID to DOID
cd_sync_object	Get the server to send a newly replicated object
cdi_bg_init	Initialize CD background processes
cd_bg_merge	Execute a background merge
cd_bg_sync_remote_upto_ptid	Bring remote location up to date with local PTID
cdi_init	
cdi_shutdown	
cdi_execute_ack_sys	Execute acknowledged request using system session
cdi_execute_ack	Execute acknowledged request
cdi_apply_locks	Apply locks for txn
cdi_abort_get_txn	Remove all locks already set for a txn
//Forced update location (used to change update location when executing clash handler functions)	
cdi_register_forced_update_location	Register location to be used as update location for thread
cdi_unregister_forced_update_location	Unregister location to be used as update location for thread
cdi_get_forced_update_location	Get forced update location for thread
cdi_sync_upto_ptid	Bring location up to date with PTID
cdi_sync_upto_now	Bring location up to date with latest PTID
cdi_sync_loc_list	Make any location list consistent with destination location list and return info on mismatch of PTIDs
cdi_read_loc_list	Read location list
cdi_sync_upto_a_ptid	Bring location up to date with PTID

Since updates are cached during a transaction, special handling of reads performed when updates are cached is

required. In one embodiment, the caller of cd_read( ) or cd_read( ) sees the results of all updates previously executed in the transaction. In an alternative embodiment, for cd_read( ) reads will see all previously added objects and will see the modified attributes of objects, but will not see the effects of moves or removes. Thus if an object is removed during a transaction the read will behave as if it has not been removed. The same is true for moved objects. Modifications to keys will have no effect on reads using the keys. The cd_read( ) function behaves as if none of the updates in the current transaction have been applied.

In one embodiment, the consistency processor **76**, which processes all distributed object database requests, includes background processes that manage object database updates on local locations and synchronization of locations. Within this embodiment, a CP module contains a dispatcher for all requests which call functions that have a prefix of "cpXX_"; a CPR module processes read requests; a CPU module processes update and lock requests; a CPSM module processes synchronization and merging requests; a CP_BG module controls background processing which includes scheduling multiple background threads, controlling the state of all local locations and synchronization of local locations with local and remote locations; and a CPUI module provides functions that are shared by the CP_BG and CPx modules. These modules provide functionality according to the following interface:

cp_init	Includes performing recovering of input locations and recovery of TL (transaction logger 88) and OP (object processor 86)
cp_shutdown	Shutdown CP
cp_process	Process a consistency request
cp_clear_stats	Reset CP statistics
cp_dump_stats	Dump CP statistics to the log
cpr_process_read	Process OP read or lookup request
cpr_process_readv	Process readv request
cpu_register_ptid	Register use of a PTID at a reference location
cpu_execute_txn	Execute single txn at reference location
cpu_commit	Commit all txns for session
cpu_add_locks	Add list of locks
cpu_remove_locks	Remove list of locks
cpu_abort_pre_txn	Remove object locks for specified transaction
cpsm_sync_upto_ptid	Bring remote locations up to date as far as given PTID
cpsm_get_latest_ptid	Obtain the latest PTID
cpsm_get_sync_object	Remote machine wants to sync a newly replicated object
cpsm_sync_object	Add a newly replicated object to the local database
cpsm_get_sync_update	Get a local sync_update
cpsm_sync_update	Apply multiple update txns to location
cpsm_read_loc_list	Read list of locations and states
cpsm_sync_loc_list	Sync location list
cpsm_merge_loc_list	Attempt to merge my location list with other location list
cpsm_sync_finished	Remote machine is notifying us that a sync_upto_ptid has completed
cpsm_request_merge	Request a merge of this location with the control server
cpui_init	Initialize internal structures
cpui_shutdown	Shutdown CPUI subsystem
cpui_execute_txn	Execute update txn at a local location
cpui_apply_update_list_to_db	Apply an update list to an OP database
cpui_commit	Commit all txns at location
cpui_flush	Flush all txns to object database at location
cpui_replay_logged_transactions	Replay transactions from the log that have not been committed to OP
cp_bg_init	Initialize CP_BG subsystem
cp_bg_shutdown	Shutdown CP_BG subsystem
cp_bg_handle_distributed_request	Handle a request that requires remote communication
cp_bg_notify_close_txn	Notify CP_BG of a closed transaction
cp_bg_notify_commit	Notify CP_BG that all txns are committed at a location
cp_bg_attempt_send_flush	Attempt to send out and flush txns
cp_bg_notify_load	Notify CP_BG of a newly loaded DB
cp_bg_notify_unload	Notify CP_BG of a newly unloaded DB
cp_bg_flush_upto_ptid	Force all transactions upto the specified ptid to the lowpoint state

The location distributor **78** in each replica manager **46** and the location state processor **80** are used to determine the storage locations of database entries. In one embodiment, the location state processor **80** uses a cache of the current state of locations and maintains state information on the merging process. The location state processor **80** is responsible for processing remote requests which pertain to the location list.

All locations that are up at any time within a sync group are in either the ACTIVE or MERGE location sets. The ACTIVE location set contains all locations that are in sync with the local location up to certain sync watermarks. The MERGE location set contains all nodes that are not in sync with the local location, either through not having updates the active set does have, or through having updates the active set does not have.

Locations in the MERGE set enter the ACTIVE set through the two-way merging process described above, under control of the consistency distributor **74** and the consistency processor **76**. Once in the ACTIVE set, a location should never leave it until the location goes down.

Each location continuously sends out its local updates to other members of its active location set as part of the merging process. The PTID in a location's log that was last sent out in this manner is called the location's "low watermark" PTID. For a location to enter the active set it must have all PTIDS in its local log up to the low watermark PTID; only the merging process used to move a location from the MERGE to the ACTIVE location set is capable of propagating early transactions. Each location also maintains a "high watermark" PTID which is the last transaction (in local log order) that has been committed, and is thus a candidate for sending out in a background sync update.

The replica managers **46** track the last transaction sequence number made by every location up to the low watermark PTID in order to know whether a location is up to date with another location's low watermark. The log ordering may be different in different locations, up to an interleave.

One embodiment of the location state processor **80** provides functionality according to the following interface:

ls_init	Initialize LS
ls_shutdown	Shutdown LS
ls_close_db	Close out all entries for a database

-continued

ls_allocate_new_lid	Allocate a new location identifier for use by a new replica
ls_add	Add a new location
ls_remove	ReTlove a location
ls_modify_local_lid	Modify a location entry's local transaction identifier (sequence number)
ls_modify_state	Modify a location entry's state
ls_get_loc_list	Get list of locations
ls_get_loc_sync_list	Get list of locations for syncing
ls_get_next_loc	Get next location
ls_get_first_in_loc_list	Get first location in list that is in current location set
ls_get_loc_entry	Get location entry given lid (location identifier)
ls_get_first_ref_loc	Get nearest reference location in provided list
ls_get_first_ref_loc_in_list	Get first reference location in provided list
ls_get_lock_loc	Get lock location for location set
ls_higher_priority	Determine which location has highest priority
ls_complete_merge	Complete the merge process
ls_set_sync_watermarks	Set the high and low watermark PIDs used in syncing and merging

The object distributor **82** manages ACLs and otherwise manages access to objects in the database. In one embodiment, the object distributor **82** provides functionality according to this interface:

```
typedef void* sdr_od_db_handle; //open database handle
//list <strong(AIX,sdr_od_txn_id)
//object distributor transaction instance identifier
typedef void* sdr_od_txn_id;
#define SDR_OD_INVALID_TXN_ID (sdr_od_txn_id)0
typedef struct //Txn info returned by NdrOdGetTxnInfo
{
  sdr_od_db_handle      db;          /* database */
  sdr_dodh_session_type session;     /* session */
} sdr_d_txn_info;
//Start a new client txn for this session
sdr_rst EXPORT
NdrOdStartClientTxn(
  sdr_od_db_handle      db_handle,
  /* -> Handle to the open DB */
  sdr_dodh_session_type session,     /* ->session */
  sdr_od_txn_id         *txn_id);    /* <- txn id */
//Find out what databases are available
sdr_rst EXPORT
NdrOdEnumerateDBs(
  sdr_od_enum_flags     flags,
  /* -> Determines which databases are included in search */
  sdr_os_db_name        search_name,
  /* -> The database name (may be wild) */
  sdr_os_db_type_name   search_type,
  /* -> The database type (may be wild) */
  sdr_dodh_database_id_type search_id,
  /* -> The database id (may be wild) */
  sdr_os_db_name        name,
  /* <- The database name */
  sdr_os_db_type_name   type,
  /* <- The database type */
  sdr_dodh_database_id_type *id,
  /* <- The database id */
  UINT16                *index);
  /* <-> Set to 0 to start else use
  previous returned value */
//Start a new txn for this session
sdr_rst EXPORT
NdrOdStartTxn(
  sdr_od_db_handle      db_handle,
  /* -> Handle to the open DB */
  sdr_dodh_session_type session,
  /* -> session */
  sdr_od_txn_id         *txn_id);
  /* <- txn id */
```

The interface includes NdrOdCloseTxn( ), which closes updates for the current transaction and causes all updates since the last NdrOdStartTxn( ) call to be applied. Either all updates will be applied, or none will be applied. NdrOdCloseTxn( ) does not commit the updates, that is, they are not written to disk. NdrOdCommit( ) is used to commit closed updates to disk. However, after calling NdrOdCloseTxn( ), no further updates may be applied in the transaction. This function is also where all the locking and updates previously cached actually get done. Consequently, most locking and/or consistency errors are reported here (after synchronization) so that the transaction can be retried.

```
sdr_rst EXPORT
NdrOdCloseTxn(sdr_od_txn_id      txn_id);    /* -> txn_id */
```

The NdrOdEndTxn( ) function ends the current transaction and executes an implicit NdrOdCloseTxn( ). No error is returned if no transaction is currently open:

```
sdr_rst EXPORT
NdrOdEndTxn(sdr_od_txn_id      txn_id);    /* -> txn id */
```

The NdrOdCommit function commits all previously closed transactions for the session to disk:

```
sdr_rst EXPORT
NdrOdCommit(
  sdr_od_db_handle      db,      /* -> DB to commit */
  sdr_dodh_session_type session); /* -> session */
  The interface also includes the following functions:
//Abort current txn
sdr_rst EXPORT
NdrOdAbortTxn(sdr_od_txn_id      txn_id);    /* -> txn_id
*/
//Get info on current txn
sdr_rst EXPORT
NdrOdGetTxnInfo(
  sdr_od_txn_id      txn_id,     /* -> txn_id */
  sdr_od_txn_info*   txn_info);  /* <- txn info */
//Lookup an object using parent Distributed Object Identifier
//DOID; encodes location info to make it routing distributor
//requests to the right machine; includes DOID of sibling key
or
//using global key; the key value MUST be a configuration
structure.
sdr_rst EXPORT
NdrodLookupByKey(
  sdr_od_txn_id      txn_id,     /* -> txn_id */
  sdr_dc_db_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  sdr_os_class       class_id,
  /* -> Class id of superclass to match*/
  /* Acts as filter when key contents wildcard. */
  sdr_dodh_doid_class* parent_doid,  /* -> Parent DOID
*/
  sdr_os_attribute   key_id,
  /* -> Type of unique key */
  UINT16             key_length,
  /* -> Length, in bytes, of the key value */
  VOID*              key,            /* ->Key value
*/
  sdr_dodh_doid_class* doid);
  /* <- Pointer to returned DOID of object */
//Lookup an object using DOID
//This checks the existence of the object and updates its DOID
```

-continued

```
ndr_int EXPORT
NdrOdLookup(
  ndr_od_txn_id      txn_id,      /* <txn_id */
  ndr_dodb_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_dodb_doid_class*       doid,      /* -> DOID */
  ndr_dodb_doid_class* new_doid);
  /* <- Updated DOID of object */
//Lookup an object's parent using DOID
ndr_int EXPORT
NdrOdLookupParent(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_dodb_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_dodb_doid_class* doid,      /* -> DOID */
  ndr_dodb_doid_class* parent_doid);
  /* <- Parent DOID of object */
//Read an object using parent DOID and sibling key or using
//global key. It's always OK to read an object with an out of
//date parent doid as the parent's list is not used to get the
//reference location. The key value MUST be a contiguous
//structure
ndr_int EXPORT
NdrOdReadByKey(
  ndr_od_txn_id      txn_id,      /* -> txn id */
  ndr_dodb_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_os_class       class_id,
  /* -> Class id. of superclass to match */
  /*       and superclass structures to be returned */
  ndr_dodb_doid_class* parent_doid, /* -> Parent DOID */
  ndr_os_attribute       key_id, /* -> Type of unique key */
  UINT16             key_length,
  /* -> Length, in bytes, of the key value */
  VOID*              key,      /* -> Key value */
  UINT16             max_length,
  /* -> Max length of data read */
  UINT16*            length,
  /* <- Final length of data read */
  ndr_os_object*     object);
  /* -> Pointer to object buffer */
//Read an object using DOID
ndr_int EXPORT
NdrOdRead(
  ndr_od_txn_id      txn_id,      /* -> txn id */
  ndr_do_db_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_os_class       class_id,
  /* -> Class id. of superclass to match */
  /*       and superclass structures to be returned */
  ndr_dodb_doid_class* doid,      /* -> DOID */
  UINT16             max_length,
  /* -> Max length of data read */
  UINT16*            length,
  /* <- Final length of data read */
  ndr_os_object*     object);
  /* -> Pointer to object buffer */
```

An NdrOdReadn( ) function which reads multiple objects using parent DOID and wildcards behaves as if none of the updates in the transaction have been applied. Interpretation of wildcard values in the key is done by registered keying functions. NdrOdReadn( ) reads either up to max_objects, or up to the maximum number of objects that will fit in the max_length object buffer:

```
ndr_int EXPORT
NdrOdReadn(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_do_db_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_os_class       class_id,
  /* -> Class id. of superclass to match
  and superclass structures to be returned */
  ndr_os_class       read_as_class,
  Class id. target objects are to be read as */
  ndr_dodb_doid_class* parent_doid,      /* -> Parent DOID */
  ndr_os_attribute       key_id,      /* -> Type of unique key */
  UINT16             key_length,
  /* -> Length, in bytes, of the key value */
  VOID*              key,
  /* ->Key value to match, can contain wildcard.
  NULL implies TO match all objects under parent containing
  the key id */
  UINT16             max_length,
  /* ->Max length of data read */
  UINT16*            length,
  /* <- Final length of data read */
  ndr_dodb_object_list*      object_list,
  /* -> Pointer to object buffer */
  UINT16             max_objects,
  /* -> Max number of objects read. Use OD_MAX_OBJECTS to
  read max that will fit in buffer */
  ndr_dodb_context_type*     context);
  /* <-> set to DODB_CONTEXT_START to start a new read,
  or a previously returned context to continue a previous
  read. <- set to DODB_CONTEXT_END if all objects read,
  or a value that can be used to continue reading at the
  next object */
#define  NDR_OD_MAX_OBJECTS     0xFFFF
```

The NdrOdLock( ) function explicitly adds an exclusive or shared lock to an object using the object's DOID. The lock call is called implicitly for all updates, but should be called explicitly if read locks are required. The lock is only taken when the transaction is initially executed. It is not executed when the update is merged. The lock is applied at the end of a transaction. If it fails the transaction is aborted and should be re-tried by the caller. One embodiment does not utilize locks to control concurrency but instead relies on retries and clash handling:

```
ndr_int EXPORT
NdrOdLock(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_dodb_doid_class* doid,      /* -> Object's DOID */
  BOOLEAN            is_exclusive);
  /* -> TRUE => take exclusive lock */
  The intent was also includes:
//Add agent defined lock to object
ndr_int EXPORT
NdrOdAddAgentLock(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_dodb_doid_class* doid,      /* -> Object's DOID */
  ndr_dodb_lock_type       lock_type,
  /* -> Type of lock */
  ndr_dodb_lock_flags_type lock_flags,
  /* -> Flags that allow multiple locks to be taken
  in single call. Each bit corresponds to a separate
  lock, e.g. used for read/write flags on file open */
  ndr_dodb_lock_deny_flags_type deny_flags);
  /* -> Bits set that correspond to lock_flags bits
  causes the corresponding lock to be denied */
//Remove agent defined lock
ndr_int EXPORT
NdrOdRemoveAgentLock(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_dodb_doid_class* doid,      /* -> Object's DOID */
  ndr_dodb_lock_type       lock_type);
  /* -> Type of lock */
```

The following four calls are used to append various types of updates onto an open transaction. Any of them may return NDR_OK indicating success, NDR_CD_EXCEEDED_TXN_LIMITS indicating that transaction limits have been exceeded, or some other error indicator. In the case of exceeded transaction limits the transaction state will not have been changed and the failed call will have had no effect. The caller is expected to commit or abort the transaction as appropriate. In all other error cases the transaction

is automatically aborted before returning the error to the caller:

```
//Modify a single attribute in a previously read object
//The object distributor caches the modifications and only
//applies them at above txn time
ndr_int EXPORT
NdrOdModifyAttribute(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_double_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_double_doid_class* doid,
  /* ->      DOID of previous read version of object.
  Used to verify object has not been modified by another
  user since previously read */
  ndr_os_attribute      attribute_id,
  /* -> Identifies attribute to be modified
  VOID*      value);      /* -> New attribute value */
//Add a new object
//The DOID attribute does not need to be filled in by the
caller.
//The DOID will be set up before writing the object to the
//database.
ndr_int EXPORT
NdrOdAdd(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_double_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_double_doid_class* parent doid,      /* -> Parent DOID */
  ndr_os_class      class_id,
  /* -> Class Id of object */
  ndr_os_object*      object)
  /* -> Pointer to agent object */
//Remove an object using DOID
ndr_int EXPORT
NdrOdRemove(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_double_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_double_doid_class* doid);      /* -> DOID */
//Move an object using DOID
ndr_int EXPORT
NdrOdMove(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndrodedb_access_rights_type rights_needed_on_parent,
  /* -> rights needed on parent */
  ndr_double_doid_class* doid,      /* -> DOID */
  ndr_double_doid_class* target_parent_doid);
  /* -> Target parent DOID */
//Set a marker in an open transaction. The state of the
//transaction at the time the marker is set can be reverted
//to at any time before the transaction is closed by
//calling NdrOdRevertToMarker().
//Only the last marker in a transaction is significant.
//This call may return NDR_CD_EXCEEDED_TXN_LIMITS which
//should be treated as for the update appending calls above
ndr_int EXPORT
NdrOdSetMarker(ndr_od_txn_id      txn id);      /* -> txn_id */
//Revert a txn's state to the last previously marked state
ndr_int EXPORT
NdrOdRevertToMarker(ndr_od_txn_id      txn_id);      /* -> txn_id */
//Add a <user-id, rights-mask> pair to an object's
//access rights, overwriting any previous rights-mask for
//that user
ndr_int EXPORT
NdrOdAddAccessRight(
  ndr_od_txn_id      txn id,      /* -> txn_id */
  ndr_double_doid_class*      doid,      /* -> Object DOID */
  ndr_double_auth_id_type      user,
  /* -> User to whom rights are to be granted */
  ndr_double_access_rights_type rights,
  /* -> Rights to be granted to that user */
//Remove any <user-id, rights-masks> pair from an object's
//access rights for a given user-id
ndr_int EXPORT
NdrOdRemoveAccessRight(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_double_doid_class*      doid,      /* -> Object DOID */
  ndr_double_auth_id_type      user);
  /* -> User whose rights are to be revoked */
```

```
//Get the array of all <user-id, rights-mask> pairs for an
object
ndr_int EXPORT
NdrOdGetAccessRights(
  ndr_od_txn_id      txn id,      /* -> txn_id */
  ndr_double_doid_class* doid,      /* -> Object DOID */
  UINT16*      act_count,
  /* <- Number of ACL entries for this object */
  ndr_double_acl_element_type* acl);
  /* <- Rights information for that object */
//Get the effective access rights for the current session
//for an object
ndr_int EXPORT
NdrOdGetEffectiveAccessRight(
  ndr_od_txn_id      txn_id,      /* -> txn_id */
  ndr_double_doid_class*      doid,      /* -> Object DOID */
  ndr_double_access_rights_type* rights);
  /* <- Effective rights for the current session */
//Convert UDID to DOID
ndr_int EXPORT
NdrOdConvertUdid2Doid(
  ndr_os_class      class_id,
  /* -> Class id. of object */
  ndr_double_uuid_type*      udid,      /* -> UDID */
  ndr_double_doid_class*      doid);      /* <- Updated DOID */
//Convert UDID to DOID
ndr_int EXPORT
NdrOdConvertUdid2LocalDoid(
  ndr_os_class      class_id,
  /* -> Class id. of object */
  ndr_double_lkd_type      location,
  /* -> Location on which object exists */
  ndr_double_uuid_type*      udid,      /* -> UDID */
  ndr_double_doid_class*      doid);      /* <- Updated DOID */
```

The object processor **86** provides a local hierarchical object-oriented database for objects whose syntax is defined in the object schema **84**. In one embodiment, the object processor **86** is built as a layered structure providing functionality according to an interface to the structure which is described below. The embodiment also includes a module for object attribute semantics processing, a set of global secondary indexes, a hierarchy manager, a B-tree manager, a record manager, and a page manager. Suitable modules and managers are readily obtained or constructed by those familiar with database internals. A brief description of the various components follows.

The page manager provides functionality according to a logical file interface of free-form fixed length pages addressed by logical page number. Rollback and commit at this level provide anti-crash recovery.

The record manager provides for the packing of variable length keyed records into fixed length pages.

The B-tree manager uses the facilities of the record and page managers to provide general B-trees supporting variable length records and variable length keys.

The hierarchy manager imposes a hierarchical structure on records by use of structured B-tree keys and a global UOID->full name index.

The secondary index manager provides generalized global indexing capabilities to records.

The attribute manager interprets the schema **84** in order to raise the interface of the object processor **86** from a record-level to an object-level interface.

The interface module of the object processor **86** uses lower level interfaces to provide functionality according to the following interface:

op_Init	Initializes object processor
op_Shutdown	Shuts down object processor

-continued

op_add_database	Creates a new volume
op_mount_database	Mounts a specified volume for use
op_dismount_database	Dismounts the specified volume
op_remove_database	Removes a specified volume (permanently)
op_read	Read an object by UOID
op_readv	Read one or more objects with wildcards
op_execute_update_list	Apply one or more updates
op_commit	Commit updates to a specified volume
op_rollback	Rollback to the last committed state
op_free_fnversion_list	Free up an fnversion list returned from update execution
op_clear_stats	Clear object processor statistics
op_dump_stats	Dump statistics to the log

Due to higher level requirements of trigger functions in a set of trigger function registrations **94**, in one embodiment it is necessary to have the old values of modified attributes available on a selective basis. This is done by means of a 'preservation list' produced by op_execute_updates( ). The preservation list contains an update list specifying old attribute values for all executed updates that require it (as determined by a callback function), together with pointers to the original causative updates. These updates may not actually be present in the input update list, as in the case of an object removal that generates removes for any descendant objects it may have. Preservation lists reside in object processor **86** memory and must thus be freed up by the caller as soon as they are no longer needed.

The transaction logger **88** provides a generic transaction log subsystem. The logs maintained by the logger **88** provide keyed access to transaction updates keyed according to location identifier and processor transaction identifier (PTID). In one embodiment, a non-write-through cache is used to batch uncommitted transaction updates.

The transaction logger **88** is used by the consistency processor **76** to support fast recovery after a crash. Recovery causes the target database to be updated with any transactions that were committed to the log by the logger **88** but were not written to the target database. The log file header contains a "shutdown OK" flag which is used on startup to determine if recovery is required for the location.

The transaction logger **88** is also used by the consistency processor **76** to support fast synchronization. The update log created by the logger **88** is used to replay the updates from one location to a second location using minimal disk and network **10** transfers.

The file distributor **90** distributes file contents to appropriate locations in the network **10**. A file processor **92** supports each file distributor **90** by carrying out requested read, write, lock, or other operations locally.

The file distributor **90** hides from agents the complexities caused by the distributed nature of files. To the extent possible, the interface portion of the file distributor **90** resembles file system interfaces that are familiar in the art. An open file is denoted by a numeric fork_id and functions are provided to read, write, open, and otherwise manipulate and manage files and their contents.

However, a class in the schema **84** can be given a REPLICATED_FILE property. Whenever an object of such a class is created in the database, a distributed file is created by the file distributor **90** and file processor **92** to hold the file contents associated with that object. For instance, the Hierarchy Agent might create such an object to denote a leaf node in the directory hierarchy. In short, in one embodiment the file distributor **90** neither has nor needs an explicit externally called mechanism for creating files.

Moreover, the distributed file is deleted from storage when the corresponding object is deleted from the database. The locations at which the file is stored are precisely those at which the object exists. When a file with more than one replica **56** is modified and closed, the file distributors **90** and file processors **92** at the various locations holding the replicas **56** ensure that all replicas **56** of the file receive the new contents. It is not necessary for the agent to expressly manage any aspect of file content distribution.

A distributed file is identified by the UOID of the corresponding object; no built-in hierarchical naming scheme is used. A transaction identifier is also required when opening a file, to identify the session for which the file is to be opened. In one embodiment, the file distributor **90** and file processor **92** provide functionality according to the following interface:

```
//An ndr_fd_fork_id is the id by which an FD open fork is known
typedef SINT16 ndr_fd_fork_id;
#define  NDR_FD_NOT_A_FORK_ID (-1)
//An ndr_fd_open_mode is a bit-mask which specifies whether a
//fork is open for reading and/or writing
typedef UINT16 ndr_fd_open_mode;
#define  NDR_FD_OPEN_READ_MODE                0x0001
#define  NDR_FD_OPEN_WRITE_MODE               0x0002
#define  NDR_FD_OPEN_EXCL_MODE                0x0004
#define  NDR_FD_OPEN_EXTERNAL_MODES           0x0007
//The remaining open modes are private to the replica manager:
#define  NDR_FD_OPEN_SYNC_MODE                0x0008
#define  NDR_FD_OPEN_CLOSE_ON_EOF_MODE        0x0010
#define  NDR_FD_OPEN_READ_NOW                 0x0020
```

In one alternative embodiment, opening a file with an NdrFdOpenFile( ) function returns pointers to two functions together with a separate fork_id for use with these two functions only. These pointers are of the type ndr_fd_io_function, and may be used as alternatives to NdrFdReadFile( ) and NdrFdWriteFile( ) when accessing that open file only. The functions should be at least as efficient as NdrFdReadFile( ) and NdrFdWriteFile( ) and will be significantly faster when the file access is to a local location. Their use does require that the caller maintain a mapping from the open fork id onto these function pointers. For this reason, NdrFdReadFile( ) and NdrFdWriteFile( ) should always be available for all open files in this alternative embodiment:

```
typedef ndr_int EXPORT (*ndr_fd_io_function)(
  ndr_fd_fork_id      fork_id,      /* -> fd of open fork */
  UINT32              offset,
  /* -> Offset at which to start reading */
  UINT16*             length,
  /* <-> desired length on entry, actual length on
  exit. These will only differ if an error
  is encountered (such as end of file) */
  UINT8*              data,
  /* <-> Data read or written */
  ndr_os_txn_id       txn_id);      /* -> txn_id */
```

A "clash" occurs during synchronization when two desired changes to the database are inconsistent. Clashes arise from "independent" updates, namely, updates performed on separate replicas **56** while the computers holding the replicas **56** were disconnected. Thus, clashes always take place between a pair of "clashing updates" which together define a "clash condition." A "repairing update" is an update that removes a clash condition caused by a clashing update.

A "transient clash" is a clash that is not present in the final states of the two replicas **56** being merged. Transient clashes only arise when log-based or hybrid merging is used. For

instance, suppose two users each create a file of a given name at two locations **36**, **38** while those locations are disconnected. The user at the first location **36** then deletes (or renames or moves) the file in question before reconnection such that it no longer clashes with anything on the second location **38**. On merging the replicas **56** of the two locations **36**, **38**, the original add update for the file from the first location **36** will clash with the replica **56** of the second location **38**, yet the final result of applying the update stream from the first location **36** to the replica **56** on the second location **38** is a state that is compatible with that replica **56**.

By contrast, "persistent clashes" create inconsistencies that are present in the final states of two replicas **56**. A clash whose type is unknown is a "potential clash."

A "file contents clash" occurs when a file's contents have been independently modified on two computers **28**, or when a file has been removed from one replica **56** and the file's contents have been independently modified on another replica **56**.

An "incompatible manipulation clash" occurs when an object's attributes have been independently modified, when an object has been removed in one replica **56** and the object's attributes have been modified in another replica **56**, when an object has been removed in one replica **56** and moved in the hierarchy in another replica **56**, when a parent object such as a file directory has been removed in one replica **56** and has been given a child object in another replica **56**, or when an object has been independently moved in different ways. Thus, although clashes are discussed here in connection with files and the file distributor **90**, clashes are not limited to updates involving files.

A "unique key clash" occurs when two different objects are given the same key and both objects reside in a portion of the database in which that key should be unique. In a database representing a file system hierarchy, for instance, operations that add, move, or modify files or directories may create a file or directory in one replica **56** that clashes on reconnection with a different but identically-named file or directory in another replica **56**.

A "permission clash" occurs when a change in file access or modification permissions that is made to a central server replica **56** would prohibit an independent update made to a mobile or client computer replica **56** from being applied to the server replica **56**. A permission clash is an example of an "external clash," namely, a clash detected by reference to a structure external to the database. Permission clashes and other external clashes may be detected by trigger functions.

A "grouped attribute" is a database object attribute that is associated with other database object attributes such that changing the value of any attribute in a group creates a clash with the other attributes in the group. For instance, filename and rename-inhibit attributes are preferably grouped together, while filename and file-access-date attributes are preferably not grouped together. Without attribute grouping, a change to any attribute of an object is assumed to clash with a change to any other attribute of the object or another change to the same attribute.

"Eliminating a clash" means identifying the basis for the clash and eliminating it. "Recovering from a clash" means identifying the basis for the clash and either eliminating that basis or presenting alternative resolutions of the clash to a user to choose from. "Regressing an update" means undoing the update on at least one replica **56**. Creating a "recovery item" means creating a duplicate object in a shadow database and then remapping uses of the recovery item's key so that subsequent updates are performed on the recovery item instead of the original object. If the database represents a file system hierarchy, recovery items may be gathered in a "single directory hierarchy" or "recovery directory" that contains a directory at the root of the volume, recovered items, and copies of any directories necessary to connect the recovered items properly with the root.

A clash handler function of one of the types below can be registered with the file distributor **90** for a database type to be called whenever the file distributor **90** detects a clash caused by disconnected modification or removal of a file's contents. The parameters are those of a regular clash handler plus the object DOID with NDR_OS_CLASS_FLAG_HAS_PARTIALLY_REPLICATED_FILE property (the file object defined by the object schema **84**) and possibly a duplicated object return:

```
//Call back to a user in respect of clashes detected at the
//database level
typedef ndr_int32 EXPORT (*ndr_fd_object_clash_fn)(
  ndr_od_db_handle      db,     /* -> Database */
  ndr_death_session_type      session,
  /* -> session to use in od_start_txn */
  ndr_od_clash_info*      info,
  /* -> Information on clash */
  ndr_death_doid_class*      old_doid,
  /* -> DOID of file with clashing contents */
  ndr_death_doid_class*      new_doid);
  /* <- Doid of duplicated file */
//Call back to the host in respect of clashes detected at the
//filesystem level
// (via pre trigger functions)
typedef ndr_int32 EXPORT (*ndr_fd_filesys_clash_fn)(
  ndr_od_db_handle      db,     /* -> Database */
  ndr_death_session_type      session,
  /* -> session to use in od_start_txn */
  ndr_od_clash_info*      info,
  /* -> Information on clash */
  ndr_death_doid_class*      doid);
  /* -> DOID of file with clashing contents */
```

A parameter block such as the following is passed to clash handling functions to provide them with information about the clash:

```
typedef struct
{
  ndr_death_pid_type*      pid;
  /* -> PID of clashing txn */
  ndr_od_clash_type      clash_type;
  /* -> Clash type */
  ndr_os_class      class_id;
  /* -> Class id of object causing the clash */
  ndr_os_attribute      attr_id;
  /* -> Attr id of object causing the clash */
  ndr_death_update_list*      update_list;
  /* -> Update list of transaction */
  ndr_death_update*      update;
  /* -> Update causing clash (always a pointer
     into *update_list */
  BOOLEAN      is_higher_priority;
  /* -> Relative priority of location
     to which update is being applied.
     TRUE=> Applying to location with higher
     priority (e.g. to location set with
     central location) */
  void*      agent_merge_info;
  /* -> Value which is reserved for (arbitrary)
     use by agent clash handlers. It is
     guaranteed to be set to NULL on the
     first clash of a merge, and preserved
     for all subsequent clashes within that
     merge */
} ndr_od_clash_info;
```

A close handler function of type ndr_fd_close_fn can be registered with the file distributor **90** for a database type to

be called whenever the file distributor **90** closes a modified local copy of the file contents, passing the new length and modification date/time and user identifier:

```
typedef ndr_rst EXPORT (*ndr_fd_close_fn)(
    ndr_od_db_handle      db,      /* -> Database */
    ndr_dodh_session_type      session,
    /* -> session to use in ot_start_txn */
    ndr_os_class      class_id,
    /* -> Class ID of file */
    ndr_dodh_uoid_type*      uoid,      /* -> UOID */
    UINT32      length,
    /* -> length of closed file */
    UINT16      time,
    /* -> modification time */
    UINT16      date,
    /* -> modification date */
    UINT32      updatedBy);
    /* -> modification user */
```

A creation handler function of type ndr_fd_creation_fn can be registered with the file distributor **90** for a database type to be called whenever the file distributor **90** creates a local copy of the file contents. This allows the replica manager **46** on a central server computer **28** to update the master copy of the file to reflect the attributes of the file created while disconnected:

```
typedef ndr_rst EXPORT (*ndr_fd_creation_fn)(
    ndr_od_txn_id      txn_id,      /* -> txn_id */
    ndr_os_class      class_id,
    /* -> Class ID of file */
    ndr_dodh_uoid_type*      uoid);      /* -> UOID of file */
```

The file distributor **90** embodiment also provides the following:

```
//Return aggregated information about all volumes
ndr_rst EXPORT
NdrFdVolumeInfo(
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    UINT32*      cluster_size,
    /* <- Number of bytes per cluster */
    UINT16*      total_clusters,
    /* <- Total number of clusters */
    UINT16*      free_clusters);
    /* <- Number of free clusters */
//Add a file
ndr_rst EXPORT
NdrFdAddFile(
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    ndr_dodh_uoid_
type* uoid,
    /* -> Uoid of file cre-
ated */
    UINT32      length);
    /* -> Length of existing file (0 when new) */
//Remove a file
ndr_rst EXPORT
NdrFdRemoveFile(
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    ndr_dodh_uoid_type* uoid);
    /* -> Uoid of file removed */
    //Open a file for reading or writing by a task
    ndr_rst EXPORT
    NdrFdOpenFile(
    ndr_od_txn_id      txn_id, /* -> txn_id */
    ndr_os_class      class_id,
    /* -> Class ID of file to open */
    ndr_dodh_uoid_type uoid,
    /* -> Uoid of file to open */
    ndr_fd_open_mode  open_mode,
    /* -> Open for read and/or write? */
    ndr_fd_fork_id*      fork_id,
```


-continued

```
    /* <- FD Fork Id of open file */
    BOOLEAN      is_create,
    /* -> TRUE if open as part of create */
    ndr_fd_io_function*  read_function,
    /* <- Function to be used for READ operations */
    ndr_fd_io_function*  write_function,
    /* <- Function to be used for WRITE operations */
    ndr_fd_fork_id*      io_fork_id,
    /* <- FD Fork Id used
with above two functions
(only) */
    UINT16*      num_forks_remaining);
    /* <- Number of forks remaining to be opened
    on same machine */
//Read from a file
ndr_rst EXPORT
NdrFdReadFile(
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    ndr_fd_fork_id      fork_id,  /* -> Id of open fork */
    UINT32      offset,
    /* -> Offset at which to start reading */
    UINT16      req_length,
    /* -> Number of bytes requested to read */
    UINT8*      data,      /* <- Data read */
    UINT16*      act_length);
    /* <- Actual number of bytes read */
//Write to a file
ndr_rst EXPORT
NdrFdWriteFile(
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    ndr_fd_fork_id      fork_id,  /* -> Id of open fork */
    UINT32      offset,
    /* -> Offset at which to start writing */
    UINT16      req_length,
    /* -> Number of bytes requested to write */
    UINT8*      data);      /* -> Data to be written */
//Get the current length of an open file
ndr_rst EXPORT
NdrFdGetOpenFileLength(
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    ndr_fd_fork_id      fork_id,  /* -> Id of open fork */
    UINT32*      length);
    /* <- Length of that open file */
//Lock or Unlock a range of bytes in an open file
ndr_rst EXPORT
NdrFdClearPhysicalRecord ( or NdrFdLockPhysicalRecord(
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    ndr_fd_fork_id      fork_id,  /* -> Id of open fork */
    UINT32      offset,  /* -> Offset for lock */
    UINT32      req_length);
    /* -> Number of bytes requested to lock */
//Ensure a file's contents are on disk
ndr_rst EXPORT
NdrFdCommitFile(
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    ndr_fd_fork_id      fork_id);  /* -> Id of open fork
*/
//Close a file, having completed reading and writing
ndr_rst EXPORT
NdrFdCloseFile
    ndr_od_txn_id      txn_id,  /* -> txn_id */
    ndr_fd_fork_id      fork_id);  /* -> Id of open fork
*/
//Given a UOID of a file or directory ensure its name
//in the specified namespace, along with its parent's UOID
ndr_rst EXPORT
NdrFdGetFilename(
    ndr_od_db_handle      db,
    /* -> handle to current database */
    ndr_dodh_uoid type*  file_or_dir_id,
    /* -> Uoid of object whose name is wanted */
    ndr_os_attr_property  namespace,
    /* -> Namespace (e.g. DOS) of name wanted */
    void*      name_buffer,
    /* <- Buffer to receive name */
    UINT16*      name_size,
    /* -> Size of provided buffer */
    ndr_dodh_uoid_type*  parent_dir_id);
    /* <- Parent UOID of object (NULL at root) */
//Callback functions to be used with
```

-continued

```
//NdrFdRegisterChangedIdCallback
typedef nds_ret EXPORT
(*NdrFdChangedIdCallback) (
    nds_rd_db_handle    db,         /* -> Database Id */
    nds_os_class        class_id,
    /* -> Class ID of file or dir */
    nds_dodb_oid_type   *oid,       /* -> Oid of file or dir
*/
    UINT32              new_id);
    /* -> New Id allocated by underlying file system */
```

A NdrFdRegisterChangedIdCallback( ) function provides registration of a callback function to be called when a change to a file or directory's unique identifier is made. On a NetWare 4.x server this normally happens only when the file or directory is created by an internal file distributor **90** trigger function. However the identifier will be needed by agents for tasks such as directory enumeration. Because trigger functions cannot directly modify replicated objects, a record of the identifier change is queued within the file distributor **80** and the callback is made asynchronously.

```
nds_ret EXPORT
NdrFdRegisterChangedIdCallback
    nds_os_db_type_handle       db_type, /* -> Database type */
    NdrFdChangedIdCallback      fn); /* -> Callback function */
```

The interface also provides the following:

```
    //Register clash handlers for contents clashes for files held
    in
    //a database of the given type.
    nds_ret EXPORT
    NdrFdRegisterClashHandlers(
        nds_os_db_type_handle   db_type, // -> Database type
        nds_os_class            class_id,
        // -> Class ID of contents 'container' eg file
        nds_fd_object_clash_fn  object_clash_fn,
        // -> Clash handler for dealing with conflicts
        // -> between objects (e.g. contents modification
        //      and removal)
        nds_fd_filesys_clash_fn filesys_clash_fn,
        // -> Clash handler for conflicts that arise
        //      through some characteristic of the file
        //      system (e.g. access rights on delete)
        nds_fd_filesys_clash_fn filesys_clash_fn);
    //Register a trigger-like routine to be called when a local
    //replica of a file is modified. The routine takes the length
    //and modification date/time of the local replica of the file.
    nds_ret EXPORT
    NdrFdRegisterCloseHandler(
        nds_os_db_type_handle   db_type, // -> Database type
        nds_os_class            class_id,
        /* -> Class ID of file */
        nds_fd_close_fn         close_fn);
        /* -> Clash handler to call */
    //Register a trigger-like routine to be called when a local
    //replica of a file is has been created. This allows the
    //replica manager on a central server to update the
    //server's master copy of the file to reflect the attributes
    //of the file created during the disconnection.
    nds_ret EXPORT
    NdrFdRegisterCreationHandler(
        nds_os_db_type_handle   db_type, /* -> Database type */
        nds_os_class            class_id,
        /* -> Class ID of file */
        nds_fd_creation_fn      creation_fn);
        /* -> Creation handler to call */
    //De-register a clash or close or creation handler for
    //contents clashes for files held in a database of the given
    type
    nds_ret EXPORT
    NdrFdDeRegisterClashHandler( or CloseHandler( or
```

-continued

```
    CreationHandler(
        nds_os_db_type_handle   db_type, // -> Database type
        nds_os_class            class_id); // -> Class ID of fi>
    //Synchronize all the files to and from this client for the
    //passed database. Return control when the files are up to
    date.
    nds_ret EXPORT
    NdrFdSynchronizeFiles(nds_rd_db_handle db);
    //Called from pre trigger functions to check whether
    //or not the current connection has sufficient
    //per-user rights to perform a particular operation
    //on a particular file system object.
    nds_ret
    NdrFdCheckRights(
        nds_dodb_oid_type*      file_oid,
        // oid of object requiring rights to operation
        nds_rd_db_handle        db,
        // database raising the pre trigger
        UINT16                  operation);
        // bits representing operation
    //Note that a file has been locally modified, setting
    //modification info and triggering propagation onto other
    //replicas.
    nds_ret EXPORT
    NdrFdNoteFileModified
        nds_rd_txn_id           txn_id, /* -> txn_id */
        nds_dodb_dood_class*    file_dood);
```

The trigger function registrations **94** identify trigger functions that are provided by agents and registered with the object distributor **82**. A registered trigger function is called on each event when the associated event occurs. Suitable events include object modification events such as the addition, removal, movement, or modification of an object. Because the trigger functions are called on each location, they can be used to handle mechanisms such as file replication, where the file contents are not stored within the target database, while ensuring that the existence, content, and location of the file tracks the modifications to the target database. All objects must have been locked, either implicitly or via NdrOdLock( ), in the triggering transaction before the corresponding trigger function is called, and other objects may only be modified if the trigger function is being called for the first time at the location in question.

In an alternative embodiment, the replica manager **46** comprises a NetWare Loadable Module ("NLM") and an NWAdmin snap-in module. The NLM uses hooks in the NetWare file system **48** to intercept updates to the local NetWare storage **54**, and uses standard NetWare file system Application Programmer's Interface ("API") calls to update the storage **54** when synchronizing. The architecture is symmetric, with the same code running on all computers **28**.

The NLM has three major internal subsystems. An environment subsystem provides portability by separating the other two internal subsystems from the operating system environment such as the Windows NT or UNIX environment. The environment subsystem provides execution, debugging, scheduling, thread, and memory management services. A Distributed NetWare ("DNW") subsystem implements NetWare semantics by intercepting NetWare file system calls and calls from a DNW API and making corresponding requests of a dispatch layer discussed below. A distributed responder subsystem implements the replica manager **46** to provide a distributed disconnectable object database which supports replication, transaction synchronization, and schema-definable objects, including file objects, as described herein.

An application layer contains application programs and the NWAdmin snap-in. These programs interface with the replica manager **46** either by calling an API or by attempting to access the storage device **54** and being intercepted. An

intercept layer in the replica manager **46** intercepts and routes external requests for file system updates that target a replica **56**. A dispatch later receives the routed requests and dispatches them to an appropriate agent **44**.

The agents **44**, which have very little knowledge of the distributed nature of the database, invoke the consistency distributor **74**, location distributor **78**, object distributor **82**, and/or file distributor **90**. For example, a directory create would result in an object distributor **82** call to NdrOdAdd( ) to add a new object of type directory.

In contrast to the agents **44**, the distributors **74**,**78**,**82**, and **90** have little semantic knowledge of the data but know how it is distributed. The object distributor **82** uses the location distributor **78** to control multi-location operations such as replication and synchronization. The consistency distributor **74** manages transaction semantics, such as when it buffers updates made after a call to NdrOdStartTxn( ) and applies them atomically when NdrOdEndTxn( ) is called. The file distributor **90** manages the replication of file contents.

The processors **76**, **86**, **88**, and **92** process requests for the local location **40**. The consistency processor **76** handles transaction semantics and synchronization, and uses the transaction logger **88** to log updates to the database. The logged updates are used to synchronize other locations **40** and to provide recovery in the event of a clash or a crash. The logger **88** maintains a compressed transaction log. The log is "compressed," for example, in that multiple updates to the "last time modified" attribute of a file object will be represented by a single update. The logger **88** maintains a short sequential on-disk log of recent transactions; the longer-term log is held in the object database as update log entry objects.

The object processor **86** implements a local object store and supports the following access methods: hierarchical (e.g., add file object under directory object); global indexed (e.g., read any object using its UOID); and local indexed (e.g., read files and directories within a directory in name order). The object processor **86** uses a variant of a B*-tree. The object processor **86** uses a page table to support atomic commitment of transactional updates, providing rollback and protection against crashes of the computer **40**.

A file system layer in the file system interface **48** provides a flat file system interface built on the local first file system. It re-maps the flat file system calls to the corresponding files in the hierarchical NetWare volumes to support the current NetWare file system.

With reference to FIGS. **1** through **4** and particular focus on FIG. **4**, a method of the present invention for synchronizing transactions in the network **10** of connectable computers **28** is illustrated. The transactions target entries in a distributed hierarchical database that contains convergently consistent replicas **56** residing on separate computers **28** in the network **10**. The method comprises the following computer-implemented steps.

A connecting step **100** uses the replica manager **46** and network link manager **50** to establish a network connection between a first computer **36** and a second computer **38**. For purposes of illustrating the method, the first computer **36** shown in FIG. **1** is a client computer **20** and the second computer **38** is a server computer **16**. However, a server and a client, or two servers, or two clients, may also be synchronized and otherwise managed according to the present invention.

A first identifying step **102** identifies a first transaction that targets an entry in a first replica on the first computer **36**. This may be accomplished using the replica manager **46** described above, and may include intervening in a chain of calls that begins with a file system update request and ends in an operating system request on the first computer **36**. A single file system update operation, such as creating a file, is converted into a transaction that involves a set of update operations to the target object database. The first identifying step **102** may also include access to the update log maintained by the transaction logger **88**.

The replica may be a replica **56** of a distributed hierarchical database that includes objects and object attributes defined according to the schema **84**, which is accessible outside the database. Thus, the database entries may include, without limitation, file and directory entries for a file system and/or Novell directory services entries.

A locating step **104** locates a second replica **56** that resides on the second computer **38** and that corresponds in entries (but not necessarily in entry values) to the first replica **56**. This may be accomplished using the replica manager **46** described above, and may include access to the location state processor **80**.

A first transferring step **106** transfers an update based on the first transaction over the network connection from the first computer **36** to the second computer **38**. The update may include operation and object-identifying information from a log entry, a transaction sequence number corresponding to the first transaction, and a location identifier corresponding to the first computer **36**. The first transaction may be one of a plurality of transactions completed at the first computer **36**, in which case each completed transaction has a corresponding transaction sequence number.

The transfer may be accomplished in accordance with a SyncUpdate request and the respective consistency distributors **74** of the computers **36**, **38**. If the contents of a distributed file have been modified, the file distributors **90** on the two computers **36**, **38** are also utilized.

A first applying step **108** performed on the second computer **38** atomically applies the first transaction update to the second replica **56**. This may be accomplished using the consistency processor **76**, object processor **86**, transaction processor **88**, file system interface **48**, and storage device and controller **54** of the second computer **38**. The applying step **108** may set a database object lock that serializes updates to the first replica **56**, or users may rely on a combination of retries and clash handling as noted above. If file contents are involved, the file processor **92** on the second computer **38** is also utilized.

One method of the present invention includes additional computer-implemented steps. A second identifying step **110** identifies a second transaction that targets an entry in the second replica **56** on the second computer **38**. This may be accomplished in a manner similar to the first identifying step **102**. A second transferring step **112** transfers an update based on the second transaction over the network connection from the second computer **38** to the first computer **36**. This may be accomplished in a manner similar to the first transferring step **106**. A second applying step **114** performed on the first computer **36** applies the second transaction update to the first replica **56**. This may be accomplished in a manner similar to the first applying step **108**.

The identifying step **102** and/or **110** may be preceded by the computer-implemented step of recording the transaction on a non-volatile storage medium using the storage device and controller **54**. The recording step may enter a representation of the transaction in a transaction log maintained by the transaction logger **88**.

A detecting step **118** may detect a missed update by detecting a gap in a plurality of transferred transaction sequence numbers. To facilitate gap detection, in one

embodiment the transaction sequence numbers are consecutive and monotonic for all completed transactions.

Although the invention is illustrated with respect to synchronization of two computers **36**, **38**, those of skill in the art will appreciate that a more than two computers may also be synchronized. In such cases, additional transferring steps transfer the transaction updates to other computers **28** in the network **10**, and additional applying steps apply the transaction updates to replicas **56** on the other computers **28**.

In summary the present invention provides a system and method for properly synchronizing transactions when a disconnectable computer **28** is reconnected to the network **10**. The invention is not limited to file system operations but can instead be extended to support a variety of database objects by using the schema **84**, object distributor **82**, object processor **86**, and other modules. Clash handling means may be used to identify potentially conflicting database changes and allow their resolution by either automatic or manual means. Clash handling and retries also make locks optional.

The synchronization time period does not depend directly on the total number of files, directories, or other objects in the replica **56**. Rather, the required time depends on the number and size of the objects that require updating. This facilitates synchronization over slow links, such as mobile computer modems and WAN server connections. Unlike conventional systems such as state-based systems, the present invention is therefore readily scaled upward to handle larger networks. Moreover, the dynamic intervention and processing of operations by the replica managers **46** allows systems of the present invention to support continuous synchronization across slow links. The replica managers **46** also allow the use of consistent file locations regardless of whether a mobile computer is connected to the network **10**.

Although particular methods embodying the present invention are expressly illustrated and described herein, it will be appreciated that apparatus and article embodiments may be formed according to methods of the present invention. Unless otherwise expressly indicated, the description herein of methods of the present invention therefore extends to corresponding apparatus and articles, and the description of apparatus and articles of the present invention extends likewise to corresponding methods.

The invention may be embodied in other specific forms without departing from its essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. Any explanations provided herein of the scientific principles employed in the present invention are illustrative only. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

What is claimed and desired to be secured by patent is:

1. A method for synchronizing transactions in a system, the system including at least two computers capable of being connected by a network link, each of the computers including a storage device containing a distributed hierarchical database replica, a device controller in signal communication with the storage device, a replica manager in signal communication with the device controller and the network link, and a database manager in signal communication with the replica manager, the method comprising the computer-implemented steps of:

routing database transactions with the database managers through the replica managers to the device controllers;

connecting at least two computers with the network link; and

using each of at least two replica managers to route a transaction to another replica manager after the two computers are connected by the network link.

2. The method of claim **1**, further comprising the computer-implemented step of recording a transaction on a non-volatile storage medium.

3. The method of claim **2**, wherein said recording step comprises the step of entering a representation of the transaction in a transaction log.

4. The method of claim **1** for synchronizing transactions in the system, the system containing a network of connectable computers, the transactions targeting entries in the distributed hierarchical database replicas, the replicas residing on separate computers in the network, said method comprising the steps of:

obtaining a network connection between a first computer and a second computer as part of said step of connecting at least two computers with the network link;

identifying a first transaction that targets a distributed hierarchical database entry in a first replica on the first computer, as part of said step of routing database transactions; and

as part of said step of using each of at least two replica managers, locating a corresponding second replica that resides on the second computer, transferring an update based on the first transaction over the network connection from the first computer to the second computer, and applying the first transaction update to the second replica.

5. The method of claim **4**, wherein the replicas contain file descriptors and directory descriptors for a file system.

6. The method of claim **4**, wherein the replicas contain directory services entries.

7. The method of claim **4**, wherein said step of transferring an update comprises transferring a transaction sequence number corresponding to the first transaction and a location identifier corresponding to the first computer.

8. The method of claim **7**, wherein the first transaction is one of a plurality of transactions completed at the first computer and each completed transaction has a corresponding transaction sequence number.

9. The method of claim **8**, wherein the transaction sequence numbers are generated in a predetermined order.

10. The method of claim **9**, wherein the transaction sequence numbers are consecutive and monotonic for all completed transactions.

11. The method of claim **8**, further comprising the computer-implemented step of detecting a missed update by detecting a gap in a plurality of transferred transaction sequence numbers.

12. The method of claim **7**, wherein the first transaction is one of a plurality of transactions completed at the first computer, each completed transaction has a corresponding transaction sequence number, and the transaction sequence numbers are consecutive and monotonic for all completed transactions.

13. The method of claim **12**, further comprising the computer-implemented step of detecting a missed update by detecting a gap in a plurality of transferred transaction sequence numbers.

14. The method of claim **4**, wherein said transferring step further comprises transferring the first transaction update to at least one computer other than the first and second computers, and said applying step further comprises applying the first transaction update to at least one replica other than the first and second replicas.

15. The method of claim **4**, wherein said applying step comprises setting a database object lock that serializes updates to the first replica.

16. The method of claim 4, wherein said applying step comprises applying the first transaction to the second replica atomically.

17. A system comprising at least two computers capable of being connected by a network link, each of said computers comprising:

a storage device containing a replica, said replica containing entries of a distributed hierarchical database;

a device controller in signal communication with said storage device;

a replica manager in signal communication with said device controller and said network link; and

a database manager in signal communication with said replica manager, said database manager on each computer configured to route database transactions to said device controller only through said replica manager, and said replica managers configured to route such transactions to each other after said computers are connected by said network link.

18. The system of claim 17, wherein each of said replica managers comprises a replica distributor and a replica processor.

19. The system of claim 18, wherein said replica distributor comprises a consistency distributor and a location distributor.

20. The system of claim 18, wherein said replica distributor comprises a consistency distributor, a location distributor, an object distributor, and an object schema.

21. The system of claim 20, wherein said replica distributor further comprises a file distributor.

22. The system of claim 18, wherein said replica processor comprises a consistency processor and a location state processor.

23. The system of claim 18, wherein said replica processor comprises a consistency processor, a location state processor, an object processor, and a transaction logger.

24. The system of claim 23, wherein said replica processor further comprises a file processor.

25. The system of claim 17, wherein said replica manager comprises trigger function registrations, each registration associating a registered trigger function with a database operation such that the registered trigger function will be invoked on each replica, once the computers are connected, if the associated operation is requested of the database manager.

26. The system of claim 25, wherein the associated operation belongs to the group consisting of add, remove, modify, and move operations.

27. The system of claim 17, wherein said replicas contain file descriptors and directory descriptors for a file system.

28. The system of claim 17, wherein said replicas contain directory services entries.

29. A computer-readable storage medium having a configuration that represents data and instructions which cause a first computer and a second computer in a system to perform method steps for synchronizing transactions, the system including at least two computers capable of being connected by a network link, each of the computers including a storage device containing a distributed hierarchical database replica, a device controller in signal communication with the storage device, a replica manager in signal communication with the device controller and the network link, and a database manager in signal communication with the replica manager, the method comprising the steps of:

routing database transactions with the database managers through the replica managers to the device controllers;

connecting at least two computers with the network link; and

using each of at least two replica managers to route a transaction to another replica manager after the two computers are connected by the network link by:

identifying a first transaction that targets an entry in the first replica, which resides on the first computer;

transferring an update based on the first transaction over the network connection from the first computer to the second computer;

applying the first transaction update to the second replica, which resides on the second computer;

identifying a second transaction that targets an entry in the second replica;

transferring an update based on the second transaction over the network connection from the second computer to the first computer; and

applying the second transaction update to the first replica.

30. The storage medium of claim 29, wherein the method further comprises the step of entering a representation of a transaction in a transaction log.

31. The storage medium of claim 29, wherein the distributed hierarchical database includes objects and object attributes defined according to a schema that is accessible outside the database.

32. The storage medium of claim 29, wherein the replicas contain file descriptors and directory descriptors for a file system.

33. The storage medium of claim 29, wherein the replicas contain directory services entries.

34. The storage medium of claim 29, wherein the step of transferring an update comprises transferring a transaction sequence number corresponding to the first transaction and a location identifier corresponding to the first computer.

* * * * *

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	
Filing Date	2013-04-03
First Named Inventor	7,822,816
Art Unit	
Examiner Name	
Attorney Docket Number	20351.RX816

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	5704029		1997-12-30	Gerald V. Wright, Jr.	
	2	6477373	B1	2002-11-05	Rappaport et al.	
	3	6584464	B1	2003-06-24	David Warthen	
	4	5991771		1999-11-23	Falls et al.	
	5	5442786		1995-08-15	Robert E. Bowen	

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20020007303	A1	2002-01-17	Brookler et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2013-04-03
	First Named Inventor	7,822,816
	Art Unit	
	Examiner Name	
	Attorney Docket Number	20351.RX816

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] ¡	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1	0779759	EP	A2	1997-06-18	Alain Rossmann		☐
	2	99/33390	WO		1999-07-08	Benigno		☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2013-04-03
	First Named Inventor	7,822,816
	Art Unit	
	Examiner Name	
	Attorney Docket Number	20351.RX816

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☒ A certification statement is not submitted herewith.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Jay J. Guiliano/	Date (YYYY-MM-DD)	2013-04-03
Name/Print	Jay J. Guiliano	Registration Number	41,810

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted today via the Office electronic filing system (EFS-Web) in accordance with 37 CFR §1.6 (a)(4).

Date: April 3, 2013 Signature: /Stephanie Dominguez/
Printed Name: Stephanie Dominguez

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Reexamination of: Inventors: Payne, J. David Patent No.: 7,822,816 Filed: August 19, 2003 For: SYSTEM AND METHOD FOR DATA MANAGEMENT	REQUEST FOR REEXAMINATION UNDER 35 U.S.C. §§ 302-307 AND 37 C.F.R. § 1.510

CERTIFICATE OF SERVICE

The undersigned hereby certifies that a copy of this **REQUEST FOR *EX PARTE* REEXAMINATION OF U.S. PATENT 7,822,816** along with all exhibits and supporting documentation has been served via USPS Priority Mail on April 3, 2013, upon the following:

Fellers Snider Blankenship
Bailey & Tippens
The Kennedy Building
321 South Boston, Suite 800
Tulsa, OK 74103-3318

/Stephanie Dominguez/
Stephanie Dominguez

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted today via the Office electronic filing system (EFS-Web) in accordance with 37 CFR §1.6 (a)(4).

Date: April 3, 2013 Signature: /Stephanie Dominguez/
Printed Name: Stephanie Dominguez

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Reexamination of: Inventors: Payne, J. David Patent No.: 7,822,816 Filed: August 19, 2003 For: SYSTEM AND METHOD FOR DATA MANAGEMENT	REQUEST FOR REEXAMINATION UNDER 35 U.S.C. §§ 302-307 AND 37 C.F.R. § 1.510

Mail Stop *Ex Parte* Reexamination
ATTN: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR *EX PARTE* REEXAMINATION OF U.S. PATENT 7,822,816

TABLE OF CONTENTS

TABLE OF EXHIBITS

LIST OF EXHIBITS

The exhibits to the present Request are arranged in two groups: prior art ("PA"), relevant patents ("PAT"), and Other Exhibits (OTH).

A. PRIOR ART (PA)

PA-SB08A/B	USPTO Form SB/08A/B
PA-A	U.S. Patent No. 5,704,029 to Wright ("Wright")
PA-B	U.S. Patent No. 6,477,373 to Rappaport et al. ("Rappaport")
PA-C	U.S. Patent No. 6,584,464 to Warthen ("Warthen")
PA-D	U.S. Patent App. No. 2002/0007303 to Brookler et al. ("Brookler")
PA-E	European Patent Application EP 0779,759 to Rossmann ("Rossmann")
PA-F	PCT Published Application WO 99/33390 to Benigno ("Benigno")
PA-G	U.S. Patent No. 5,991,771 to Falls et al. ("Falls")
PA-H	U.S. Patent No. 5,442,786 to Bowen ("Bowen")

B. RELEVANT PATENT MATERIALS (PAT)

PAT-A	U.S. Patent No. 7,822,816 ("the '816 patent")

C. OTHER EXHIBITS (OTH)

OTH-A	PowerPoint showing unpatentability of claim 1 of the '816 patent.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Reexamination of: Inventors: Payne, J. David Patent No.: 7,822,816 Filed: August 19, 2003 For: SYSTEM AND METHOD FOR DATA MANAGEMENT	REQUEST FOR REEXAMINATION UNDER 35 U.S.C. §§ 302-307 AND 37 C.F.R. § 1.510

Mail Stop *Ex Parte* Reexamination
ATTN: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR EX PARTE REEXAMINATION OF U.S. PATENT 7,822,816

Dear Sir:

Pursuant to 35 U.S.C. §§ 302-307, 37 C.F.R. § 1.510, and MPEP § 2295, GEICO Corporation, GEICO Casualty Company, GEICO General Insurance Company, GEICO Indemnity Company and Government Employees Insurance Company (collectively, "GEICO") (herein after "Requester") hereby respectfully request reexamination of Claims 1-14 U.S. Patent No. 7,822,816 ("the '816 patent") filed and issued to. *See* Exhibit PAT-A.

This Request is based on the cited prior art documents set forth herein and on the accompanying Form PTO-SB/08A/B. *See* Exhibit PA-SB/08A/B. All of the cited prior art patents and publications constitute effective prior art as to the claims of the 1-14 patent under 35 U.S.C. § 102 and 35 U.S.C. § 103.

Pursuant to 37 C.F.R. § 1.510 Requester hereby respectfully requests reexamination pursuant to 35 U.S.C. §§ 301 *et seq.* and 37 C.F.R. § 1.510 *et. seq.*, of Original Claims 1-14 of the '816 patent. Reexamination is requested in view of the substantial new questions of patentability ("SNQs") set forth in detail below and in the accompanying claim charts. Requester

reserves all rights and defenses available including, without limitation, defenses as to invalidity and unenforceability. By simply filing this Request in compliance with applicable statutes, rules, and regulations, Requester does not represent, agree or concur that the '816 patent is enforceable. By asserting the SNQs herein, Requester specifically asserts that Original Claims 1-14 of the '816 patent are in fact not patentable.

Accordingly, the U.S. Patent and Trademark Office ("the Office") should reexamine and find claims 1-14 of the '816 patent unpatentable and cancel these claims, rendering them null, void, and otherwise unenforceable.

I. REQUIREMENTS FOR *EX PARTE* REEXAMINATION UNDER 37 C.F.R. § 1.510

Requester satisfies each requirement for *Ex Parte* reexamination of the '816 patent pursuant to 37 C.F.R. § 1.510. A full copy of the '816 patent is submitted herein as Exhibit PAT-A in accordance with 37 C.F.R. § 1.510(b)(4)

Pursuant to 37 C.F.R. §1.510(b)(3), a copy of every patent or printed publication relied upon to present an SNQ is submitted herein at Exhibits PA-A through PA-H, citation of which may be found on the accompanying Form PTO-SB/08A as Exhibit PTO-SB/08A in accordance with 37 C.F.R. § 1.510. Each of the cited prior art publications constitute effective prior art as to the claims 1-14 of the '816 patent under 35 U.S.C. § 102 and 35 U.S.C. § 103. Furthermore, each piece of prior art submitted was either not considered by the Office during the prosecution of the '816 patent or is being presented in a new light under MPEP § 2242 as set forth in the detailed explanation below and in the attached claim charts.

A statement pointing out each SNQ based on the cited patents and printed publications, and a detailed explanation of the pertinency and manner of applying the patents and printed publications to claims 1-14 of the '816 patent, is presented below and in attached claim charts in accordance with 37 C.F.R. § 1.510(b)(2).

A copy of this request has been served in its entirety on the patent owner in accordance with 37 C.F.R. § 1.510(b)(5) at the following address:

In accordance with 37 C.F.R. § 1.510(a), a credit card authorization to cover the Fee for reexamination of $12,000.00 (large entity) is attached. If this authorization is missing or defective, please charge the Fee to the Novak Druce and Quigg Deposit Account No. 14-1437.

II. OVERVIEW

A. DESCRIPTION OF THE '816 PATENT

The '816 patent generally discusses a system or method that automates the manual activities associated with completion of forms, for example hospital intake forms. This automation takes place via computerized questionnaires that have the questions filed out locally and then transmitted and stored in a central database. '816 patent at 4:55-60. The system contemplates having an operating system that allows multiple devices to access the data stored in the database. '816 at 5:6-11.

A specific example provided by the '816 patent is for a "handheld computer" to accept the inputs of an internet based questionnaire. '816 patent at 8:5-37. A user builds the questionnaire by putting questions into a form and then the server will send the completed questionnaire to the appropriate handheld devices. '816 at 8:38-67. Based on this system, the '816 patent states that this allows a system where questions in a questionnaire are updated at the server and sent to each of the handhelds, thereby keeping a consistent questionnaire throughout the system. '816 patent at 9:1-14.

B. THE '816 PATENT APPLICATION PROSECUTION HISTORY

The Examiner during the original prosecution rejected claims 1 and 5 as anticipated by Lew (U.S. Patent Application No. 2004/0210472), which the Examiner found to disclose each and every limitation of the claims. *See* OA issued August 10, 2006 at 2-3. The Examiner found that Sendowski (U.S. Patent Application Publication No. 2003/0198934) anticipated claim 7, and that claims 2-4, 6, and 9-11 were rendered obvious by Lew in view of Sendowski. *See* OA issued August 10, 2006 at 3-9. Finally, the Examiner found that claim 8 was obvious over Sendowski in view of Joao (U.S. Patent Application No. 2001/0056374). *See* OA issued August 10, 2006 at 9.

In response on February 12, 2007, Patent Owner amended the claims in an attempt to overcome the prior art, adding limitations that are not found in the claims at issue in this proceeding. *See* Reply filed February 12, 2007. After re-filing the Reply due to a non-compliant amendment, the Examiner again rejected the claims on May 22, 2007, maintaining the anticipation rejection of claim 7 over Sendowski, while rejecting the remaining claims as obvious over Lew (U.S. Patent Application Publication No. 2004/0210472) and Porter (U.S. Patent No. 6,163,811) or Lew in view of Porter and Sendowski. *See* OA issued May 22, 2007. Patent Owner's only argument that was applicable after the new grounds of rejections was that

Sendowski doesn't disclose a "loosely networked computer." The Examiner disagreed and found that the disclosure of connecting to the internet was fits within the broadest reasonable definition of that phrase based on the Patent Owner's disclosed embodiments. *See* OA issued May 22, 2007 at 12.

To overcome the rejections of the May 22, 2007 Office Action, Patent Owner filed declarations under 37 C.F.R. 1.131 to antedate the references. *See* Reply filed September 24, 2007. However, the Examiner found that the Declaration was not sufficient to establish prior conception, and therefore, the Patent Owner failed to overcome the rejections. *See* OA issued October 30, 2007 at 13. In response, the Patent Owner reargued that the Declaration was sufficient and asked for reconsideration in light of a document that allegedly showed proof of conception. *See* Reply filed April 30, 2008. Based on this further evidence, the prior art of record was rendered moot and the Examiner withdrew the rejections of record, but provided new grounds of rejection over Peters (U.S. Patent No. 5,842,195) and Joao (U.S. Patent Application No. 2001/0056374) or Porter (U.S. Patent No. 6,163,811) and Brookler (U.S. Patent Application No. 2002/0007303.) *See* Office Action issued September 4, 2008.

There were some intermediate Office Actions and responses, but on May 3, 2010 Patent Owner filed a final claim set that was deemed allowable by the Examiner. Specifically, after cancelling four claims via an Examiner's Amendment, the claims were allowed because:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3.

C. Related Co-Pending Litigation Requires Treatment with Special Dispatch

The '816 patent is presently the subject of *Macrosolve, Inc. v. GEICO Insurance Agency Case No. 6:12-cv-74* pending in the U.S. District Court for the Eastern District of Texas.

Accordingly, pursuant to 35 U.S.C. § 314, Requester respectfully urges that this Request be granted and reexamination conducted with "special dispatch" in accordance with M.P.E.P. § 2261.

D. CLAIM CONSTRUCTION

For purposes of this Request, the claim terms are presented by the Requester in accordance with 37 C.F.R. § 1.555(b) and MPEP § 2111. Specifically, each term of the claims is to be given its "broadest reasonable construction" consistent with the specification. MPEP § 2111; *In re Swanson*, No. 07-1534 (Fed. Cir. 2008); *In re Trans Texas Holding Corp.*, 498 F.3d 1290, 1298 (Fed. Cir. 2007) (citing *In re Yamamoto*, 740 F.2d 1569, 1571 (Fed. Cir. 1984)).

Although the District Court has yet to rule on the scope of these claim limitations, the Federal Circuit noted in *Trans Texas* that the Office has traditionally applied a broader standard than a Court does when interpreting claim scope. MPEP § 2111. The Office applies to the verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art. *In re Morris*, 127 F.3d 1048, 1054-55, 44 U.S.P.Q.2d 1023, 1027-28 (Fed. Cir. 1997). The rationale underlying the "broadest reasonable construction" standard is that it reduces the possibility that a claim, after issue or certificate of reexamination, will be interpreted more broadly than is justified. 37 C.F.R. § 1.555(b), MPEP § 2111.

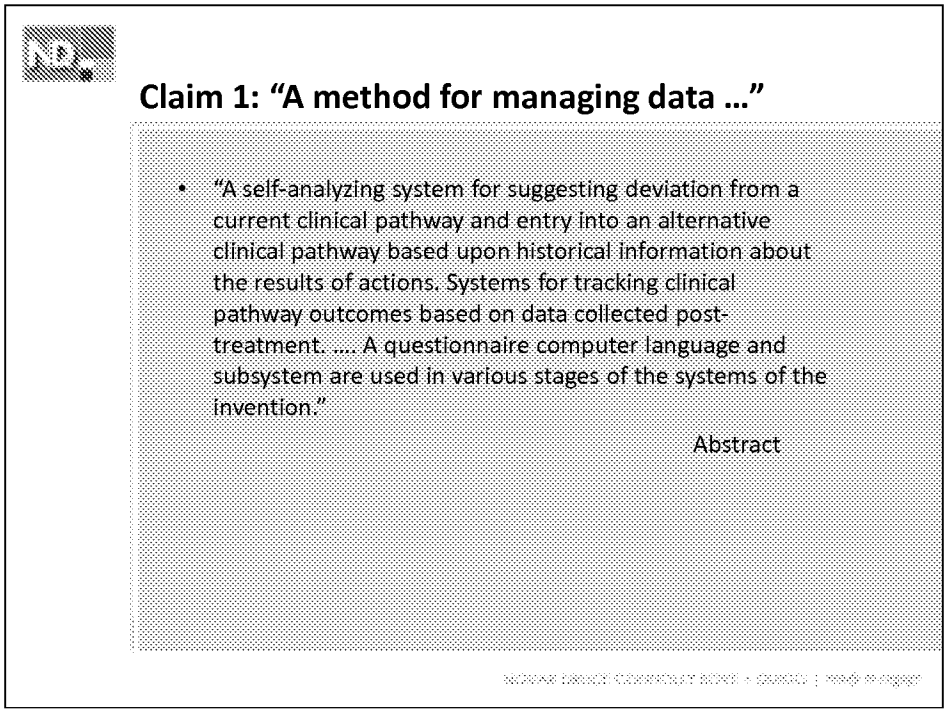
Because the standards of claim interpretation used in the courts in patent litigation are different from the claim interpretation standards used in the Office in claim examination proceedings (including reexamination), any claim interpretations submitted herein for the purpose of demonstrating an SNQ are neither binding upon Requester in any litigation related to the '816 patent; nor do such claim interpretations necessarily correspond to the construction of claims under the legal standards that are mandated to be used by the Courts in patent litigation. *See* MPEP § 2286 II (determination of an SNQ is made independently of a court's decision on validity because of different standards of proof and claim interpretation employed by the District Courts and the Office); *see also Trans Texas Holding*, 498 F.3d at 1297-98; *In re Zletz*, 893 F.2d 319, 322, 13 USPQ2d 1320, 1322 (Fed. Cir. 1989).

The interpretation and/or construction of the claims in the '816 patent presented either implicitly or explicitly herein should not be viewed as constituting, in whole or in part, Requester's own interpretation and/or construction of such claims, but instead should be viewed

as constituting an interpretation and/or construction of such claims as may be raised by the Patent Owner through a broadest reasonable claim construction. In fact, Requester expressly reserves the right to present its own interpretation of such claims at a later time, which interpretation may differ, in whole or in part, from that presented herein.

III. POWERPOINT REGARDING UNPATENTABILITY OF CLAIM 1 OF THE '816 PATENT

Prior to instituting this ex parte proceeding in the Office, attached document OTH-A was presented to patent owner Macrosolve to demonstrate that exemplary claim 1 of the '816 patent is unpatentable. While not explicitly stated in the presentation, the reference in the PowerPoint is Benigno. The unpatentability portion of the PowerPoint against claim 1 is excerpted below, as a primer of the teachings in the Benigno publication. A more detailed treatment of Benigno is in Sections IV and V, below.



"(a) creating a questionnaire …"

- "Entire questionnaires are versioned, and may be easily modified, or recalled from earlier versions. Questions once entered may be reused in many questionnaires."

12:17-31

"(b) tokenizing said questionnaire …"

- "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."

19:10-24

"(c) establishing a first wireless … connection" (1 of 4)

- "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. … may be accomplished in a variety of ways, including over telephone lines, via a wireless connection (cellular or otherwise), via the Internet, etc."

47:6-13

"(c) establishing a first wireless … connection" (2 of 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 1B."

47:6-8

"(c) establishing a first wireless…connection" (3 of 4)

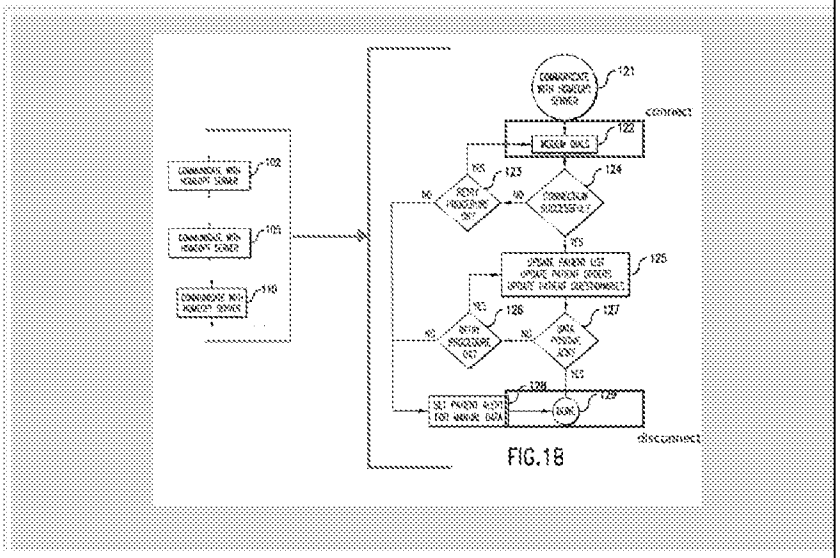


FIG.1B

"(c) establishing a first wireless…connection" (4 of 4)

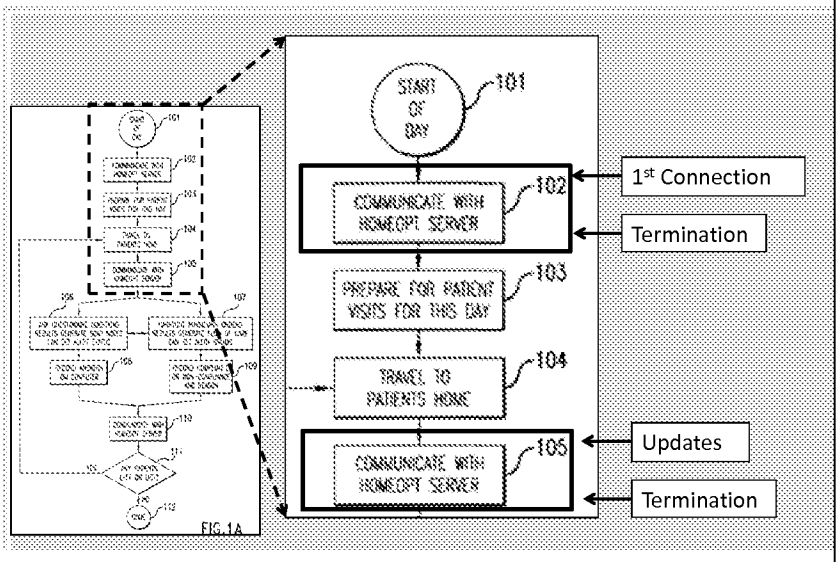


FIG.1A

"(d) transmitting said plurality of tokens…"

- "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)."

46:4-7

"(e) terminating first wireless … connection…" (1 of 3)

- "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 modem and standard phone lines, via wireless transmission (e.g., cellular, etc.), via the Internet, or via any other communication link."

46:4-9

"(e) terminating first wireless…connection…" (2 of 3)

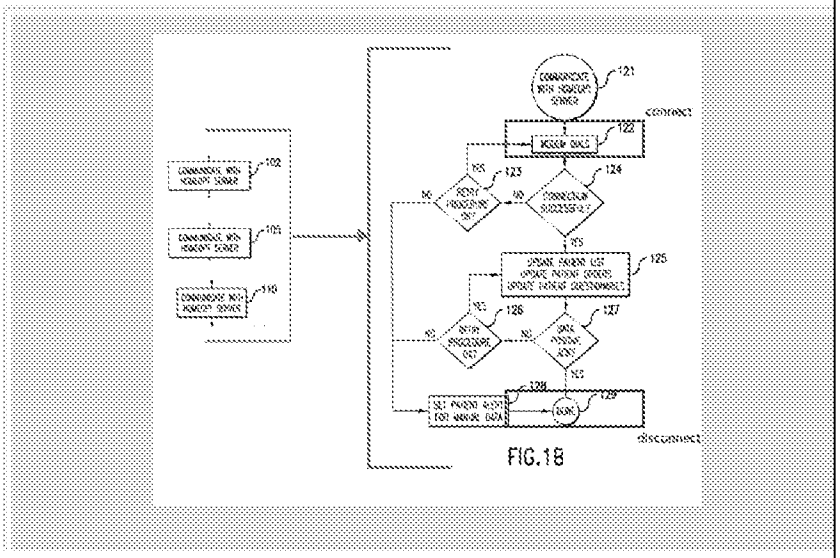
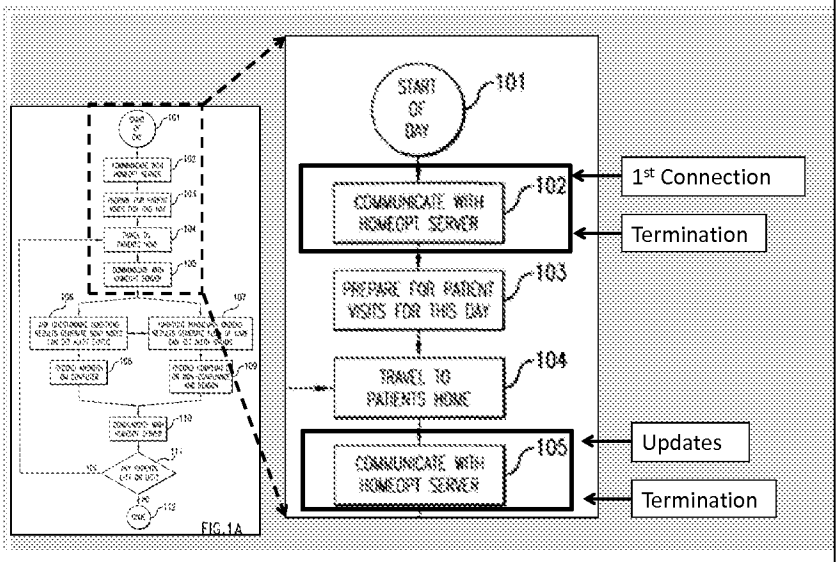


FIG.1B

"(e) terminating first wireless…connection…" (3 of 3)

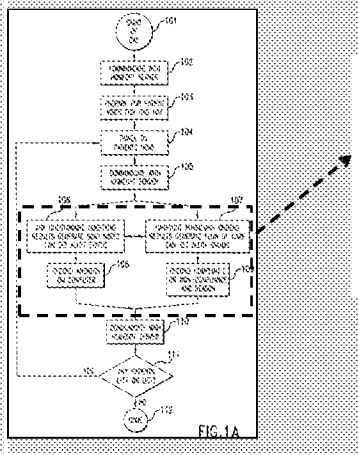


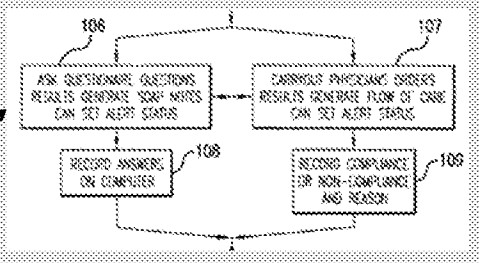
"(f) … executing a portion of said plurality of tokens" (1 of 2)

- "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. … for ultimate retransmission to the server 402."

46:16-20

"(f) … executing a portion of said plurality of tokens" (2 of 2)



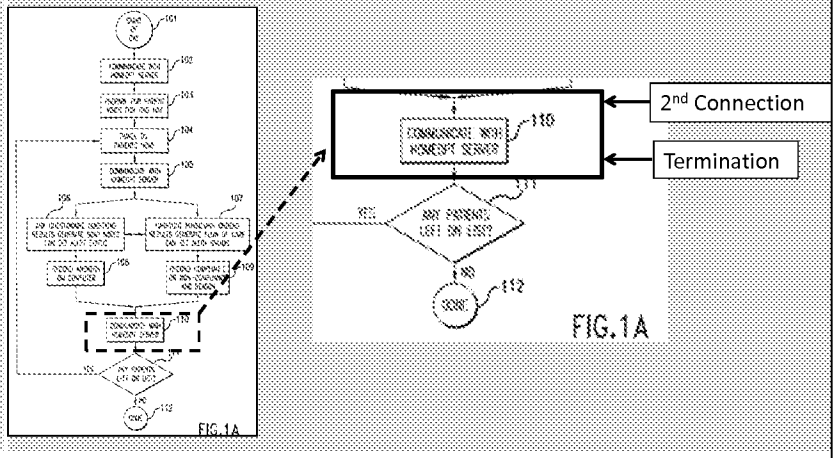


"(g) establishing a second wireless … connection…" (1 of 2)

- "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."

46:30-31

"(g) establishing a second wireless … connection…" (2 of 2)



"(h) after said second [connection]… transmitting… response"

- "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. Again, all such stored data may later be transmitted back to the server 402."

  46:16-28

- "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."

  46:30-31

"(i) storing said transmitted response at said server"

- "The questionnaire and its answers are all stored."

  23:10

- "Again, all such stored data may later be transmitted back to the server 402."

  46:27-28

IV. SUMMARY OF EACH SUBSTANTIAL NEW QUESTIONS OF PATENTABILITY UNDER 37 C.F.R. § 1.510 (B)(2)

A. ROSSMANN IN VIEW OF RAPPAPORT RAISES A SUBSTANTIAL NEW QUESTION OF PATENTABILITY WITH RESPECT TO CLAIMS 1-3 AND 5-14

The following prior art is applied in the SNQ presented herein:

- European Patent Application EP 0779,759 to Rossmann ("Rossmann") – filed on December 11, 1996 and published on June 18, 1997 making it effective prior art under 35 U.S.C. § 102; and
- U.S. Patent No. 6,477,373 to Rappaport et al. ("Rappaport") – filed on August 10, 2000, claiming priority to August 10, 1999, and issued November 5, 2002, making it effective prior art under 35 U.S.C. § 102.

Rossmann and Rappaport were never presented individually or in combination to the Office during the original prosecution. Accordingly, by teaching the limitations of the claims that the Office found missing in the prior art, this combination raises a substantial new question of patentability.

In the prosecution history, the Office found that:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. Rossmann in view of Rappaport teaches these limitations.

Rossmann teaches a card deck wherein each of the cards is a single operation and can be communicated to a computer from a server and from a computer to a server through any known two-way data communication network. Rossmann p. 15, lines 8-12, p. 14, lines 35-38, p. 26, lines 5-6, and p. 28, lines 39-41. Further, since each of the cards in the card deck can be transmitted through a single operation, the connection is effectively established and terminated with each transmission. In addition, Rossmann is combined with Rappaport which teaches that the number of channels available for cellular sessions is limited, and data transfers can be

interrupted (i.e., terminated) to allow for real-time communication. Rappaport at 7:44-63. Based on this combination, it would be obvious to a person of ordinary skill in the art to terminate and reestablish data communications between the mobile device and server of Rossmann.

Furthermore, Rappaport teaches that the data can be processed while the data connection is not in use, because the mobile device continues to function "undisturbed by link failures" and "in a manner that is transparent to end users." Rappaport at 2:44-58. Accordingly, the mobile devices can continue to process data in a transparent manner, as if the data connection is still available, in anticipation of reestablishing the connection.

After the user response data is processed at the mobile device, the cards are then transmitted to a server for collection and processing as appropriate within the system taught by Rossmann. Rossmann p. 9, lines 15-18; p. 11, line 53 – p. 12, line 2, and p. 15, lines 23-27.

The technological features found absent in the prior art were:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. However, Rossmann in view of Rappaport teaches these technological features. As discussed above, Rossmann teaches a system involving a card deck wherein the server and mobile device can communicate the cards to the other device. Further, Rossmann can be combined with Rappaport which teaches that the cards are transmitted between the server and a mobile device.

Accordingly, an Examiner would consider Rossmann in view of Rappaport and the knowledge of a person of ordinary skill in the art important in determining patentability as it presents the new technological feature that was found absent in the prior art and deemed important during the prosecution of the '816 patent. *See* Rossmann p. 15, lines 8-12, 14, lines 35-38, p. 26, lines 5-6, and p. 28, lines 39-41; Rappaport at 7:44-63. Such an SNQ is not cumulative,

as the references meet all the claim limitations for claims 1-3 and 5-14 and the references were not presented or discussed in combination during the prosecution of the '816 patent.

In view of the above, reexamination should be ordered, and the aforementioned claim should be rendered null, void, and otherwise unenforceable in light of this reexamination request.

B. ROSSMANN IN VIEW OF RAPPAPORT AND BOWEN RAISES A SUBSTANTIAL NEW QUESTION OF PATENTABILITY WITH RESPECT TO CLAIMS 4

The following prior art is applied in the SNQ presented herein:

- European Patent Application EP 0779,759 to Rossmann ("Rossmann") – filed on December 11, 1996 and published on June 18, 1997 making it effective prior art under 35 U.S.C. § 102; and
- U.S. Patent No. 6,477,373 to Rappaport et al. ("Rappaport") – filed on August 10, 2000, claiming priority to August 10, 1999, and issued November 5, 2002, making it effective prior art under 35 U.S.C. § 102.
- U.S. Patent No. 5,442,786 to Bowen ("Bowen") – filed on April 28, 1994 and issued on August 15, 1995 making it effective prior art under 35 U.S.C. § 102.

Rossmann, Rappaport and Bowen were never presented individually or in combination to the Office during the original prosecution. Accordingly, by teaching the limitations of the claims that the Office found missing in the prior art, this combination raises a substantial new question of patentability.

In the prosecution history, the Office found that:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. Rossmann in view of Rappaport teaches these limitations.

Rossmann teaches a card deck wherein each of the cards is a single operation and can be communicated to a computer from a server and from a computer to a server through any known two-way data communication network. Rossmann p. 15, lines 8-12, p. 14, lines 35-38, p. 26,

lines 5-6, and p. 28, lines 39-41. Further, since each of the cards in the card deck can be transmitted through a single operation, the connection is effectively established and terminated with each transmission. In addition, Rossmann is combined with Rappaport which teaches that the number of channels available for cellular sessions is limited, and data transfers can be interrupted (i.e., terminated) to allow for real-time communication. Rappaport at 7:44-63. Based on this combination, it would be obvious to a person of ordinary skill in the art to terminate and reestablish data communications between the mobile device and server of Rossmann.

Furthermore, Rappaport teaches that the data can be processed while the data connection is not in use, because the mobile device continues to function "undisturbed by link failures" and "in a manner that is transparent to end users." Rappaport at 2:44-58. Accordingly, the mobile devices can continue to process data in a transparent manner, as if the data connection is still available, in anticipation of reestablishing the connection.

After the user response data is processed at the mobile device, the cards are then transmitted to a server for collection and processing as appropriate within the system taught by Rossmann. Rossmann p. 9, lines 15-18; p. 11, line 53 – p. 12, line 2, and p. 15, lines 23-27.

Finally, claim 4, requires, amongst other limitations, "assigning at least one token to each branch in said questionnaire to identify the required program control associated with said branch." Bowen teaches that it was well known in the art to develop questionnaires based on "hierarchical data tree[s]" where "the system first creates a vertical leg of the data tree, before creating horizontal branches…." Bowen at Abstract. Accordingly, since Rossmann teaches that the application discussed therein can be used universally on any two way communication device, it would be obvious to a person of ordinary skill in the art that the hierarchical tree described by Bowen can be used to create a questionnaire in Rossmann. Rossmann at page 3, lines 41-49.

The technological features found absent in the prior art were:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. However, Rossmann in view of Rappaport and Bowen teaches these technological features. As discussed above, Rossmann teaches a system involving a card deck wherein the server and mobile device can communicate the cards to the other device. Further, Rossmann can be combined with Rappaport and Bowen which teach that the cards are transmitted between the server and a mobile device and that the questionnaire can be a tree structure.

Accordingly, an Examiner would consider Rossmann in view of Rappaport and Bowen and the knowledge of a person of ordinary skill in the art important in determining patentability as it presents the new technological feature that was found absent in the prior art and deemed important during the prosecution of the '816 patent. *See* Rossmann p. 15, lines 8-12, 14, lines 35-38, p. 26, lines 5-6, and p. 28, lines 39-41; Rappaport at 7:44-63; Bowen at Abstract. Such an SNQ is not cumulative, as the references meet all the claim limitations for claim 4 and the references were not presented or discussed in combination during the prosecution of the '816 patent.

In view of the above, reexamination should be ordered, and the aforementioned claim should be rendered null, void, and otherwise unenforceable in light of this reexamination request.

C. ROSSMANN IN VIEW OF FALLS RAISES A SUBSTANTIAL NEW QUESTION OF PATENTABILITY WITH RESPECT TO CLAIMS 1-14

The following prior art is applied in the SNQ presented herein:

- European Patent Application EP 0779,759 to Rossmann ("Rossmann") – filed on December 11, 1996 and published on June 18, 1997 making it effective prior art under 35 U.S.C. § 102; and
- U.S. Patent No. 5,991,771 to Falls et al. ("Falls") – filed on July 3, 1997 and published November 23, 1999, making it effective prior art under 35 U.S.C. § 102.

Rossmann and Falls were never presented individually or in combination to the Office during the original prosecution. Accordingly, by teaching the limitations of the claims that the Office found missing in the prior art, this combination raises a substantial new question of patentability.

In the prosecution history, the Office found that:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. Rossmann in view of Falls teaches these limitations.

Rossmann teaches a card deck wherein each of the cards is a single operation and can be communicated to a computer from a server and from a computer to a server through any known two-way data communication network. Rossmann p. 15, lines 8-12, p. 14, lines 35-38, p. 26, lines 5-6, and p. 28, lines 39-41. Further, since each of the cards in the card deck can be transmitted through a single operation, the connection is effectively established and terminated with each transmission. In addition, Rossmann is combined with Falls which teaches that the even though the system is disconnected from the network communications, a "virtual network" will allow the mobile device to continue normal operations. Falls at Abstract and 3:16-35. The questionnaire can then be synchronized upon reestablishing the network connection. Falls at Abstract and 3:16-35. Based on this combination, it would be obvious to a person of ordinary skill in the art to terminate and reestablish data communications between the mobile device and server of Rossmann.

After the user response data is processed at the mobile device, the cards are then transmitted to a server for collection and processing as appropriate within the system taught by Rossmann. Rossmann p. 9, lines 15-18; p. 11, line 53 – p. 12, line 2, and p. 15, lines 23-27.

The technological features found absent in the prior art were:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless

> modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. However, Rossmann in view of Falls teaches these technological features. As discussed above, Rossmann teaches a system involving a card deck wherein the server and mobile device can communicate the cards to the other device. Further, Rossmann can be combined with Falls which teaches that the mobile devices can be disconnectable from the server, and that the cards transmitted can be synchronized after a disconnection occurs. Falls at Abstract, 3:16-35, 5:21-31, and 35:47-63.

Accordingly, an Examiner would consider Rossmann in view of Falls and the knowledge of a person of ordinary skill in the art important in determining patentability as it presents the new technological feature that was found absent in the prior art and deemed important during the prosecution of the '816 patent. *See* Rossmann p. 15, lines 8-12, 14, lines 35-38, p. 26, lines 5-6, and p. 28, lines 39-41; Falls at Abstract, 3:16-35, 5:21-31, and 35:47-63. Such an SNQ is not cumulative, as the references meet all the claim limitations for claims 1-14 and the references were not presented or discussed in combination during the prosecution of the '816 patent.

In view of the above, reexamination should be ordered, and the aforementioned claim should be rendered null, void, and otherwise unenforceable in light of this reexamination request.

D. BENIGNO IN VIEW OF FALLS RAISES A SUBSTANTIAL NEW QUESTION OF PATENTABILITY WITH RESPECT TO CLAIMS 1-14

The following prior art is applied in the SNQ presented herein:

- PCT Published Application WO 99/33390 to Benigno ("Benigno") – filed on December 23, 1998 and published on July 8, 1999 making it effective prior art under 35 U.S.C. § 102; and
- U.S. Patent No. 5,991,771 to Falls et al. ("Falls") – filed on July 3, 1997 and published November 23, 1999, making it effective prior art under 35 U.S.C. § 102.

Benigno and Falls were never presented individually or in combination to the Office during the original prosecution. Accordingly, by teaching the limitations of the claims that the Office found missing in the prior art, this combination raises a substantial new question of patentability.

In the prosecution history, the Office found that:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. Benigno in view of teaches these limitations.

Benigno teaches a questionnaire based on creating a standard of care for treatment of patients that keeps nurses and doctors in constant communication. Benigno at 46:4-9 and 22-24. The nurse is able to answer questions in the questionnaire and based on the responses provided by the patient, the information is updated in the server and subsequent questions are asked. Benigno at 12:17-31. This also allows for individual questions to be used throughout multiple questionnaires, thereby increasing efficiency of the questionnaire database. *Id.* The individual questions are "tokenized representations" that are communicated between the server and the mobile device via wireless network connections. Benigno at 19:10-24, 13:1-10, and 46:4-9. The mobile device can be disconnected from the network communications due to losing the connection as is inevitable in wireless communication or due to the nurse closing the connection. Benigno at 46:4-24 and FIG.1A. Further, as shown in FIG. 1B, the modem of the system dials at the beginning of each communication step 102, 105, and 110. Dialing each time is only necessary if the modem is disconnected. The nurse can continue to input data into the questionnaire, even though the system is disconnected from the network communications, further, when combined with Falls, the combination further teaches that the questionnaire can continue to be used and a "virtual network" will allow the mobile device to continue normal operations when there is no network connection. Benigno at 46:16-28; Falls at Abstract and 3:16-35. The questionnaire can then be synchronized upon reestablishing the network connection. Benigno at FIG. 1B; Falls at Abstract and 3:16-35. The questionnaire is then stored. Benigno at 23:10. The application of this combination of references is further shown in Annotated FIG. 1A, below.

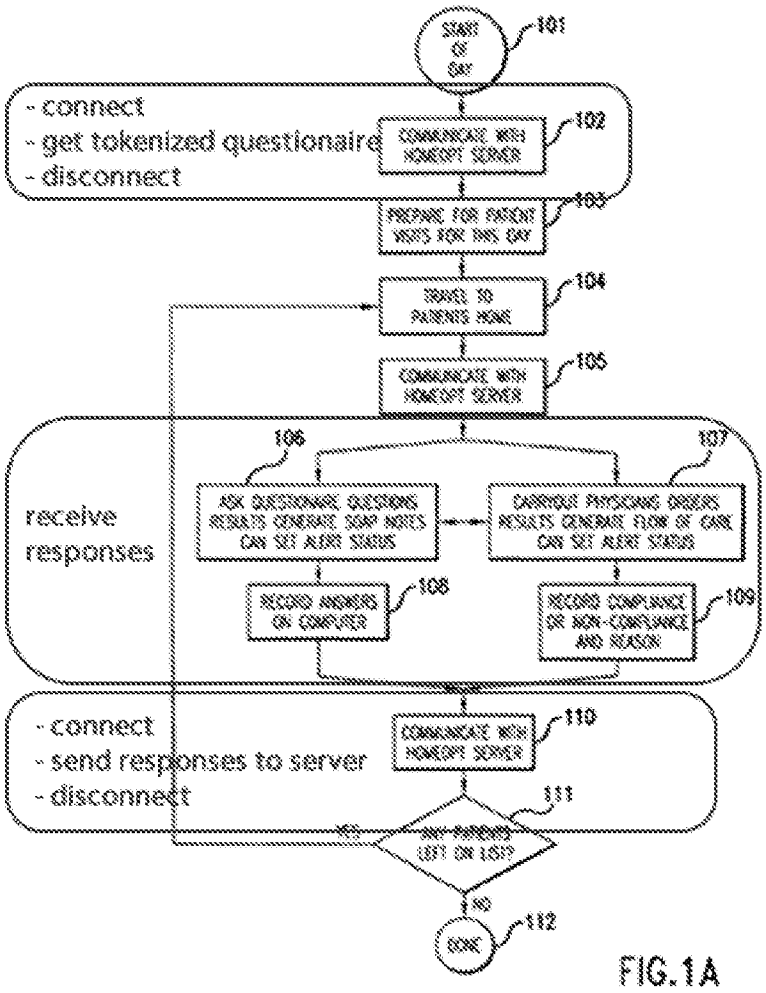


FIG.1A

As previously stated, Benigno in view of Falls teaches the technological features allegedly missing in the prior art during the original prosecution. As discussed above, Benigno teaches a questionnaire based medical communications system that communicates between mobile devices and a server. Further, Benigno can be combined with Falls which teaches that the questions are synchronized between the server and a mobile device after periods without network communications.

Accordingly, an Examiner would consider Benigno in view of Falls and the knowledge of a person of ordinary skill in the art important in determining patentability as it presents the new technological feature that was found absent in the prior art and deemed important during the prosecution of the '816 patent. *See* Benigno at 46:16-28; Falls at Abstract and 3:16-35. Such an SNQ is not cumulative, as the references meet all the claim limitations for claims 1-14 and the

references were not presented or discussed in combination during the prosecution of the '816 patent.

In view of the above, reexamination should be ordered, and the aforementioned claim should be rendered null, void, and otherwise unenforceable in light of this reexamination request.

E. BENIGNO IN VIEW OF RAPPAPORT RAISES A SUBSTANTIAL NEW QUESTION OF PATENTABILITY WITH RESPECT TO CLAIMS 1-14

The following prior art is applied in the SNQ presented herein:

- PCT Published Application WO 99/33390 to Benigno ("Benigno") – filed on December 23, 1998 and published on July 8, 1999 making it effective prior art under 35 U.S.C. § 102; and
- U.S. Patent No. 6,477,373 to Rappaport et al. ("Rappaport") – filed on August 10, 2000, claiming priority to August 10, 1999, and issued November 5, 2002, making it effective prior art under 35 U.S.C. § 102.

Benigno and Rappaport were never presented individually or in combination to the Office during the original prosecution. Accordingly, by teaching the limitations of the claims that the Office found missing in the prior art, this combination raises a substantial new question of patentability.

In the prosecution history, the Office found that:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. Benigno in view of teaches these limitations.

Benigno teaches a questionnaire based on creating a standard of care for treatment of patients that keeps nurses and doctors in constant communication. Benigno at 46:4-9 and 22-24. The nurse is able to answer questions in the questionnaire and based on the responses provided by the patient, the information is updated in the server and subsequent questions are asked.

Benigno at 12:17-31. This also allows for individual questions to be used throughout multiple questionnaires, thereby increasing efficiency of the questionnaire database. *Id* The individual questions are "tokenized representations" that are communicated between the server and the mobile device via wireless network connections. Benigno at 19:10-24, 13:1-10, and 46:4-9. The mobile device can be disconnected from the network communications due to losing the connection as is inevitable in wireless communication or due to the nurse closing the connection. Benigno at 46:4-24 and FIG.1A. Further, as shown in FIG. 1B, the modem of the system dials at the beginning of each communication step 102, 105, and 110. Dialing each time is only necessary if the modem is disconnected. The nurse can continue to input data into the questionnaire, even though the system is disconnected from the network communications. Benigno at 46:16-28. The questionnaire is then stored. Benigno at 23:10.

Furthermore, Rappaport teaches that the data can be processed while the data connection is not in use, because the mobile device continues to function "undisturbed by link failures" and "in a manner that is transparent to end users." Rappaport at 2:44-58. Accordingly, the mobile devices can continue to process data in a transparent manner, as if the data connection is still available, in anticipation of reestablishing the connection. This allows a nurse to continue to input answers to medical questions, even when the connection has failed.

The application of the combined references is further shown in Annotated FIG. 1A, below.

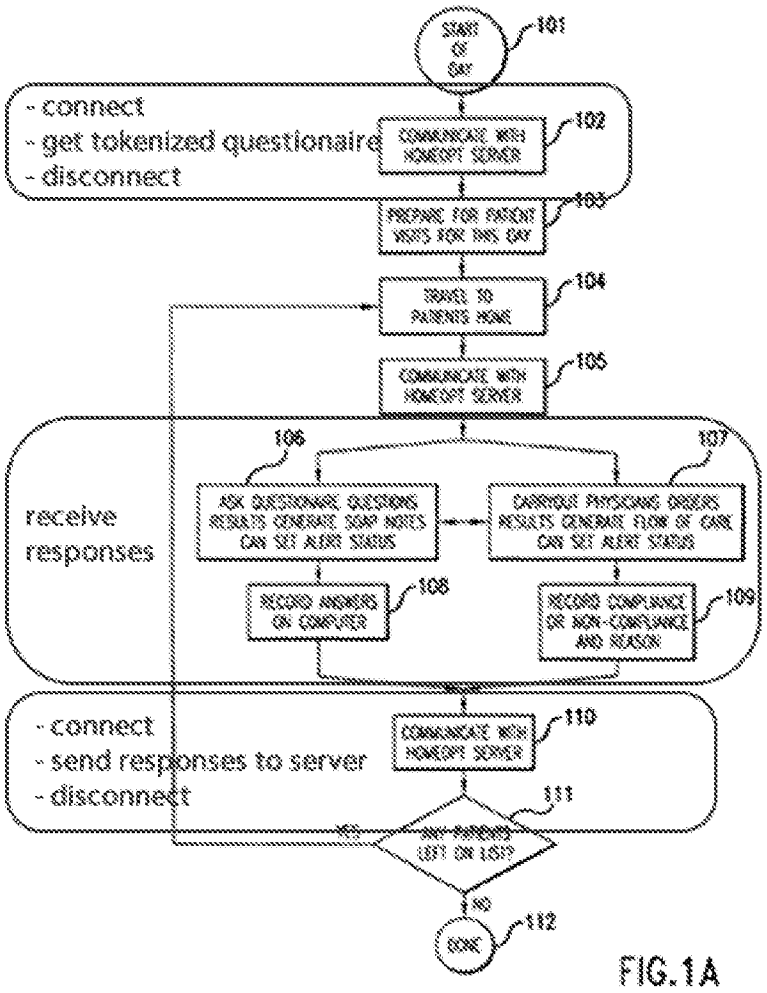


FIG.1A

As previously stated, Benigno in view of Rappaport teaches the technological features allegedly missing in the prior art during the original prosecution. As discussed above, Benigno teaches a questionnaire based medical communications system that communicates between mobile devices and a server. Further, Benigno can be combined with Rappaport which teaches that the questions can continue to be answered even when the mobile device loses network communications, since the questions can be sent to the server when the communications are restored.

Accordingly, an Examiner would consider Benigno in view of Rappaport and the knowledge of a person of ordinary skill in the art important in determining patentability as it presents the new technological feature that was found absent in the prior art and deemed important during the prosecution of the '816 patent. *See* Benigno at 46:16-28; Rappaport at 7:44-63. Such an SNQ is not cumulative, as the references meet all the claim limitations for claims 1-

14 and the references were not presented or discussed in combination during the prosecution of the '816 patent.

In view of the above, reexamination should be ordered, and the aforementioned claim should be rendered null, void, and otherwise unenforceable in light of this reexamination request.

F. WRIGHT IN VIEW OF WARTHEN, RAPPAPORT, BROOKLER , AND ROSSMANN RAISES A SUBSTANTIAL NEW QUESTION OF PATENTABILITY WITH RESPECT TO CLAIMS 1, 2, 5-7, AND 11-14

The following prior art is applied in the SNQ presented herein:

- U.S. Patent No. 5,704,029 to Wright ("Wright")-filed on May 23, 1994 and issued on December 30, 1997, making it effective prior art under 35 U.S.C. § 102.
- PA-B U.S. Patent No. 6,477,373 to Rappaport et al. ("Rappaport")-filed on August 10, 2000, claiming priority to August 10, 1999, and issued November 5, 2002, making it effective prior art under 35 U.S.C. § 102.
- PA-C U.S. Patent No. 6,584,464 to Warthen ("Warthen")-filed on March 19, 1999 and issued on June 24, 2003, making it effective prior art under 35 U.S.C. §102.
- PA-D U.S. Patent App. No. 2002/0007303 to Brookler et al. ("Brookler")-filed on April 30, 2001, claims priority to an application filed on May 1, 2000, and published on January 17, 2002, making it effective prior art under 35 U.S.C. § 102.
- European Patent Application EP 0779,759 to Rossmann ("Rossmann")-filed on December 11, 1996 and published on June 18, 1997 making it effective prior art under 35 U.S.C. § 102

Wright, Rappaport, Warthen, Brookler, and Rossmann were never presented individually or in combination to the Office during the original prosecution. Accordingly, by teaching the limitations of the claims that the Office found missing in the prior art, this combination raises a substantial new question of patentability.

In the prosecution history, the Office found that:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3 Wright in view of Rappaport, Warthen, and Brookler teaches these limitations.

Wright and Warthen teach a system that creates a questionnaire and tokenizes that data of that questionnaire. Wright at ABSTRACT and 13:38-67; Warthen at ABSTRACT and 2:1-11. Rappaport then teaches that the systems taught by Wright and Warthen are both able to establish two-way communications between the mobile device and the server. Rappaport at Abstract and 16:30-36. Further, Rappaport teaches that the number of channels available for cellular sessions is limited, and data transfers can be interrupted (i.e., terminated) to allow for real-time communication. Rappaport at 7:44-63. Rappaport also teaches that the data can be processed while the data connection is not in use, because the mobile device continues to function "undisturbed by link failures" and "in a manner that is transparent to end users." Rappaport at 2:44-58. Accordingly, the mobile devices can continue to process data in a transparent manner, as if the data connection is still available, in anticipation of reestablishing the connection. Finally, Brookler teaches that the tokenized data can be transmitted using multiple network connections, and processed at the server. Brookler at [0033] and FIG. 1.

Also, Rossmann is added to teach that the system described by the above combination can include a step where a report can be printed. Rossmann p. 11, lines 4-8

The technological features found absent in the prior art were:

> The cited prior arts [sic] fail to disclose or suggest transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection, terminating said first wireless modem or wireless LAN network connection with said remote computing device, 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, establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server, 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 in conjunction with all other limitations in the claim.

See Notice of Allowance issued September 7, 2010 at 2-3. However, Wright in view of Rappaport, Warthen, and Brookler teaches these technological features. As discussed above, Wright in view of Rappaport, Warthen, and Brookler teaches a system involving a tokenized

questionnaire wherein the server and mobile device can communicate the data to the other device. Further, Rappaport teaches that the data are transmitted between the server and a mobile device.

Accordingly, an Examiner would consider Wright in view of Rappaport, Warthen, and Brookler and the knowledge of a person of ordinary skill in the art important in determining patentability as it presents the new technological feature that was found absent in the prior art and deemed important during the prosecution of the '816 patent. *See* Wright at ABSTRACT and 13:38-67; Warthen at ABSTRACT and 2:1-11; Rappaport at 7:44-63 and 16:30-36; and Brookler at [0033] and FIG. 1. Such an SNQ is not cumulative, as the references meet all the claim limitations for claims 1, 2, 5-7, and 11-14 and the references were not presented or discussed in combination during the prosecution of the '816 patent.

In view of the above, reexamination should be ordered, and the aforementioned claim should be rendered null, void, and otherwise unenforceable in light of this reexamination request.

V. MANNER OF APPLYING THE CLAIMS AS REQUIRED BY 37 C.F.R. § 1.510 (B)(2)

A. CLAIMS 1-3 AND 5-14 OF THE '816 PATENT ARE RENDERED OBVIOUS BY ROSSMANN IN VIEW OF RAPPAPORT AND THE KNOWLEDGE OF A PERSON OF ORDINARY SKILL IN THE ART

Please see the below claim chart that applies the teachings of Rossmann in view of Rappaport to claims 1-3 and 5-14 of the '816 patent.

Reasons to Combine:

A person of ordinary skill in the art would recognize that Rossmann teaches that a mobile device that connects to a server and transmits information between the two devices. *See* Rossmann p. 6, lines 31-37 and 14, lines 35-38. It would have been obvious to combine Rossmann 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.

Brief Description of Application within Charts Below:

Rossmann teaches a card deck wherein each of the cards is a single operation and is a representation of a question in a questionnaire, this is tokenization within the broadest reasonable interpretation of the claims of the '816 patent. Rossmann, p. 10, lines 38-48, 11, lines 15-16, 14,

lines 55-58, 15, lines 2-7, and 15, lines 56-57. These tokens, i.e., cards, can be communicated to a computer from a server and from a computer to a server through any known two-way data communication network. Rossmann p. 15, lines 8-12, p. 14, lines 35-38, p. 26, lines 5-6, and p. 28, lines 39-41. The method described by Rossmann can be supplemented by the addition of Rappaport, which teaches that mobile devices can terminate and reestablish connections. Rappaport at 7:44-63. Accordingly, the cards and card decks can be communicated between a mobile device and a server.

As will be appreciated with reference to the citations contained in the charts below, each of the claim limitations is explicitly taught by the combination of Rossmann and Rappaport. Accordingly, the rejections should be adopted.

CLAIM 1	**Teachings of Rossmann in view of Rappaport**
1. A method for managing data including the steps of:	As indicated above, the two-way data communication device of this invention utilizes a client module to transmit a message including a resource locator selected by the user over the two-way data communication network to a server on a server computer on the computer network. For example, the computer network can be a corporate wide area network, a corporate local area network, the Internet, or any combination of computer networks. The server processes the message, i.e., executes the application addressed by the resource locator and transmits a response over the two-way data communication network to the two-way data communication device, which stores the response in a memory. The client module interprets the response and generates a user interface using information in the response. In one embodiment, the user interface includes at least one user data input option that is associated with a resource locator. Rossmann p. 4, lines 1-9
(a) creating a questionnaire comprising a series of questions;	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card

	deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the **appropriate purchase order** and transmits the purchase order as a card deck to the client process in cellular telephone 100 over airnet network 150. The client process interprets the card deck and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines, 38-48 For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12
(b) tokenizing said questionnaire; thereby producing a plurality of tokens representing said questionnaire;	In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the appropriate purchase order and **transmits the purchase order as a card deck** to the client process in cellular telephone 100 over airnet network 150.

The client process interprets the card deck and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209.

Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105.

After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F.
Rossmann, p. 10, lines 38-48

In addition, the client process using the information transmitted from server computer 121, **i.e., the cards, generates a wide-variety of user interfaces as illustrated in Figures 2A to 2H.**
Rossmann p. 11, lines 15-16

Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression.
Rossmann p. 14, lines 55-58

Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network.
Rossmann p. 15, lines 2-7

For example, if the user of cellular telephone 700 requested a fax as in Figure 2F, the HTTP request identifies a common gateway interface application in CGI programs 761 that accepts as input data the telephone number and grabs the information to be faxed. The CGI application generates an e-mail transmission to the fax gateway. **Similarly, for a stock quote**, server 749, in response to the HTTP request, launches a

	common gateway interface application that sends out a stock query over Internet 140 to a stock quote service provider using the ticker tape symbol passed as input data by server 749 to the common gateway interface application. When the response to the stock query is received, the common gateway interface application builds a PIDL deck that includes the data in the response to the stock query. The interface presented in Table 7 for TIL manager module 1403 is designed with the assumption that TIL is a direct tokenization of PIDL as described in Appendix I. Rossmann p. 15, lines 56-57
(c) establishing a first wireless modem or wireless LAN network connection with a remote computing device;	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38
(d) transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection;	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a

	binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression. Rossmann p. 14, lines 55-58 Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network. Rossmann p. 15, lines 2-7
(e) terminating said first wireless modem or wireless LAN network connection with said remote computing device;	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38 After the transaction is logged, processing transfers to transmit result 1317. In transmit result 1317, ANT request processor 1204 returns the deck to client 702. After the deck is transmitted, ANT request processor 1204 is terminated. Rossmann p. 26, lines 5-6. Routine NM_lnit initializes network manager module 1402 and so is called before any other calls in network manager module 1402. Routine NM_ Terminate closes processing of network manager module 1402 and so is called after all other calls in network manager module 1402. Rossmann p. 28, lines 39-41

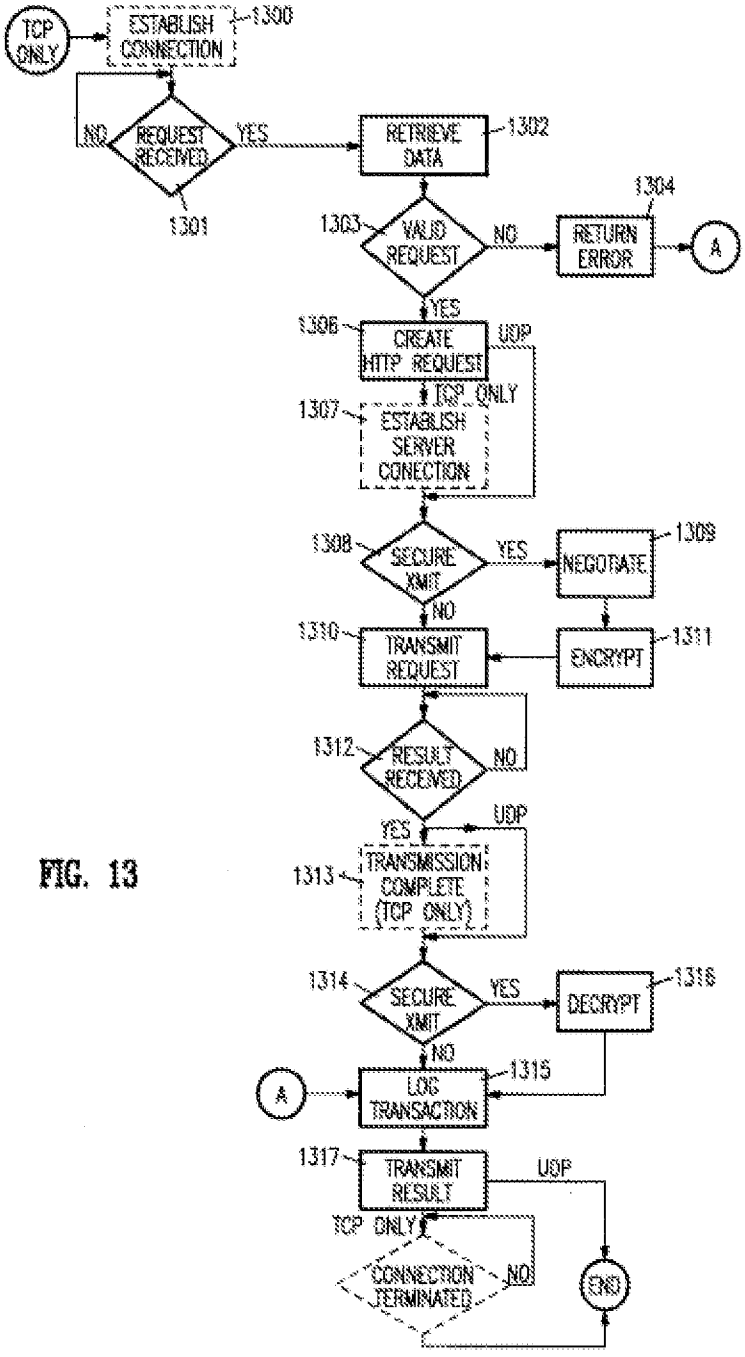


FIG. 13

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 col. 7, lines 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 col. 2, lines 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;	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12 The client process in cellular telephone 100 interprets the display card that includes image and text data and generates screen display 300 on display screen 105 (Fig. 3A). Screen display 300 includes a home key 301, and an info key 302. When the user selects home key 301, the user is returned to the home screen. Info key 302 functions in a manner similar to that described above for info key 205. When the user presses a predetermined key, the client process interprets the choice card and a second screen display 304 (Fig. 3B) is driven on display screen 105. Screen display 304 is a menu of the personal

	information that is stored on *server* computer 131 for use by the user of cellular telephone 100. Multi-display screen card indicator 203, e.g., the hand with a finger pointing down, illustrates to the user that the list has additional items that appear on the next screen display. Screen display 304 also indicates the number of E-mail messages, faxes, and voice messages waiting for the user. Rossmann p. 11, lines 43-52
(g) establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server;	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38
(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	When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 The user scrolls the screen display line by line until screen display 305 is on display screen 105. Initially, the fourth item in the menu is not highlighted. In this example, the user presses the four key on the keypad of cellular telephone 100 to view the user's schedule. In response to the key press, the client module in cellular telephone 100 transmits a message, including a resource locator associated with the menu item selected by pressing the four key, to server computer 131 using data capable cellular telephone network 110 and corporate local area network

	130. In response to the message, server computer 131 executes the application identified in the resource locator. Upon completion of the execution, server computer 131 transmits, over corporate local area network 130 and data capable cellular telephone network 110 to cellular telephone 100, a card deck that includes a choice card that describes the user's schedule for that day. Rossmann p. 11, line 53 – Rossmann p. 12, line 2 As indicated above, each interaction with the user of cellular telephone 700 is described by a deck or a series of decks. Logically, the user retrieves a terminal interaction language deck stored in a memory 716 of cellular telephone 700 after receipt from computer 743 over CDPD network 710. The user reviews the information displayed by cards in the deck and makes choices and/or enters requested information and then requests another deck, as described above with respect to Figures 2A to 2H, for example. Rossmann p. 15, lines 23-27
(i) storing said transmitted response at said server.	The server processes the message, i.e., executes the application addressed by the resource locator and transmits a response over the two-way data communication network to the two-way data communication device, which **stores** the response in a memory. Rossmann p. 4, lines 5-7 When the user presses a predetermined key, the client process interprets the choice card and a second screen display 304 (Fig. 3B) is driven on display screen 105. Screen display 304 is a menu of the personal information that is stored on *server* computer 131 for use by the user of cellular telephone 100. Multi-display screen card indicator 203, e.g., the hand with a finger pointing down, illustrates to the user that the list has additional items that appear on the next screen display. Screen display 304 also indicates the number of E-mail messages, faxes, and *voice* messages waiting for the user. Rossmann p. 11, lines 47-52 Computer 743 may contain **stored** static telephone interaction description language decks. Computer 743 also generates telephone interaction description language decks in response to data from, or choices made by, the user of cellular telephone 700. Rossmann p. 15, lines 13-15
CLAIM 2	
2. The method for managing data of	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display

claim 1 further comprising the step of: (j) translating said response to a format recognizable by a particular computer program; and	card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by **interpreting** the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines 43-48
(k) accessing the translated response from a computer executing said particular computer program.	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by **interpreting** the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines 43-48

CLAIM 3	
3. The method for managing data of claim 1 wherein step (a) includes the substeps of: (a) creating a questionnaire by:	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the **appropriate purchase order** and transmits the purchase order as a card deck to the client process in cellular telephone 100 over airnet network 150. The client process interprets the card deck and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines, 38-48 For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one

	action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12
(i) entering a series of questions into a questionnaire design computer program;	This invention allows for the first time two-way communications devices such as cellular telephones, two-way pagers, and telephones to become open application platforms which in turn empowers software developers to deliver value-added applications and services to any two-way communication device that incorporates the principles of this invention. This is a radical shift from the current situation where telephones and two-way pagers are closed, proprietary systems. Consequently, an even playing field is created for the market to invent new uses for two-way communication devices and for two-way communication networks. Any entity from corporations to individuals can make new applications available to the installed base of two-way data communication devices that include this invention without physical modification or addition to the two-way communication device. Years after purchase, a two-way communication device incorporating this invention will run all the applications which were developed since its purchase. Rossmann p. 3, lines 41-49 In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs.

	Rossmann p. 9, lines 15-18
(ii) identifying within said questionnaire design computer program the type of response allowed for each question of said series of questions; and	This invention allows for the first time two-way communications devices such as cellular telephones, two-way pagers, and telephones to become open application platforms which in turn empowers software developers to deliver value-added applications and services to any two-way communication device that incorporates the principles of this invention. This is a radical shift from the current situation where telephones and two-way pagers are closed, proprietary systems. Consequently, an even playing field is created for the market to invent new uses for two-way communication devices and for two-way communication networks. Any entity from corporations to individuals can make new applications available to the installed base of two-way data communication devices that include this invention without physical modification or addition to the two-way communication device. Years after purchase, a two-way communication device incorporating this invention will run all the applications which were developed since its purchase. Rossmann p. 3, lines 41-49 When the user selects the at least one user data input option, the client module interprets the selection and if required, appends any input data to the resource allocator associated with the at least one user data input option. The client module transmits a message including the resource locator with any appended input data to the server computer. Alternatively, the resource locator with any appended data can be addressed to another server computer, or can address an object stored in the two-way communication device. If the resource locator addresses an object on a server computer, the client module provides the message to the network interface module which in turn transmits the message over the two-way data communication network. Rossmann p. 4, lines 15-21 An important aspect of this invention is that the message includes all information necessary for the client module to generate the user interface and a particular user interface can be independent from other user interfaces. Unlike prior art systems that gave the user a predetermined menu from which to select items, or limited the user to an E-mail like format, according to the principles of this invention, the user interfaces and possible interactions available to the user are determined only by the applications that developers make available. The possible interactions and user interfaces for one application can be totally different and independent from the possible interactions and user interfaces of another application. Thus, a cellular telephone, two-way pager, and a telephone all truly become an open platform.

	Rossmann at p. 4, lines 29-35 In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18
(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.	In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8
CLAIM 5	
5. A method for modifying a questionnaire used in	Thus, the client module only interprets this information and interacts appropriately with the hardware of the two-way data communication device. Consequently, to update an application requires only **changes** on

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;	the server computer and not changes in each two-way data communication device that communicates with that server computer. This invention eliminates the usual requirement for distribution of application software, and application software updates to the end user of the two-way data communication device. Rossmann p. 4, lines 47-51
(b) tokenizing said at least one incremental change to said questionnaire;	For local services, like local message store, there are two basic approaches that can be used. First, local services are implemented in a CGI-like manner. Each local service has an entry point which is called with an argument list. A TIL deck is returned via the event manager. From that point on, the TIL deck is processed in the standard manner. This approach limits local services to the same constraints as remote services. A less restrictive approach is to allow the local service to field events instead of the standard event loop. The local service would construct TIL cards on-the-fly and feed them to user interface manager 1406. Note that the local service would need to cooperate with the standard event loop with regard to the history, the pushed card list, and any other state that is normally managed by the event loop. Table 4 is a listing of processes for the architecture for navigation manager module 1401. Rossmann p. 26, lines 37-44
(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,	For local services, like local message store, there are two basic approaches that can be used. First, local services are implemented in a CGI-like manner. Each local service has an entry point which is called with an argument list. A TIL deck is returned via the event manager. From that point on, the TIL deck is processed in the standard manner. This approach limits local services to the same constraints as remote services. A less restrictive approach is to allow the local service to field events instead of the standard event loop. The local service would construct TIL cards on-the-fly and feed them to user interface manager 1406. Note that the local service would need to cooperate with the standard event loop with regard to the history, the pushed card list, and any other state that is normally managed by the event loop. Table 4 is a listing of processes for the architecture for navigation manager module 1401. Rossmann p. 26, lines 37-44
(d) incorporating said transmitted tokens into said questionnaire at said loosely networked remote computing	Thus, the client module only interprets this information and interacts appropriately with the hardware of the two-way data communication device. Consequently, to update an application requires only **changes** on the server computer and not changes in each two-way data communication device that communicates with that server computer. This invention eliminates the usual requirement for distribution of

device, thereby modifying said questionnaire.	application software, and application software updates to the end user of the two-way data communication device. Rossmann p. 4, lines 47-51 For local services, like local message store, there are two basic approaches that can be used. First, local services are implemented in a CGI-like manner. Each local service has an entry point which is called with an argument list. A TIL deck is returned via the event manager. From that point on, the TIL deck is processed in the standard manner. This approach limits local services to the same constraints as remote services. A less restrictive approach is to allow the local service to field events instead of the standard event loop. The local service would construct TIL cards on-the-fly and feed them to user interface manager 1406. Note that the local service would need to cooperate with the standard event loop with regard to the history, the pushed card list, and any other state that is normally managed by the event loop. Table 4 is a listing of processes for the architecture for navigation manager module 1401. Rossmann p. 26, lines 37-44
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.	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37
CLAIM 7	
7. The method of claim 1 further including performing at least the steps (c)-(k) for at least two different remote computing device	Each wireless communication device 100 that includes this invention can communicate over airnet network 150 with any server computer 121, 131, and 141 on airnet network 150 that includes at least one application that communicates and interacts with the processes of this invention that are included within device 100. Thus, device 100 can access information on the computer network and provide information to the computer network. Similarly, a two-way pager 101, and a telephone

types using the same tokens.	102 with a modem 103, that each include this invention, can communicate over airnet network 150 with any of server computers 121, 131, and 141 that includes at least one application that communicates and interacts with the processes of this invention that are included within devices 101 and 102. Rossmann p. 6, lines 38-44
CLAIM 8	
8. A method for managing data transfers between computers including the steps of:	As indicated above, the two-way data communication device of this invention utilizes a client module to transmit a message including a resource locator selected by the user over the two-way data communication network to a server on a server computer on the computer network. For example, the computer network can be a corporate wide area network, a corporate local area network, the Internet, or any combination of computer networks. The server processes the message, i.e., executes the application addressed by the resource locator and transmits a response over the two-way data communication network to the two-way data communication device, which stores the response in a memory. The client module interprets the response and generates a user interface using information in the response. In one embodiment, the user interface includes at least one user data input option that is associated with a resource locator. Rossmann p. 4, lines 1-9
(a) creating a questionnaire at a first site in a first computer;	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a

	resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the **appropriate purchase order** and transmits the purchase order as a card deck to the client process in cellular telephone 100 over airnet network 150. The client process interprets the card deck and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines, 38-48 For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12
(b) tokenizing said questionnaire, thereby producing a tokenized questionnaire;	In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the appropriate purchase order and **transmits the purchase order as a card deck** to the client process in cellular telephone 100 over airnet network 150. **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105.

After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F.
Rossmann, p. 10, lines 38-48

In addition, the client process using the information transmitted from server computer 121, **i.e., the cards, generates a wide-variety of user interfaces as illustrated in Figures 2A to 2H.**
Rossmann p. 11, lines 15-16

Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression.
Rossmann p. 14, lines 55-58

Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network.
Rossmann p. 15, lines 2-7

For example, if the user of cellular telephone 700 requested a fax as in Figure 2F, the HTTP request identifies a common gateway interface application in CGI programs 761 that accepts as input data the telephone number and grabs the information to be faxed. The CGI application generates an e-mail transmission to the fax gateway. **Similarly, for a stock quote**, server 749, in response to the HTTP request, launches a common gateway interface application that sends out a stock query over Internet 140 to a stock quote service provider using the ticker tape symbol passed as input data by server 749 to the common gateway interface application. When the response to the stock query is received, the common
gateway interface application builds a PIDL deck that includes the data in the response to the stock query.

The interface presented in Table 7 for TIL manager module 1403 is

	designed with the assumption that TIL is a direct tokenization of PIDL as described in Appendix I. Rossmann p. 15, lines 56-57
(c) bringing a remote computer into electronic communication with said first computer;	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38
(d) transmitting said tokenized questionnaire to said remote computer;	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression. Rossmann p. 14, lines 55-58 Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a

	screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network. Rossmann p. 15, lines 2-7
(e) removing said remote computer from electronic communication with said first computer;	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38 After the transaction is logged, processing transfers to transmit result 1317. In transmit result 1317, ANT request processor 1204 returns the deck to client 702. After the deck is transmitted, ANT request processor 1204 is terminated. Rossmann p. 26, lines 5-6. Routine NM_lnit initializes network manager module 1402 and so is called before any other calls in network manager module 1402. Routine NM_ Terminate closes processing of network manager module 1402 and so is called after all other calls in network manager module 1402. Rossmann p. 28, lines 39-41

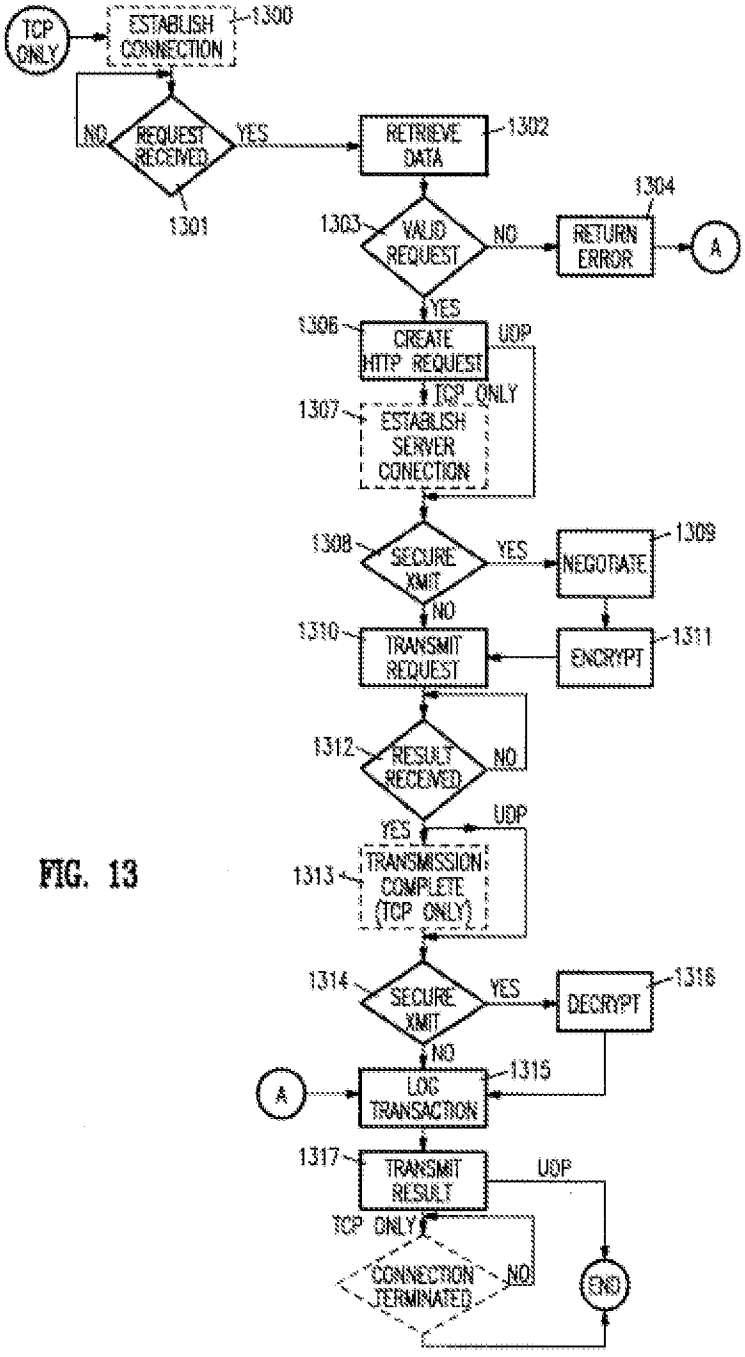


FIG. 13

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;	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12 The client process in cellular telephone 100 interprets the display card that includes image and text data and generates screen display 300 on display screen 105 (Fig. 3A). Screen display 300 includes a home key 301, and an info key 302. When the user selects home key 301, the user is returned to the home screen. Info key 302 functions in a manner similar to that described above for info key 205. When the user presses a predetermined key, the client process interprets the choice card and a second screen display 304 (Fig. 3B) is driven on display screen 105. Screen display 304 is a menu of the personal

	information that is stored on *server* computer 131 for use by the user of cellular telephone 100. Multi-display screen card indicator 203, e.g., the hand with a finger pointing down, illustrates to the user that the list has additional items that appear on the next screen display. Screen display 304 also indicates the number of E-mail messages, faxes, and voice messages waiting for the user. Rossmann p. 11, lines 43-52
(g) storing said at least one user response within said remote computer;	This invention allows for the first time two-way communications devices such as cellular telephones, two-way pagers, and telephones to become open application platforms which in turn empowers software developers to deliver value-added applications and services to any two-way communication device that incorporates the principles of this invention. This is a radical shift from the current situation where telephones and two-way pagers are closed, proprietary systems. Consequently, an even playing field is created for the market to invent new uses for two-way communication devices and for two-way communication networks. Any entity from corporations to individuals can make new applications available to the installed base of two-way data communication devices that include this invention without physical modification or addition to the two-way communication device. Years after purchase, a two-way communication device incorporating this invention will run all the applications which were developed since its purchase. Rossmann p. 3, lines 41-49 When the user selects the at least one user data input option, the client module interprets the selection and if required, appends any input data to the resource allocator associated with the at least one user data input option. The client module transmits a message including the resource locator with any appended input data to the server computer. Alternatively, the resource locator with any appended data can be addressed to another server computer, or can address an object stored in the two-way communication device. If the resource locator addresses an object on a server computer, the client module provides the message to the network interface module which in turn transmits the message over the two-way data communication network. Rossmann p. 4, lines 15-21 An important aspect of this invention is that the message includes all information necessary for the client module to generate the user interface and a particular user interface can be independent from other user interfaces. Unlike prior art systems that gave the user a predetermined menu from which to select items, or limited the user to an E-mail like format, according to the principles of this invention, the user interfaces and possible interactions available to the user are determined only by the

	applications that developers make available. The possible interactions and user interfaces for one application can be totally different and independent from the possible interactions and user interfaces of another application. Thus, a cellular telephone, two-way pager, and a telephone all truly become an open platform. Rossmann at p. 4, lines 29-35 In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18
(h) modifying said questionnaire with incremental changes at a second computer located at a second site;	Thus, the client module only interprets this information and interacts appropriately with the hardware of the two-way data communication device. Consequently, to update an application requires only **changes** on the server computer and not changes in each two-way data communication device that communicates with that server computer. This invention eliminates the usual requirement for distribution of application software, and application software updates to the end user of the two-way data communication device. Rossmann p. 4, lines 47-51
(i) placing said remote computer into electrical communication with said second computer;	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely

	below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38
(j) transmitting said incremental changes from said second computer to said remote computer;	This invention allows for the first time two-way communications devices such as cellular telephones, two-way pagers, and telephones to become open application platforms which in turn empowers software developers to deliver value-added applications and services to any two-way communication device that incorporates the principles of this invention. This is a radical shift from the current situation where telephones and two-way pagers are closed, proprietary systems. Consequently, an even playing field is created for the market to invent new uses for two-way communication devices and for two-way communication networks. Any entity from corporations to individuals can make new applications available to the installed base of two-way data communication devices that include this invention without physical modification or addition to the two-way communication device. Years after purchase, a two-way communication device incorporating this invention will run all the applications which were developed since its purchase. Rossmann p. 3, lines 41-49 For local services, like local message store, there are two basic approaches that can be used. First, local services are implemented in a CGI-like manner. Each local service has an entry point which is called with an argument list. A TIL deck is returned via the event manager. From that point on, the TIL deck is processed in the standard manner. This approach limits local services to the same constraints as remote services. A less restrictive approach is to allow the local service to field events instead of the standard event loop. The local service would construct TIL cards on-the-fly and feed them to user interface manager 1406. Note that the local service would need to cooperate with the standard event loop with regard to the history, the pushed card list, and any other state that is normally managed by the event loop. Table 4 is a

	listing of processes for the architecture for navigation manager module 1401. Rossmann p. 26, lines 37-44
(k) modifying said transmitted tokenized questionnaire in said remote computer with said incremental changes, thereby creating a modified tokenized questionnaire;	Thus, the client module only interprets this information and interacts appropriately with the hardware of the two-way data communication device. Consequently, to update an application requires only **changes** on the server computer and not changes in each two-way data communication device that communicates with that server computer. This invention eliminates the usual requirement for distribution of application software, and application software updates to the end user of the two-way data communication device. Rossmann p. 4, lines 47-51 For local services, like local message store, there are two basic approaches that can be used. First, local services are implemented in a CGI-like manner. Each local service has an entry point which is called with an argument list. A TIL deck is returned via the event manager. From that point on, the TIL deck is processed in the standard manner. This approach limits local services to the same constraints as remote services. A less restrictive approach is to allow the local service to field events instead of the standard event loop. The local service would construct TIL cards on-the-fly and feed them to user interface manager 1406. Note that the local service would need to cooperate with the standard event loop with regard to the history, the pushed card list, and any other state that is normally managed by the event loop. Table 4 is a listing of processes for the architecture for navigation manager module 1401. Rossmann p. 26, lines 37-44
(l) removing said remote computer from electronic communication with said second computer;	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38

	After the transaction is logged, processing transfers to transmit result 1317. In transmit result 1317, ANT request processor 1204 returns the deck to client 702. After the deck is transmitted, ANT request processor 1204 is terminated. Rossmann p. 26, lines 5-6. Routine NM_lnit initializes network manager module 1402 and so is called before any other calls in network manager module 1402. Routine NM_ Terminate closes processing of network manager module 1402 and so is called after all other calls in network manager module 1402. Rossmann p. 28, lines 39-41

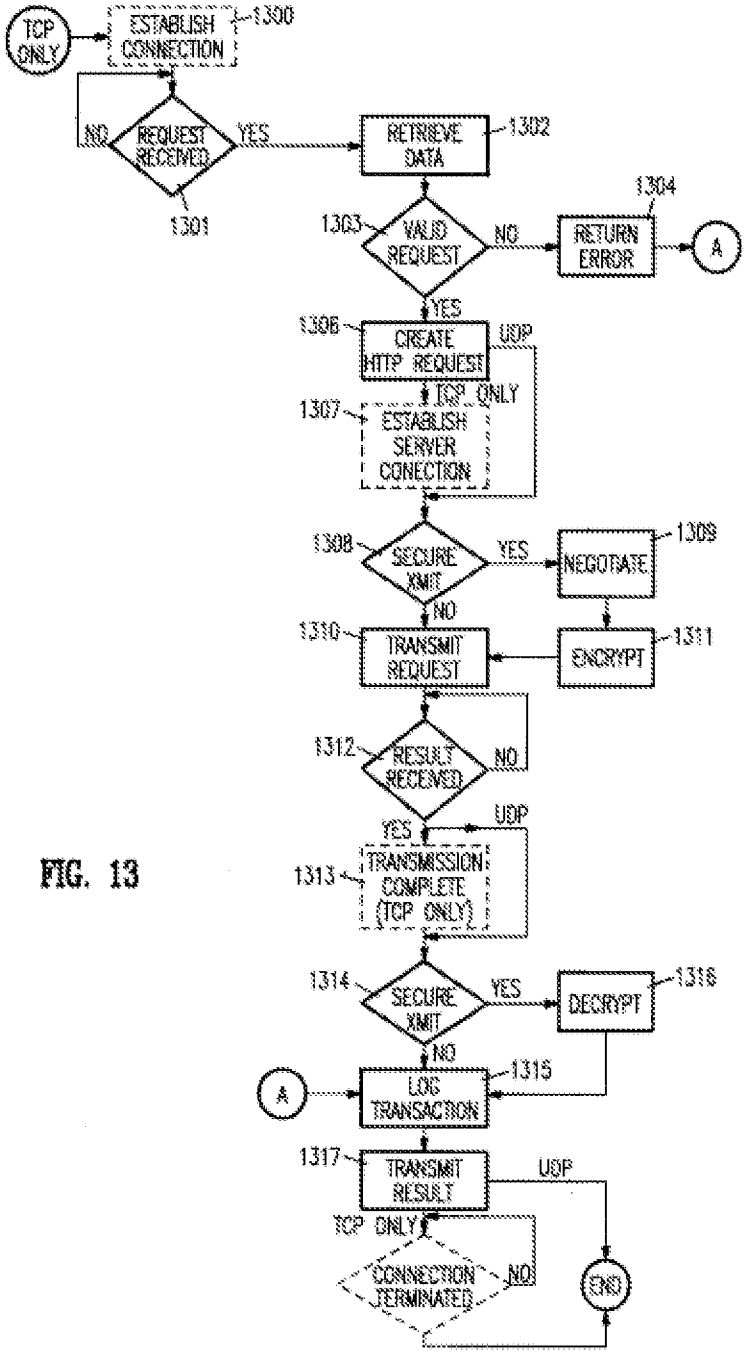


FIG. 13

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
(m) within said remote computer, using said modified tokenized questionnaire to obtain at least one additional user response;	When the user selects the at least one user data input option, the client module interprets the selection and if required, appends any input data to the resource allocator associated with the at least one user data input option. The client module transmits a message including the resource locator with any appended input data to the server computer. Alternatively, the resource locator with any appended data can be addressed to another server computer, or can address an object stored in the two-way communication device. If the resource locator addresses an object on a server computer, the client module provides the message to the network interface module which in turn transmits the message over the two-way data communication network. Rossmann p. 4, lines 15-21 An important aspect of this invention is that the message includes all information necessary for the client module to generate the user interface and a particular user interface can be independent from other user interfaces. Unlike prior art systems that gave the user a predetermined menu from which to select items, or limited the user to an E-mail like format, according to the principles of this invention, the user interfaces

	and possible interactions available to the user are determined only by the applications that developers make available. The possible interactions and user interfaces for one application can be totally different and independent from the possible interactions and user interfaces of another application. Thus, a cellular telephone, two-way pager, and a telephone all truly become an open platform. Rossmann at p. 4, lines 29-35 In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8
(n) placing said remote computer into electronic communication with a server;	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38

(o) transmitting said at least one user response to said server;	When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 The user scrolls the screen display line by line until screen display 305 is on display screen 105. Initially, the fourth item in the menu is not highlighted. In this example, the user presses the four key on the keypad of cellular telephone 100 to view the user's schedule. In response to the key press, the client module in cellular telephone 100 transmits a message, including a resource locator associated with the menu item selected by pressing the four key, to server computer 131 using data capable cellular telephone network 110 and corporate local area network 130. In response to the message, server computer 131 executes the application identified in the resource locator. Upon completion of the execution, server computer 131 transmits, over corporate local area network 130 and data capable cellular telephone network 110 to cellular telephone 100, a card deck that includes a choice card that describes the user's schedule for that day. Rossmann p. 11, line 53 – Rossmann p. 12, line 2 As indicated above, each interaction with the user of cellular telephone 700 is described by a deck or a series of decks. Logically, the user retrieves a terminal interaction language deck stored in a memory 716 of cellular telephone 700 after receipt from computer 743 over CDPD network 710. The user reviews the information displayed by cards in the deck and makes choices and/or enters requested information and then requests another deck, as described above with respect to Figures 2A to 2H, for example. Rossmann p. 15, lines 23-27
(p) transmitting said at least one additional user response to said server;	When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 The user scrolls the screen display line by line until screen display 305 is on display screen 105. Initially, the fourth item in the menu is not

	highlighted. In this example, the user presses the four key on the keypad of cellular telephone 100 to view the user's schedule. In response to the key press, the client module in cellular telephone 100 transmits a message, including a resource locator associated with the menu item selected by pressing the four key, to server computer 131 using data capable cellular telephone network 110 and corporate local area network 130. In response to the message, server computer 131 executes the application identified in the resource locator. Upon completion of the execution, server computer 131 transmits, over corporate local area network 130 and data capable cellular telephone network 110 to cellular telephone 100, a card deck that includes a choice card that describes the user's schedule for that day. Rossmann p. 11, line 53 – Rossmann p. 12, line 2 As indicated above, each interaction with the user of cellular telephone 700 is described by a deck or a series of decks. Logically, the user retrieves a terminal interaction language deck stored in a memory 716 of cellular telephone 700 after receipt from computer 743 over CDPD network 710. The user reviews the information displayed by cards in the deck and makes choices and/or enters requested information and then requests another deck, as described above with respect to Figures 2A to 2H, for example. Rossmann p. 15, lines 23-27
(q) storing said transmitted at least one user response and said at least one additional user response at said server;	The server processes the message, i.e., executes the application addressed by the resource locator and transmits a response over the two-way data communication network to the two-way data communication device, which **stores** the response in a memory. Rossmann p. 4, lines 5-7 When the user presses a predetermined key, the client process interprets the choice card and a second screen display 304 (Fig. 3B) is driven on display screen 105. Screen display 304 is a menu of the personal information that is stored on *server* computer 131 for use by the user of cellular telephone 100. Multi-display screen card indicator 203, e.g., the hand with a finger pointing down, illustrates to the user that the list has additional items that appear on the next screen display. Screen display 304 also indicates the number of E-mail messages, faxes, and *voice* messages waiting for the user. Rossmann p. 11, lines 47-52 Computer 743 may contain **stored** static telephone interaction description language decks. Computer 743 also generates telephone interaction description language decks in response to data from, or

	choices made by, the user of cellular telephone 700. Rossmann p. 15, lines 13-15
(r) preparing a report using any of said at least one user response and said at least one additional user response; and,	When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 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 The user scrolls the screen display line by line until screen display 305 is on display screen 105. Initially, the fourth item in the menu is not highlighted. In this example, the user presses the four key on the keypad of cellular telephone 100 to view the user's schedule. In response to the key press, the client module in cellular telephone 100 transmits a message, including a resource locator associated with the menu item selected by pressing the four key, to server computer 131 using data capable cellular telephone network 110 and corporate local area network 130. In response to the message, server computer 131 executes the application identified in the resource locator. Upon completion of the execution, server computer 131 transmits, over corporate local area network 130 and data capable cellular telephone network 110 to cellular telephone 100, a card deck that includes a choice card that describes the user's schedule for that day. Rossmann p. 11, line 53 – Rossmann p. 12, line 2 As indicated above, each interaction with the user of cellular telephone 700 is described by a deck or a series of decks. Logically, the user retrieves a terminal interaction language deck stored in a memory 716 of cellular telephone 700 after receipt from computer 743 over CDPD network 710. The user reviews the information displayed by cards in the deck and makes choices and/or enters requested information and then requests another deck, as described above with respect to Figures 2A to

	2H, for example. Rossmann p. 15, lines 23-27
(s) displaying at least a portion of said report on a visually perceptible medium;	When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 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 The user scrolls the screen display line by line until screen display 305 is on display screen 105. Initially, the fourth item in the menu is not highlighted. In this example, the user presses the four key on the keypad of cellular telephone 100 to view the user's schedule. In response to the key press, the client module in cellular telephone 100 transmits a message, including a resource locator associated with the menu item selected by pressing the four key, to server computer 131 using data capable cellular telephone network 110 and corporate local area network 130. In response to the message, server computer 131 executes the application identified in the resource locator. Upon completion of the execution, server computer 131 transmits, over corporate local area network 130 and data capable cellular telephone network 110 to cellular telephone 100, a card deck that includes a choice card that describes the user's schedule for that day. Rossmann p. 11, line 53 – Rossmann p. 12, line 2 As indicated above, each interaction with the user of cellular telephone 700 is described by a deck or a series of decks. Logically, the user retrieves a terminal interaction language deck stored in a memory 716 of cellular telephone 700 after receipt from computer 743 over CDPD network 710. The user reviews the information displayed by cards in the deck and makes choices and/or enters requested information and then requests another deck, as described above with respect to Figures 2A to 2H, for example.

	Rossmann p. 15, lines 23-27
(t) performing at least steps (d)-(p) using at least two different remote computing device types using the same tokens.	Each wireless communication device 100 that includes this invention can communicate over airnet network 150 with any server computer 121, 131, and 141 on airnet network 150 that includes at least one application that communicates and interacts with the processes of this invention that are included within device 100. Thus, device 100 can access information on the computer network and provide information to the computer network. Similarly, a two-way pager 101, and a telephone 102 with a modem 103, that each include this invention, can communicate over airnet network 150 with any of server computers 121, 131, and 141 that includes at least one application that communicates and interacts with the processes of this invention that are included within devices 101 and 102. Rossmann p. 6, lines 38-44
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.	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37
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.	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37
CLAIM 11	

11. A method for collecting survey data from a user comprising the steps of:	As indicated above, the two-way data communication device of this invention utilizes a client module to transmit a message including a resource locator selected by the user over the two-way data communication network to a server on a server computer on the computer network. For example, the computer network can be a corporate wide area network, a corporate local area network, the Internet, or any combination of computer networks. The server processes the message, i.e., executes the application addressed by the resource locator and transmits a response over the two-way data communication network to the two-way data communication device, which stores the response in a memory. The client module interprets the response and generates a user interface using information in the response. In one embodiment, the user interface includes at least one user data input option that is associated with a resource locator. Rossmann p. 4, lines 1-9
(a) creating a questionnaire comprising a series of questions;	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the **appropriate purchase order** and transmits the purchase order as a card deck to the client process in cellular telephone 100 over airnet network 150. The client process interprets the card deck and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display

	209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines, 38-48 For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12
(b) tokenizing said questionnaire; thereby producing a plurality of tokens representing said questionnaire;	In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the appropriate purchase order and **transmits the purchase order as a card deck** to the client process in cellular telephone 100 over airnet network 150. **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines 38-48 In addition, the client process using the information transmitted from server computer 121, **i.e., the cards, generates a wide-variety of user**

	interfaces as illustrated in Figures 2A to 2H. Rossmann p. 11, lines 15-16 Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression. Rossmann p. 14, lines 55-58 Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network. Rossmann p. 15, lines 2-7 For example, if the user of cellular telephone 700 requested a fax as in Figure 2F, the HTTP request identifies a common gateway interface application in CGI programs 761 that accepts as input data the telephone number and grabs the information to be faxed. The CGI application generates an e-mail transmission to the fax gateway. **Similarly, for a stock quote**, server 749, in response to the HTTP request, launches a common gateway interface application that sends out a stock query over Internet 140 to a stock quote service provider using the ticker tape symbol passed as input data by server 749 to the common gateway interface application. When the response to the stock query is received, the common gateway interface application builds a PIDL deck that includes the data in the response to the stock query. The interface presented in Table 7 for TIL manager module 1403 is designed with the assumption that TIL is a direct tokenization of PIDL as described in Appendix I. Rossmann p. 15, lines 56-57
(c) storing said plurality of tokens on a computer readable medium on a first	In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator

computer;	to server computer 121. Server computer 121, in response to the message, retrieves the appropriate purchase order and **transmits the purchase order as a card deck** to the client process in cellular telephone 100 over airnet network 150. **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines 38-48 In addition, the client process using the information transmitted from server computer 121, **i.e., the cards, generates a wide-variety of user interfaces as illustrated in Figures 2A to 2H.** Rossmann p. 11, lines 15-16 Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression. Rossmann p. 14, lines 55-58 Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network. Rossmann p. 15, lines 2-7
(d) placing a handheld remote computing device into electronic communication with	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer

said first computer;	networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38
(e) transmitting said plurality of tokens to said handheld remote computing device;	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression. Rossmann p. 14, lines 55-58 Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network. Rossmann p. 15, lines 2-7

(f) taking said handheld remote computing device out of electronic communication with said first computer;	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38 After the transaction is logged, processing transfers to transmit result 1317. In transmit result 1317, ANT request processor 1204 returns the deck to client 702. After the deck is transmitted, ANT request processor 1204 is terminated. Rossmann p. 26, lines 5-6. Routine NM_lnit initializes network manager module 1402 and so is called before any other calls in network manager module 1402. Routine NM_ Terminate closes processing of network manager module 1402 and so is called after all other calls in network manager module 1402. Rossmann p. 28, lines 39-41

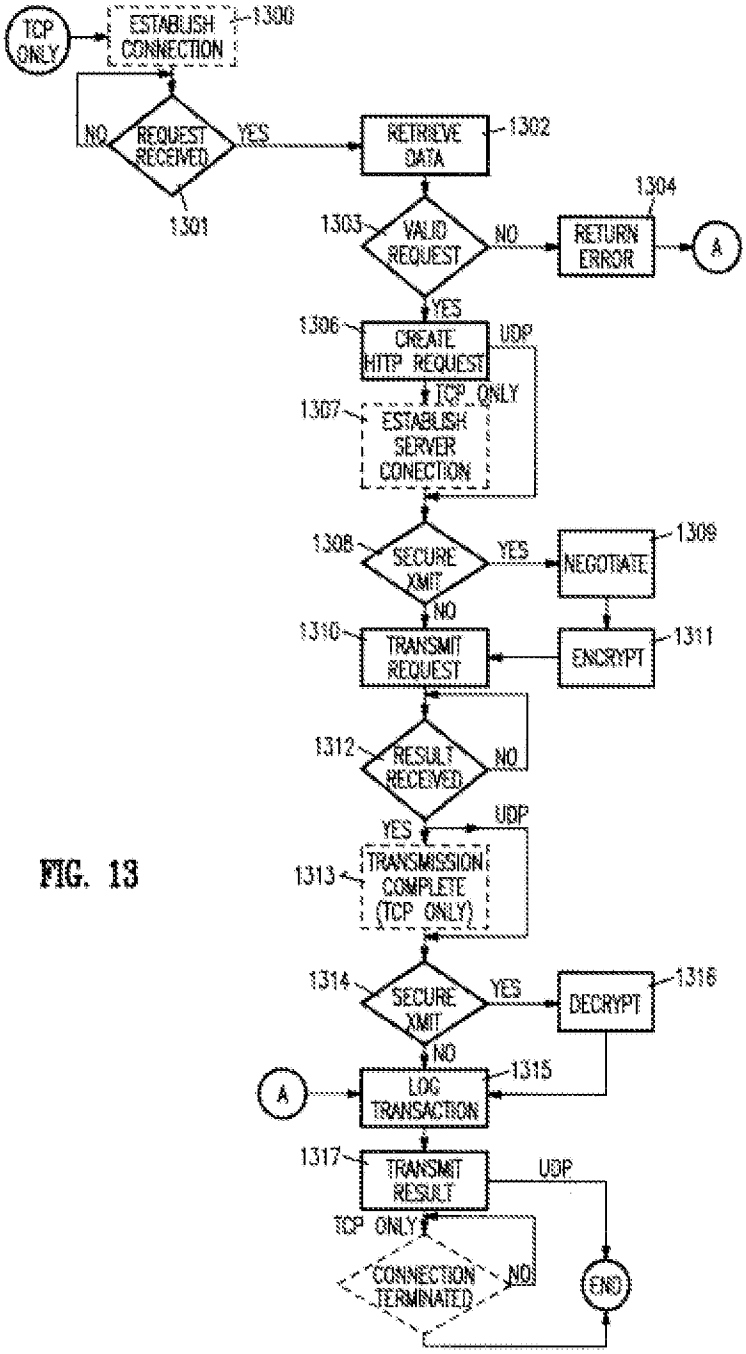


FIG. 13

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,	After the transaction is logged, processing transfers to transmit result 1317. In transmit result 1317, ANT request processor 1204 returns the deck to client 702. After the deck is transmitted, ANT request processor 1204 is terminated. Rossmann p. 26, lines 5-6. Routine NM_lnit initializes network manager module 1402 and so is called before any other calls in network manager module 1402. Routine NM_ Terminate closes processing of network manager module 1402 and so is called after all other calls in network manager module 1402. Rossmann p. 28, lines 39-41
(g1) executing at least a portion of said plurality of tokens representing said questionnaire on said handheld remote computing device to	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller.

collect a response from a user, and,	Rossmann p. 15, lines 8-12 The client process in cellular telephone 100 interprets the display card that includes image and text data and generates screen display 300 on display screen 105 (Fig. 3A). Screen display 300 includes a home key 301, and an info key 302. When the user selects home key 301 , the user is returned to the home screen. Info key 302 functions in a manner similar to that described above for info key 205. When the user presses a predetermined key, the client process interprets the choice card and a second screen display 304 (Fig. 3B) is driven on display screen 105. Screen display 304 is a menu of the personal information that is stored on *server* computer 131 for use by the user of cellular telephone 100. Multi-display screen card indicator 203, e.g., the hand with a finger pointing down, illustrates to the user that the list has additional items that appear on the next screen display. Screen display 304 also indicates the number of E-mail messages, faxes, and *voice* messages waiting for the user. Rossmann p. 11, lines 43-52
(g2) storing within said remote computing device said response from the user;	The server processes the message, i.e., executes the application addressed by the resource locator and transmits a response over the two-way data communication network to the two-way data communication device, which **stores** the response in a memory. Rossmann p. 4, lines 5-7 When the user presses a predetermined key, the client process interprets the choice card and a second screen display 304 (Fig. 3B) is driven on display screen 105. Screen display 304 is a menu of the personal information that is stored on *server* computer 131 for use by the user of cellular telephone 100. Multi-display screen card indicator 203, e.g., the hand with a finger pointing down, illustrates to the user that the list has additional items that appear on the next screen display. Screen display 304 also indicates the number of E-mail messages, faxes, and *voice* messages waiting for the user. Rossmann p. 11, lines 47-52 Computer 743 may contain **stored** static telephone interaction description language decks. Computer 743 also generates telephone interaction description language decks in response to data from, or choices made by, the user of cellular telephone 700. Rossmann p. 15, lines 13-15
(h) placing said handheld remote computing device into	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices

electronic communication with a second computer;	100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38
(i) transmitting at least a portion of said response stored within said handheld remote computing device to said second computer; and,	When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 The user scrolls the screen display line by line until screen display 305 is on display screen 105. Initially, the fourth item in the menu is not highlighted. In this example, the user presses the four key on the keypad of cellular telephone 100 to view the user's schedule. In response to the key press, the client module in cellular telephone 100 transmits a message, including a resource locator associated with the menu item selected by pressing the four key, to server computer 131 using data capable cellular telephone network 110 and corporate local area network 130. In response to the message, server computer 131 executes the application identified in the resource locator. Upon completion of the execution, server computer 131 transmits, over corporate local area network 130 and data capable cellular telephone network 110 to cellular telephone 100, a card deck that includes a choice card that describes the user's schedule for that day. Rossmann p. 11, line 53 – Rossmann p. 12, line 2 As indicated above, each interaction with the user of cellular telephone 700 is described by a deck or a series of decks. Logically, the user retrieves a terminal interaction language deck stored in a memory 716 of

	cellular telephone 700 after receipt from computer 743 over CDPD network 710. The user reviews the information displayed by cards in the deck and makes choices and/or enters requested information and then requests another deck, as described above with respect to Figures 2A to 2H, for example. Rossmann p. 15, lines 23-27
(j) forming a visually perceptible report from any of said at least a portion of said response so transmitted.	When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 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 The user scrolls the screen display line by line until screen display 305 is on display screen 105. Initially, the fourth item in the menu is not highlighted. In this example, the user presses the four key on the keypad of cellular telephone 100 to view the user's schedule. In response to the key press, the client module in cellular telephone 100 transmits a message, including a resource locator associated with the menu item selected by pressing the four key, to server computer 131 using data capable cellular telephone network 110 and corporate local area network 130. In response to the message, server computer 131 executes the application identified in the resource locator. Upon completion of the execution, server computer 131 transmits, over corporate local area network 130 and data capable cellular telephone network 110 to cellular telephone 100, a card deck that includes a choice card that describes the user's schedule for that day. Rossmann p. 11, line 53 – Rossmann p. 12, line 2 As indicated above, each interaction with the user of cellular telephone 700 is described by a deck or a series of decks. Logically, the user retrieves a terminal interaction language deck stored in a memory 716 of cellular telephone 700 after receipt from computer 743 over CDPD

	network 710. The user reviews the information displayed by cards in the deck and makes choices and/or enters requested information and then requests another deck, as described above with respect to Figures 2A to 2H, for example. Rossmann p. 15, lines 23-27
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.	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37
CLAIM 14	
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;	Thus, the client module only interprets this information and interacts appropriately with the hardware of the two-way data communication device. Consequently, to update an application requires only **changes** on the server computer and not changes in each two-way data communication device that communicates with that server computer. This invention eliminates the usual requirement for distribution of application software, and application software updates to the end user of the two-way data communication device. Rossmann p. 4, lines 47-51

(l) tokenizing said at least one incremental change to said questionnaire;	For local services, like local message store, there are two basic approaches that can be used. First, local services are implemented in a CGI-like manner. Each local service has an entry point which is called with an argument list. A TIL deck is returned via the event manager. From that point on, the TIL deck is processed in the standard manner. This approach limits local services to the same constraints as remote services. A less restrictive approach is to allow the local service to field events instead of the standard event loop. The local service would construct TIL cards on-the-fly and feed them to user interface manager 1406. Note that the local service would need to cooperate with the standard event loop with regard to the history, the pushed card list, and any other state that is normally managed by the event loop. Table 4 is a listing of processes for the architecture for navigation manager module 1401. Rossmann p. 26, lines 37-44
(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,	For local services, like local message store, there are two basic approaches that can be used. First, local services are implemented in a CGI-like manner. Each local service has an entry point which is called with an argument list. A TIL deck is returned via the event manager. From that point on, the TIL deck is processed in the standard manner. This approach limits local services to the same constraints as remote services. A less restrictive approach is to allow the local service to field events instead of the standard event loop. The local service would construct TIL cards on-the-fly and feed them to user interface manager 1406. Note that the local service would need to cooperate with the standard event loop with regard to the history, the pushed card list, and any other state that is normally managed by the event loop. Table 4 is a listing of processes for the architecture for navigation manager module 1401. Rossmann p. 26, lines 37-44
(n) incorporating said transmitted tokens into said questionnaire at said remote computing device, thereby incrementally changing said questionnaire.	Thus, the client module only interprets this information and interacts appropriately with the hardware of the two-way data communication device. Consequently, to update an application requires only **changes** on the server computer and not changes in each two-way data communication device that communicates with that server computer. This invention eliminates the usual requirement for distribution of application software, and application software updates to the end user of the two-way data communication device. Rossmann p. 4, lines 47-51 For local services, like local message store, there are two basic approaches that can be used. First, local services are implemented in a CGI-like manner. Each local service has an entry point which is called with an argument list. A TIL deck is returned via the event manager.

	From that point on, the TIL deck is processed in the standard manner. This approach limits local services to the same constraints as remote services. A less restrictive approach is to allow the local service to field events instead of the standard event loop. The local service would construct TIL cards on-the-fly and feed them to user interface manager 1406. Note that the local service would need to cooperate with the standard event loop with regard to the history, the pushed card list, and any other state that is normally managed by the event loop. Table 4 is a listing of processes for the architecture for navigation manager module 1401. Rossmann p. 26, lines 37-434

B. CLAIM 4 OF THE '816 PATENT ARE RENDERED OBVIOUS BY ROSSMANN IN VIEW OF RAPPAPORT, BOWEN, AND THE KNOWLEDGE OF A PERSON OF ORDINARY SKILL IN THE ART

Please see the below claim chart that applies the teachings of Rossmann in view of Rappaport and Bowen to claim 4 of the '816 patent.

Reasons to Combine:

A person of ordinary skill in the art would recognize that Rossmann teaches that a mobile device that connects to a server and transmits information between the two devices. *See* Rossmann p. 6, lines 31-37 and 14, lines 35-38. It would have been obvious to combine Rossmann 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. Further, Bowen teaches that the hierarchical structure of a questionnaire can be beneficial for recording user responses. Bowen at Abstract.

Brief Description of Application within Charts Below:

Rossmann teaches a card deck wherein each of the cards is a single operation and is a representation of a question in a questionnaire, this is tokenization within the broadest reasonable interpretation of the claims of the '816 patent. Rossmann, p. 10, lines 38-48, 11, lines 15-16, 14, lines 55-58, 15, lines 2-7, and 15, lines 56-57. These tokens, i.e., cards, can be communicated to a computer from a server and from a computer to a server through any known two-way data communication network. Rossmann p. 15, lines 8-12, p. 14, lines 35-38, p. 26, lines 5-6, and p. 28, lines 39-41. The method described by Rossmann can be supplemented by the addition of Rappaport, which teaches that mobile devices can terminate and reestablish connections.

Rappaport at 7:44-63. Accordingly, the cards and card decks can be communicated between a mobile device and a server. Further, Bowen teaches that the hierarchical structure of a questionnaire can be beneficial for recording user responses. Bowen at Abstract.

As will be appreciated with reference to the citations contained in the charts below, each of the claim limitations is explicitly taught by the combination of Rossmann, Rappaport, and Bowen. Accordingly, the rejections should be adopted.

CLAIM 4	Teachings from Rossmann in view of Rappaport and Bowen
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:	In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8 In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the appropriate purchase order and **transmits the purchase order as a card deck** to the client process in cellular telephone 100 over airnet network 150. **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key

	sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann, p. 10, lines 38-48 In addition, the client process using the information transmitted from server computer 121, **i.e., the cards, generates a wide-variety of user interfaces as illustrated in Figures 2A to 2H.** Rossmann p. 11, lines 15-16 Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression. Rossmann p. 14, lines 55-58 Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network. Rossmann p. 15, lines 2-7 For example, if the user of cellular telephone 700 requested a fax as in Figure 2F, the HTTP request identifies a common gateway interface application in CGI programs 761 that accepts as input data the telephone number and grabs the information to be faxed. The CGI application generates an e-mail transmission to the fax gateway. **Similarly, for a stock quote**, server 749, in response to the HTTP request, launches a common gateway interface application that sends out a stock query over Internet 140 to a stock quote service provider using the ticker tape symbol passed as input data by server 749 to the common gateway interface application. When the response to the stock query is received, the common gateway interface application builds a PIDL deck that includes the data in the response to the stock query. The interface presented in Table 7 for TIL manager module 1403 is designed with the assumption that TIL is a direct tokenization of PIDL as described in Appendix I.

	Rossmann p. 15, lines 56-57
(i) assigning at least one token to each question of said series of questions;	Specifically, the application accessed on server computer 121 generates the card deck and so in turn defines each of the various user interfaces. Each user interface permits the user to identify a particular selection. Each particular selection could result in generation of a different user interface with different selections. Thus, the user interfaces are limited only by the applications accessible to the two-way data communication device. Rossmann p. 11, lines 21-24 In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8
(ii) assigning at least one token to each response called for in said series of questions to identify the type of response required; and	The home key is associated with a pointer, that in one embodiment is a resource locator, and the card addressed by the pointer is displayed by the client process when the home key is selected by the user. Specifically, if the pointer is to a card in the current deck, the client process simply displays that card. If the pointer is to other than a card in the current deck, the client process in cellular telephone 100 retrieves the deck containing the card at the location identified by the pointer. The location could be, for example, either a memory in cellular telephone 100, or a memory in computer 121. Rossmann p. 9, line 59-p. 10, line 5 Specifically, the application accessed on server computer 121 generates the card deck and so in turn defines each of the various user interfaces. Each user interface permits the user to identify a particular selection. Each particular selection could result in generation of a different user interface with different selections. Thus, the user interfaces are limited only by the applications accessible to the two-way data communication device.

	Rossmann p. 11, lines 21-24
(iii) assigning at least one token to each branch in said questionnaire to identify the required program control associated with said branch.	The client process in cellular telephone 100 interprets the first card in the card deck from computer 141 and generates screen display 400 (Fig. 4A). When the user presses a predetermined key, cellular telephone 100 displays screen display 401 (Fig. 4B). Screen display 401 provides the user with a series of choices that group services alphabetically. When the user depresses the seven key on the keypad of cellular telephone 100, cellular telephone 100 displays a list of the services that have letters P, R, or S as the first letter in the service name. In this embodiment, screen displays 401 and 402 are a single card, e.g., a single screen. Each of the various services associated with a key has an index and when a particular choice is made by the user, the choice defines an index. The client process then displays all of the services with the index that corresponds to the index defined by the user's choice. In screen display 402, the user is given a series of choices of services that are available to the user under tab seven. Initially, item three in screen display 402 is not highlighted. In this example, the user depresses the three key on the keypad of cellular telephone 100 to select the stock quotes and item three in screen display 402 is highlighted. In response to this selection, cellular telephone 100 transmits a request for a stock quote, i.e., a message including a resource locator, over cellular telephone network 100 and internet 140 to service provider 141. In response to the request, service provider computer 141 executes the application addressed by the resource locator. The application retrieves a card deck that, in turn is transmitted to cellular telephone 100. The card deck includes a display card and an entry card. Upon receiving the card deck, the client process in cellular telephone 100 interprets the display card and generates screen display 403 (Fig. 40). When the user depresses a predetermined key, entry screen display 406 (Fig. 4E) is generated on display screen 105 of cellular telephone 100. Initially, the box with letters SUNW in screen display 406 is empty. The letters SUNW are entered in the box by the user to indicate the ticker symbol of the stock for which the user wants information. After the user has entered the stock ticker symbol, the user presses the predetermined key to indicate that the entry is complete. In response to the entry by the user, the client module appends the stock ticker symbol to the resource locator and transmits the resource locator

	to service provider computer 141 which, in turn, executes an application addressed by the resource locator to retrieve the latest stock market information for the stock ticker symbol. Service provider 141 uses the retrieved information to generate a card deck that contains the information and then transmits the card deck to cellular telephone 100. Rossmann p. 12, lines 25-52 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. Bowen col. 1, line 46- col. 2, line 2

C. CLAIMS 1 -14 OF THE '816 PATENT ARE RENDERED OBVIOUS BY ROSSMANN IN VIEW OF FALLS AND THE KNOWLEDGE OF A PERSON OF ORDINARY SKILL IN THE ART

Please see the below claim chart that applies the teachings of Rossmann in view of Falls to claims 1-14 of the '816 patent.

Reasons to Combine:

A person of ordinary skill in the art would recognize that Rossmann teaches that a mobile device that connects to a server and transmits information between the two devices. *See* Rossmann p. 6, lines 31-37 and 14, lines 35-38. It would have been obvious to combine Rossmann with Falls 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 Falls teaches

a method of synchronization upon reconnection of a mobile device to the network and server. *See* Falls at Abstract.

Brief Description of Application within Charts Below:

Rossmann teaches a card deck wherein each of the cards is a single operation and is a representation of a question in a questionnaire, this is tokenization within the broadest reasonable interpretation of the claims of the '816 patent. Rossmann, p. 10, lines 38-48, 11, lines 15-16, 14, lines 55-58, 15, lines 2-7, and 15, lines 56-57. These tokens, i.e., cards, can be communicated to a computer from a server and from a computer to a server through any known two-way data communication network. Rossmann p. 15, lines 8-12, p. 14, lines 35-38, p. 26, lines 5-6, and p. 28, lines 39-41. The method described by Rossmann can be supplemented by the addition of Falls, which teaches that mobile devices can terminate and reestablish connections. Falls at 3:16-35, 16:24-29, and 7:16-21. Accordingly, the cards and card decks can be communicated between a mobile device and a server.

As will be appreciated with reference to the citations contained in the charts below, each of the claim limitations is explicitly taught by the combination of Rossmann and Falls. Accordingly, the rejections should be adopted.

CLAIM 1	**Rossmann in view of Falls**
1. A method for managing data including the steps of:	As indicated above, the two-way data communication device of this invention utilizes a client module to transmit a message including a resource locator selected by the user over the two-way data communication network to a server on a server computer on the computer network. For example, the computer network can be a corporate wide area network, a corporate local area network, the Internet, or any combination of computer networks. The server processes the message, i.e., executes the application addressed by the resource locator and transmits a response over the two-way data communication network to the two-way data communication device, which stores the response in a memory. The client module interprets the response and generates a user interface using information in the response. In one embodiment, the user interface includes at least one user data input option that is associated with a resource locator. Rossmann p. 4, lines 1-9
(a) creating a questionnaire comprising a series of	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen

questions;	display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the **appropriate purchase order** and transmits the purchase order as a card deck to the client process in cellular telephone 100 over airnet network 150. The client process interprets the card deck and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines, 38-48 For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12

(b) tokenizing said questionnaire; thereby producing a plurality of tokens representing said questionnaire;	In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the appropriate purchase order and **transmits the purchase order as a card deck** to the client process in cellular telephone 100 over airnet network 150. **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann, p. 10, lines 38-48 In addition, the client process using the information transmitted from server computer 121, **i.e., the cards, generates a wide-variety of user interfaces as illustrated in Figures 2A to 2H.** Rossmann p. 11, lines 15-16 Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression. Rossmann p. 14, lines 55-58 Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network.

	Rossmann p. 15, lines 2-7 For example, if the user of cellular telephone 700 requested a fax as in Figure 2F, the HTTP request identifies a common gateway interface application in CGI programs 761 that accepts as input data the telephone number and grabs the information to be faxed. The CGI application generates an e-mail transmission to the fax gateway. **Similarly, for a stock quote**, server 749, in response to the HTTP request, launches a common gateway interface application that sends out a stock query over Internet 140 to a stock quote service provider using the ticker tape symbol passed as input data by server 749 to the common gateway interface application. When the response to the stock query is received, the common gateway interface application builds a PIDL deck that includes the data in the response to the stock query. The interface presented in Table 7 for TIL manager module 1403 is designed with the assumption that TIL is a direct tokenization of PIDL as described in Appendix I. Rossmann p. 15, lines 56-57
(c) establishing a first wireless modem or wireless LAN network connection with a remote computing device;	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38 A method and apparatus are disclosed for synchronizing transactions in a disconnectable network. Each transaction includes operations that were performed on a database replica on one computer while that computer was disconnected from another computer and hence from that other

	computer's replica. Transaction synchronization, which occurs after the computers are reconnected, transfers information from each computer to the other computer and applies updates to both replicas as appropriate. Transaction logs and clash handling tools may be used with the invention. Falls at Abstract The present invention provides a system and method which facilitate disconnected mobile computing in several ways. Prior to disconnection, the invention allows network administrators or users to readily select data that should be copied from a network to a mobile computer by simply identifying one or more target database subtrees. During disconnected operation of the mobile computer, the invention presents the user with a "virtual network" environment that is consistent in use and appearance with the selected portion of the actual network. Those of skill in the art will appreciate that other remote procedure call mechanisms may also be employed according to the present invention. Suitable network connections 52 may be established using packet-based, serial, internet compatible, local area, metropolitan area, wide area, and wireless network transmission systems and methods. Falls at 13:60-65 A merge process according to the present invention includes merging location sets when disconnected disconnectable computers are first connected or reconnected. For instance, merging location sets normally occurs when a computer new to the network starts up and merges into an existing location set. Falls at 16:24-29

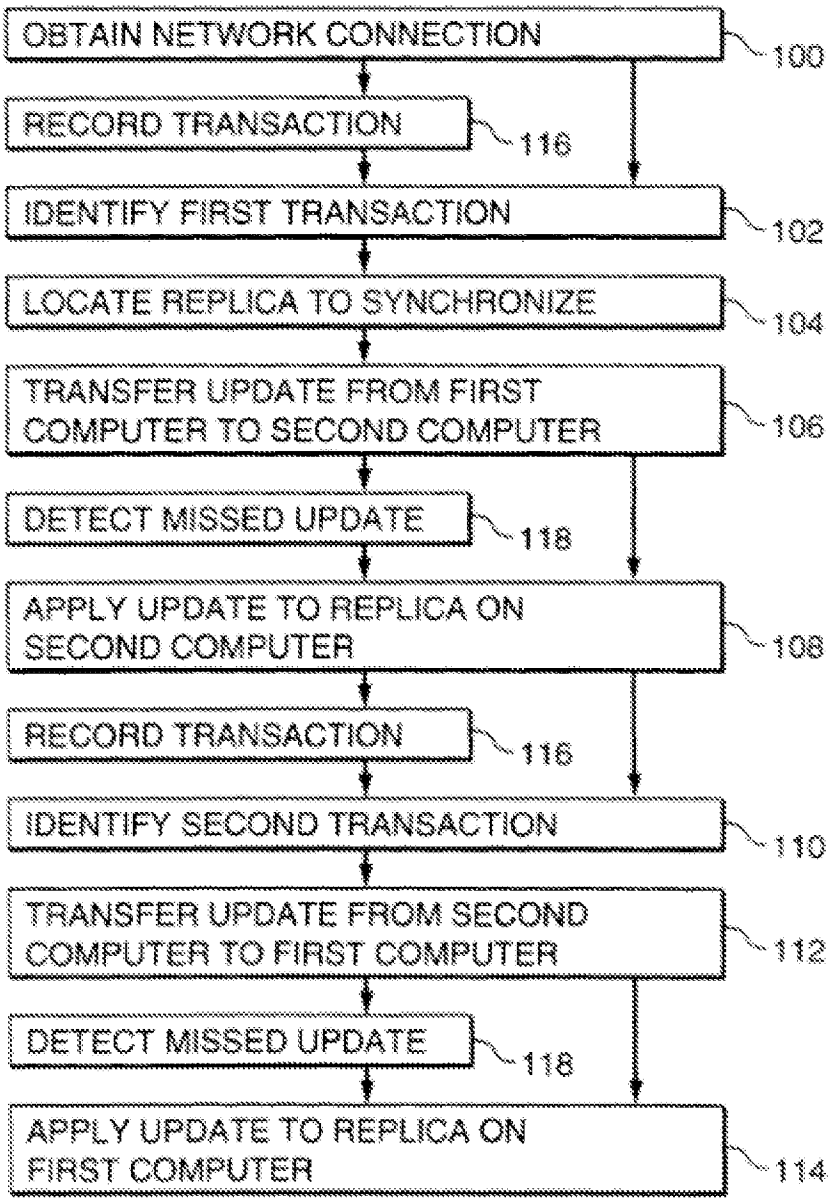


FIG. 4

(d) transmitting said plurality of tokens to a remote computing device via said first wireless modem or wireless LAN network connection;	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8

Preferably, each data type is compressed to facilitate optimal transfer over the two-way data communication network. For example, the verbs in the telephone interaction description language are compressed using a binary **tokenization**. Graphics are compressed using run length limited compression and text is compressed using anyone of the well-known techniques for text compression. Rossmann p. 14, lines 55-58

Instructions in the telephone interaction description language and in the terminal interaction language are grouped into **a deck and a card**. Each deck includes one or more cards. A card includes the information, i.e., a set of telephone interaction description language, required to generate a screen. As indicated above, a screen can be larger than the 5 number of lines in a display screen. Other equivalent terms for a card include a page and an atomic interaction. Thus, a card deck is simply a group of screens. The number of cards in a card deck is selected to facilitate efficient use of the resources in the two-way data communication device and in the airnet network.
Rossmann p. 15, lines 2-7

A method and apparatus are disclosed for synchronizing transactions in a disconnectable network. Each transaction includes operations that were performed on a database replica on one computer while that computer was disconnected from another computer and hence from that other computer's replica. Transaction synchronization, which occurs after the computers are reconnected, transfers information from each computer to the other computer and applies updates to both replicas as appropriate. Transaction logs and clash handling tools may be used with the invention.
Falls at Abstract

The present invention provides a system and method which facilitate disconnected mobile computing in several ways. Prior to disconnection, the invention allows network administrators or users to readily select data that should be copied from a network to a mobile computer by simply identifying one or more target database subtrees. During disconnected operation of the mobile computer, the invention presents the user with a "virtual network" environment that is consistent in use and appearance with the selected portion of the actual network.

Finally, upon reconnection of the mobile computer to the network, the invention synchronizes operations performed on the mobile computer during the disconnected interval with operations performed on the network during that interval. Synchronization is both substantially automatic and transactional, so minimal user intervention is needed and inconsistent internal states are avoided. Moreover, synchronization does

	not routinely discard any of the changes made on either the network or the mobile computer. Falls at 3:16-35 Each computer's replica manager communicates with the device controller of that computer and with the network link. Each replica manager also communicates with a database manager on its computer. The database manager can send database transactions to the device controller only through the replica manager, allowing the replica managers to log transactions and to synchronize the transactions after the network connection is re-established. Falls at 4:7-14 Merging occurs when two replicas 56 are resynchronized after the computers 28 on which the replicas 56 reside are reconnected following a period of disconnection. Falls at 16:35-37 In summary the present invention provides a system and method for properly synchronizing transactions when a disconnectable computer 28 is reconnected to the network 10. The invention is not limited to file system operations but can instead be extended to support a variety of database objects by using the schema 84, object distributor 82, object processor 86, and other modules. Clash handling means may be used to identify potentially conflicting database changes and allow their resolution by either automatic or manual means. Clash handling and retries also make locks optional. Falls at 37:11-18
(e) terminating said first wireless modem or wireless LAN network connection with said remote computing device;	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12 For example other two-way data communication networks for cellular telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38

	After the transaction is logged, processing transfers to transmit result 1317. In transmit result 1317, ANT request processor 1204 returns the deck to client 702. After the deck is transmitted, ANT request processor 1204 is terminated. Rossmann p. 26, lines 5-6. Routine NM_lnit initializes network manager module 1402 and so is called before any other calls in network manager module 1402. Routine NM_ Terminate closes processing of network manager module 1402 and so is called after all other calls in network manager module 1402. Rossmann p. 28, lines 39-41

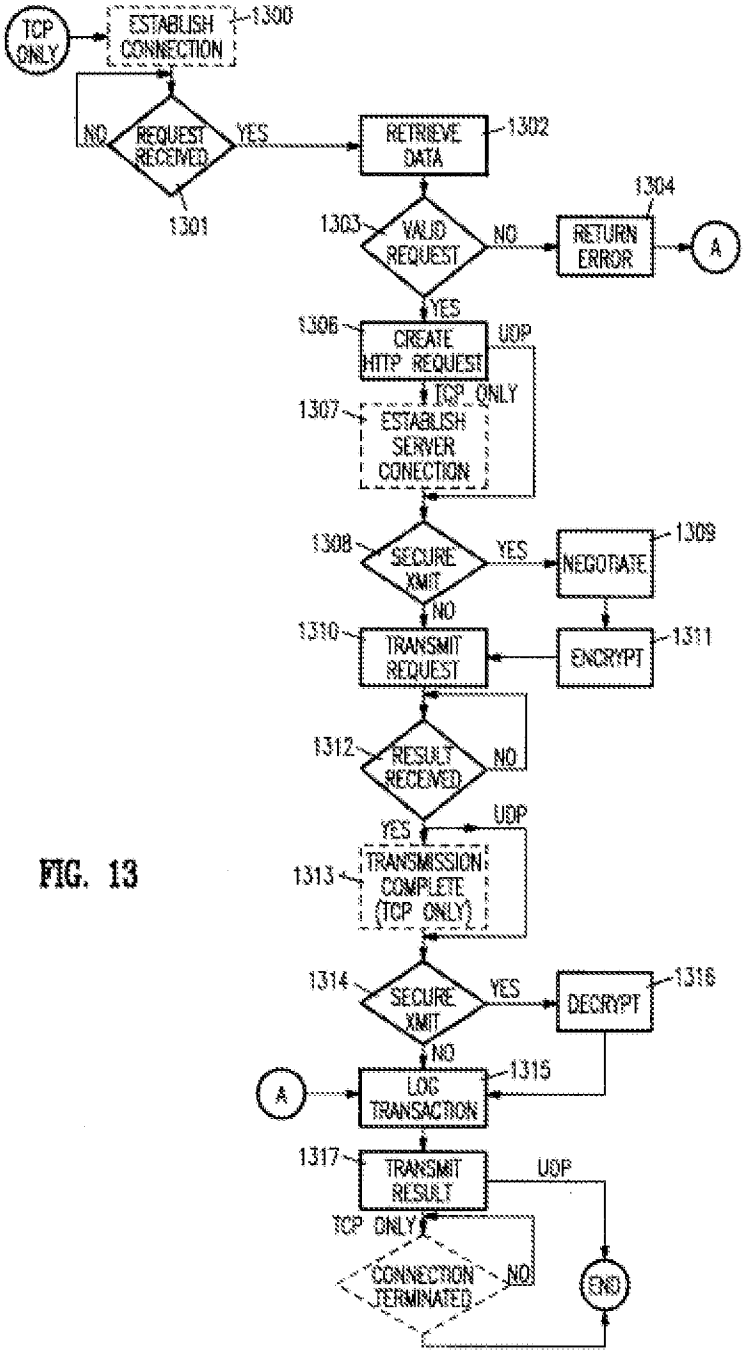


FIG. 13

A method and apparatus are disclosed for synchronizing transactions in a disconnectable network. Each transaction includes operations that were performed on a database replica on one computer while that computer was disconnected from another computer and hence from that other computer's replica. Transaction synchronization, which occurs after the computers are reconnected, transfers information from each computer to

	the other computer and applies updates to both replicas as appropriate. Transaction logs and clash handling tools may be used with the invention. Falls at Abstract The present invention provides a system and method which facilitate disconnected mobile computing in several ways. Prior to disconnection, the invention allows network administrators or users to readily select data that should be copied from a network to a mobile computer by simply identifying one or more target database subtrees. During disconnected operation of the mobile computer, the invention presents the user with a "virtual network" environment that is consistent in use and appearance with the selected portion of the actual network. Finally, upon reconnection of the mobile computer to the network, the invention synchronizes operations performed on the mobile computer during the disconnected interval with operations performed on the network during that interval. Synchronization is both substantially automatic and transactional, so minimal user intervention is needed and inconsistent internal states are avoided. Moreover, synchronization does not routinely discard any of the changes made on either the network or the mobile computer. Falls at 3:16-35 A merge process according to the present invention includes merging location sets when disconnected disconnectable computers are first connected or reconnected. For instance, merging location sets normally occurs when a computer new to the network starts up and merges into an existing location set. Falls at 16:24-29 With reference to FIG. 2, at least two of the computers 28 are disconnectable computers 40 configured according to the present invention. Each disconnectable computer 40 includes a database manager 42 which provides a location independent interface to a distributed hierarchical target database embodied in convergently consistent replicas 56. Falls at 7:16-21
(f) after said first wireless modem or wireless LAN network connection is terminated, executing at least a portion of said plurality of	For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller.

tokens representing said questionnaire at said remote computing device to collect a response from a user;	Rossmann p. 15, lines 8-12 The client process in cellular telephone 100 interprets the display card that includes image and text data and generates screen display 300 on display screen 105 (Fig. 3A). Screen display 300 includes a home key 301, and an info key 302. When the user selects home key 301, the user is returned to the home screen. Info key 302 functions in a manner similar to that described above for info key 205. When the user presses a predetermined key, the client process interprets the choice card and a second screen display 304 (Fig. 3B) is driven on display screen 105. Screen display 304 is a menu of the personal information that is stored on *server* computer 131 for use by the user of cellular telephone 100. Multi-display screen card indicator 203, e.g., the hand with a finger pointing down, illustrates to the user that the list has additional items that appear on the next screen display. Screen display 304 also indicates the number of E-mail messages, faxes, and voice messages waiting for the user. Rossmann p. 11, lines 43-52 A method and apparatus are disclosed for synchronizing transactions in a disconnectable network. Each transaction includes operations that were performed on a database replica on one computer while that computer was disconnected from another computer and hence from that other computer's replica. Transaction synchronization, which occurs after the computers are reconnected, transfers information from each computer to the other computer and applies updates to both replicas as appropriate. Transaction logs and clash handling tools may be used with the invention. Falls at Abstract The present invention provides a system and method which facilitate disconnected mobile computing in several ways. Prior to disconnection, the invention allows network administrators or users to readily select data that should be copied from a network to a mobile computer by simply identifying one or more target database subtrees. During disconnected operation of the mobile computer, the invention presents the user with a "virtual network" environment that is consistent in use and appearance with the selected portion of the actual network. Finally, upon reconnection of the mobile computer to the network, the invention synchronizes operations performed on the mobile computer during the disconnected interval with operations performed on the network during that interval. Synchronization is both substantially automatic and transactional, so minimal user intervention is needed and inconsistent internal states are avoided. Moreover, synchronization does

	not routinely discard any of the changes made on either the network or the mobile computer. Falls at 3:16-35 In operation, the replica managers synchronize transactions upon reconnection in the following manner. Using the network link, a network connection is created between the mobile computer and a network computer. The network computer need not be the network computer from which the mobile computer was disconnected. The replica manager on the mobile computer identifies a transaction that targets an object in a replica on the mobile computer, and locates a corresponding replica that resides on the network computer. The mobile computer then transfers an update based on the transaction over the network connection to the network computer. Falls at 5:21-31 Merging occurs when two replicas 56 are resynchronized after the computers 28 on which the replicas 56 reside are reconnected following a period of disconnection. Falls at 16:35-37 In summary the present invention provides a system and method for properly synchronizing transactions when a disconnectable computer 28 is reconnected to the network 10. The invention is not limited to file system operations but can instead be extended to support a variety of database objects by using the schema 84, object distributor 82, object processor 86, and other modules. Clash handling means may be used to identify potentially conflicting database changes and allow their resolution by either automatic or manual means. Clash handling and retries also make locks optional. Falls at 37:11-18
(g) establishing a second wireless modem or wireless LAN network connection between said remote computing device and a server;	According to the principles of this invention, a novel airnet network 150, i.e., a two-way data communication network, interconnects anyone, any combination, or all of two-way data communication devices 100,101, or 102, that each include this invention, with a wide variety of computer networks 120, 130, and 140, for example. As explained more completely below, each two-way data communication device 100, 101, and 102 can be configured to transmit data to and receive data from any desired combination of computers on computer networks 120, 130, and 140. Airnet network 150 is the two-way data communication path from the two-way data communication device to the particular computer that is accessed by the user of that two-way data communication device. Rossmann p. 6, lines 31-37 For example other two-way data communication networks for cellular

	telephones that may be used include TDMA, CDMA, and GSM circuit switched data networks; and the AMPS analog cellular network with a modem. Similarly, for two-way pagers, two-way data communication networks include PACT, or other priority two-way paging networks with data transport capability. Rossmann p. 14, lines 35-38 A method and apparatus are disclosed for synchronizing transactions in a disconnectable network. Each transaction includes operations that were performed on a database replica on one computer while that computer was disconnected from another computer and hence from that other computer's replica. Transaction synchronization, which occurs after the computers are reconnected, transfers information from each computer to the other computer and applies updates to both replicas as appropriate. Transaction logs and clash handling tools may be used with the invention. Falls at Abstract The present invention provides a system and method which facilitate disconnected mobile computing in several ways. Prior to disconnection, the invention allows network administrators or users to readily select data that should be copied from a network to a mobile computer by simply identifying one or more target database subtrees. During disconnected operation of the mobile computer, the invention presents the user with a "virtual network" environment that is consistent in use and appearance with the selected portion of the actual network. Finally, upon reconnection of the mobile computer to the network, the invention synchronizes operations performed on the mobile computer during the disconnected interval with operations performed on the network during that interval. Synchronization is both substantially automatic and transactional, so minimal user intervention is needed and inconsistent internal states are avoided. Moreover, synchronization does not routinely discard any of the changes made on either the network or the mobile computer. Falls at 3:16-35 Each computer's replica manager communicates with the device controller of that computer and with the network link. Each replica manager also communicates with a database manager on its computer. The database manager can send database transactions to the device controller only through the replica manager, allowing the replica managers to log transactions and to synchronize the transactions after the network connection is re-established. Falls at 4:7-14

	In operation, the replica managers synchronize transactions upon reconnection in the following manner. Using the network link, a network connection is created between the mobile computer and a network computer. The network computer need not be the network computer from which the mobile computer was disconnected. The replica manager on the mobile computer identifies a transaction that targets an object in a replica on the mobile computer, and locates a corresponding replica that resides on the network computer. The mobile computer then transfers an update based on the transaction over the network connection to the network computer. Falls at 5:21-31 More generally, the present invention provides a basis for a family of distributed software applications utilizing the target database by providing capabilities which support replication, distribution, and disconnectability. Falls at 7:53-8:1. Those of skill in the art will appreciate that other remote procedure call mechanisms may also be employed according to the present invention. Suitable network connections 52 may be established using packet-based, serial, internet compatible, local area, metropolitan area, wide area, and wireless network transmission systems and methods. Falls at 13:60-65 Merging occurs when two replicas 56 are resynchronized after the computers 28 on which the replicas 56 reside are reconnected following a period of disconnection. Falls at 16:35-37 With reference to FIGS. 1 through 4 and particular focus on FIG. 4, a method of the present invention for synchronizing transactions in the network 10 of connectable computers 28 is illustrated. The transactions target entries in a distributed hierarchical database that contains convergently consistent replicas 56 residing on separate computers 28 in the network 10. The method comprises the following computer-implemented steps. A connecting step 100 uses the replica manager 46 and network link manager 50 to establish a network connection between a first computer 36 and a second computer 38. For purposes of illustrating the method, the first computer 36 shown in FIG. 1 is a client computer 20 and the second computer 38 is a server computer 16. However, a server and a client, or two servers, or two clients, may also be synchronized and otherwise managed according to the present invention. 35:47-63

	In summary the present invention provides a system and method for properly synchronizing transactions when a disconnectable computer 28 is reconnected to the network 10. The invention is not limited to file system operations but can instead be extended to support a variety of database objects by using the schema 84, object distributor 82, object processor 86, and other modules. Clash handling means may be used to identify potentially conflicting database changes and allow their resolution by either automatic or manual means. Clash handling and retries also make locks optional. Falls at 37:11-18

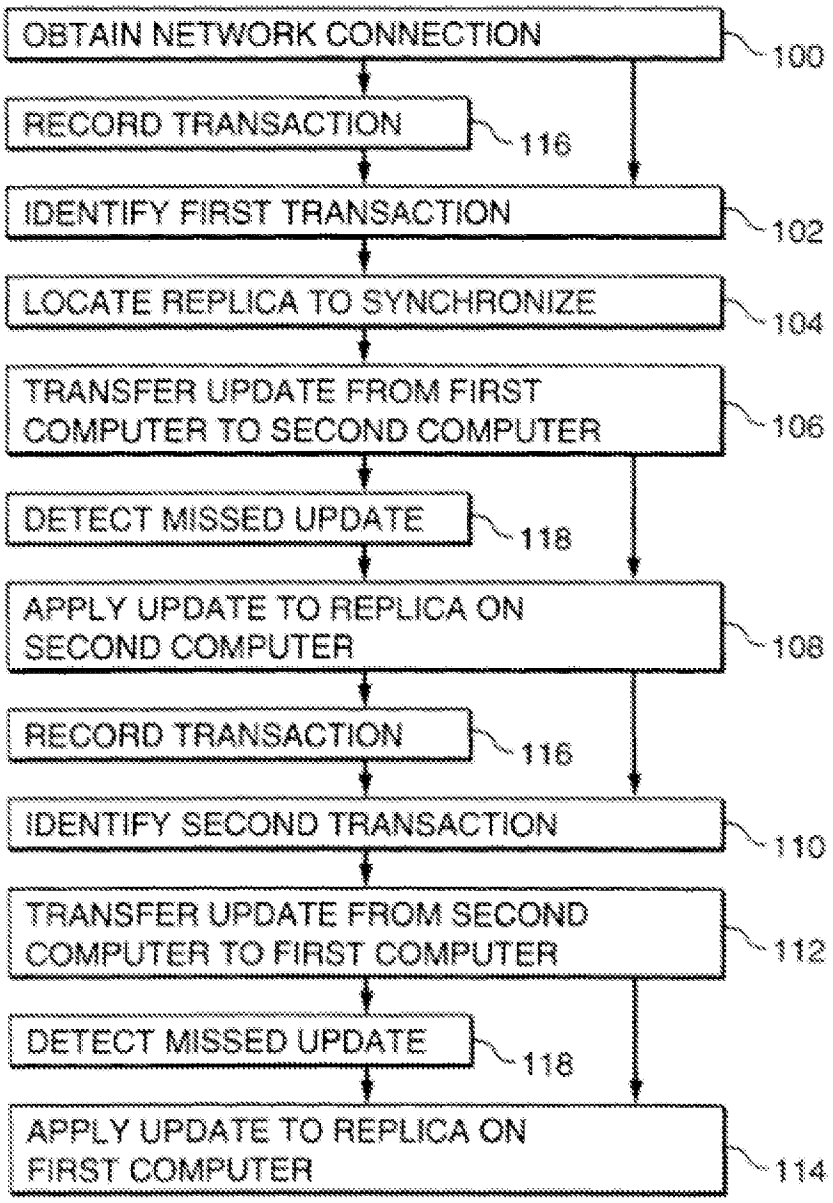


FIG. 4

(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	When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18

said second wireless modem or wireless LAN network connection; and	The user scrolls the screen display line by line until screen display 305 is on display screen 105. Initially, the fourth item in the menu is not highlighted. In this example, the user presses the four key on the keypad of cellular telephone 100 to view the user's schedule. In response to the key press, the client module in cellular telephone 100 transmits a message, including a resource locator associated with the menu item selected by pressing the four key, to server computer 131 using data capable cellular telephone network 110 and corporate local area network 130. In response to the message, server computer 131 executes the application identified in the resource locator. Upon completion of the execution, server computer 131 transmits, over corporate local area network 130 and data capable cellular telephone network 110 to cellular telephone 100, a card deck that includes a choice card that describes the user's schedule for that day. Rossmann p. 11, line 53 – Rossmann p. 12, line 2 As indicated above, each interaction with the user of cellular telephone 700 is described by a deck or a series of decks. Logically, the user retrieves a terminal interaction language deck stored in a memory 716 of cellular telephone 700 after receipt from computer 743 over CDPD network 710. The user reviews the information displayed by cards in the deck and makes choices and/or enters requested information and then requests another deck, as described above with respect to Figures 2A to 2H, for example. Rossmann p. 15, lines 23-27 A method and apparatus are disclosed for synchronizing transactions in a disconnectable network. Each transaction includes operations that were performed on a database replica on one computer while that computer was disconnected from another computer and hence from that other computer's replica. Transaction synchronization, which occurs after the computers are reconnected, transfers information from each computer to the other computer and applies updates to both replicas as appropriate. Transaction logs and clash handling tools may be used with the invention. Falls at Abstract The present invention provides a system and method which facilitate disconnected mobile computing in several ways. Prior to disconnection, the invention allows network administrators or users to readily select data that should be copied from a network to a mobile computer by simply identifying one or more target database subtrees. During disconnected operation of the mobile computer, the invention presents the user with a "virtual network" environment that is consistent in use

	and appearance with the selected portion of the actual network. Finally, upon reconnection of the mobile computer to the network, the invention synchronizes operations performed on the mobile computer during the disconnected interval with operations performed on the network during that interval. Synchronization is both substantially automatic and transactional, so minimal user intervention is needed and inconsistent internal states are avoided. Moreover, synchronization does not routinely discard any of the changes made on either the network or the mobile computer. Falls at 3:16-35 In operation, the replica managers synchronize transactions upon reconnection in the following manner. Using the network link, a network connection is created between the mobile computer and a network computer. The network computer need not be the network computer from which the mobile computer was disconnected. The replica manager on the mobile computer identifies a transaction that targets an object in a replica on the mobile computer, and locates a corresponding replica that resides on the network computer. The mobile computer then transfers an update based on the transaction over the network connection to the network computer. Falls at 5:21-31 Merging occurs when two replicas 56 are resynchronized after the computers 28 on which the replicas 56 reside are reconnected following a period of disconnection. Falls at 16:35-37 In summary the present invention provides a system and method for properly synchronizing transactions when a disconnectable computer 28 is reconnected to the network 10. The invention is not limited to file system operations but can instead be extended to support a variety of database objects by using the schema 84, object distributor 82, object processor 86, and other modules. Clash handling means may be used to identify potentially conflicting database changes and allow their resolution by either automatic or manual means. Clash handling and retries also make locks optional. Falls at 37:11-18
(i) storing said transmitted response at said server.	The server processes the message, i.e., executes the application addressed by the resource locator and transmits a response over the two-way data communication network to the two-way data communication device, which **stores** the response in a memory. Rossmann p. 4, lines 5-7

	When the user presses a predetermined key, the client process interprets the choice card and a second screen display 304 (Fig. 3B) is driven on display screen 105. Screen display 304 is a menu of the personal information that is stored on *server* computer 131 for use by the user of cellular telephone 100. Multi-display screen card indicator 203, e.g., the hand with a finger pointing down, illustrates to the user that the list has additional items that appear on the next screen display. Screen display 304 also indicates the number of E-mail messages, faxes, and *voice* messages waiting for the user. Rossmann p. 11, lines 47-52 Computer 743 may contain **stored** static telephone interaction description language decks. Computer 743 also generates telephone interaction description language decks in response to data from, or choices made by, the user of cellular telephone 700. Rossmann p. 15, lines 13-15
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	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by **interpreting** the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines 43-48
(k) accessing the translated response from a computer executing said	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process

particular computer program.	in cellular telephone 100 by **interpreting** the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 **The client process interprets the card deck** and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines 43-48
CLAIM 3	
3. The method for managing data of claim 1 wherein step (a) includes the substeps of: (a) creating a questionnaire by:	In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 4-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18 In response to entry of the purchase order number, the client process transmits a request to server computer 121 for the particular purchase order. Specifically, the client process appends the entered data to a resource locator and transmits a message containing the resource locator to server computer 121. Server computer 121, in response to the message, retrieves the **appropriate purchase order** and transmits the

	purchase order as a card deck to the client process in cellular telephone 100 over airnet network 150. The client process interprets the card deck and generates a screen display 209 (Fig. 2F). Initially, fax key 208 is not highlighted in screen display 209. Notice that screen display 209 includes multi-display screen card indicator 203 to show the user that the purchase order screen contains more information that can be displayed at one time on display screen 105. After the user reviews the purchase order, the user presses the key sequence for fax key 208 and in response, fax key 208 is highlighted as illustrated in Figure 2F. Rossmann p. 10, lines, 38-48 For simplicity, in this embodiment, each card is a single operation. Herein, an operation is defined as a related set of actions such that the user does not encounter an unanticipated delay in moving from one action to the next, i.e., the user does not have to wait for client module 702 to retrieve another card deck from computer 743. Also, a deck may include definitions of soft keys that stay in force while the deck is active, i.e., being executed by the cellular telephone microcontroller. Rossmann p. 15, lines 8-12
(i) entering a series of questions into a questionnaire design computer program;	This invention allows for the first time two-way communications devices such as cellular telephones, two-way pagers, and telephones to become open application platforms which in turn empowers software developers to deliver value-added applications and services to any two-way communication device that incorporates the principles of this invention. This is a radical shift from the current situation where telephones and two-way pagers are closed, proprietary systems. Consequently, an even playing field is created for the market to invent new uses for two-way communication devices and for two-way communication networks. Any entity from corporations to individuals can make new applications available to the installed base of two-way data communication devices that include this invention without physical modification or addition to the two-way communication device. Years after purchase, a two-way communication device incorporating this invention will run all the applications which were developed since its purchase. Rossmann p. 3, lines 41-49 In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone

	network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18
(ii) identifying within said questionnaire design computer program the type of response allowed for each question of said series of questions; and	This invention allows for the first time two-way communications devices such as cellular telephones, two-way pagers, and telephones to become open application platforms which in turn empowers software developers to deliver value-added applications and services to any two-way communication device that incorporates the principles of this invention. This is a radical shift from the current situation where telephones and two-way pagers are closed, proprietary systems. Consequently, an even playing field is created for the market to invent new uses for two-way communication devices and for two-way communication networks. Any entity from corporations to individuals can make new applications available to the installed base of two-way data communication devices that include this invention without physical modification or addition to the two-way communication device. Years after purchase, a two-way communication device incorporating this invention will run all the applications which were developed since its purchase. Rossmann p. 3, lines 41-49 When the user selects the at least one user data input option, the client module interprets the selection and if required, appends any input data to the resource allocator associated with the at least one user data input option. The client module transmits a message including the resource locator with any appended input data to the server computer.

	Alternatively, the resource locator with any appended data can be addressed to another server computer, or can address an object stored in the two-way communication device. If the resource locator addresses an object on a server computer, the client module provides the message to the network interface module which in turn transmits the message over the two-way data communication network. Rossmann p. 4, lines 15-21 An important aspect of this invention is that the message includes all information necessary for the client module to generate the user interface and a particular user interface can be independent from other user interfaces. Unlike prior art systems that gave the user a predetermined menu from which to select items, or limited the user to an E-mail like format, according to the principles of this invention, the user interfaces and possible interactions available to the user are determined only by the applications that developers make available. The possible interactions and user interfaces for one application can be totally different and independent from the possible interactions and user interfaces of another application. Thus, a cellular telephone, two-way pager, and a telephone all truly become an open platform. Rossmann at p. 4, lines 29-35 In response to the access by the user, server computer 121 transmits a card deck to cellular telephone 100 over data capable cellular telephone network 110. As explained more completely below, a card deck includes one or more cards, and each card is interpreted by the client module to generate a user interface screen. In the embodiment illustrated in Figure 2A, the initial card deck transmitted to cellular telephone 100 includes an introductory display card and a choice card. Figure 2A is an example of introductory screen display 200 that is generated on display screen 105 by the client process in cellular telephone 100 by interpreting the display card. As used herein, a display screen is the physical display apparatus in a two-way communication device. A screen display is the image presented on the display screen. Rossmann p. 9, lines 1-8 When the user presses a predetermined key, or key sequence, the client process in cellular telephone 100 interprets the next card in the card deck, i.e., the choice card, and in turn generates a menu 201 (Fig. 2B) of items that can be accessed by the user. In this embodiment, each of the menu items is available on server computer 121 to the user who, in this example, is a representative of XYZ corporation visiting ABC Designs. Rossmann p. 9, lines 15-18