Intelligent Agent Processing System and Method

Kenneth Baclawski, Waltham, MA

Abstract

A distributed computer database system including one or more front end computers and one or more computer nodes interconnected by a network into a processing engine for software agents. A software agent is a computer software system that acts autonomously, perceives its environment, interacts with other agents and can engage in behavior directed toward the fulfillment of a goal. Each agent registers its interest with the processing engine. An agent interest is an object in the same format as the objects that are examined by the processing engine. The processing engine examines an object in its environment with examination nodes. To examine an object, the examination node begins by extracting features from the object, fragmenting the features, and hashing these fragments. Each hashed fragment is transmitted to one node on the network. Each node on the network which receives a hashed fragment uses the hashed fragment of the object to perform a search on its respective partition of the database. The results of the searches of the local databases are gathered by the examination node. The gathered results are used to identify those agents for which the object is relevant to the interest of the agent. If it is determined that the object is relevant to an agent, the agent is notified and given access to the object. An agent can locate other agents by giving an object to an examination node. The agents which are determined to have an interest in this object are notified. In this way an agent can establish communication with one or more other agents.

References

[1] L. Aiello, J. Doyle, and S. Shapiro, editors. *Proc. Fifth Intern. Conf. on Principles of Knowledge Representation and Reasoning.* Morgan Kaufman Publishers, San Mateo, CA, 1996.

[2] K. Baclawski. Distributed computer database system and method, December 1997. United States Patent No. 5,694,593. Assigned to Northeastern University, Boston, MA.

[3] A. Del Bimbo, editor. *The Ninth International Conference on Image Analysis and Processing*, volume 1311. Springer, September 1997.

[4] N. Fridman. *Knowledge Representation for Intelligent Information Retrieval in Experimental Sciences.* PhD thesis, College of Computer Science, Northeastern University, Boston, MA, 1997.

[5] R. Jain. Content-centric computing in visual systems. In *The Ninth International Conference on Image Analysis and Processing, Volume II*, pages 1–13, September 1997.

[6] Y. Ohta. *Knowledge-Based Interpretation of Outdoor Natural Color Scenes.* Pitman, Boston, MA, 1985.

[7] G. Salton. *Automatic Text Processing.* Addison-Wesley, Reading, MA, 1989.

[8] G. Salton, J. Allen, and C. Buckley. Automatic structuring and retrieval of large text files. *Comm. ACM*, 37(2):97–108, February 1994.

[9] A. Tversky. Features of similarity. *Psychological review*, 84(4):327–352, July 1977.

[10] M. Woodridge and N. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

1 Field of the Invention

The invention relates to computer database systems and more specifically to distributed computer database systems.

2 Background of the Invention

A large variety of software tools have been developed whose purpose is to assist people in their day to day activities, such as word processors, spreadsheets, scheduling software, etc. Most of these software tools require direct interaction with a person to activate their functionality. A new class of software tool is being developed that assists people without requiring direct interaction. For example, software that reminds an individual about an appointment about to take place. Another example would be software that examines a variety of news sources, notifying the user when a relevant article has become available.

A more sophisticated example would be software that not only examines news sources, but also extracts the important features in each article. For example, the agent might be watching for announcements of changes in an interest rate. Based on the relevant extracted features, the agent would automatically negotiate with agents that are following the stock prices of companies in which the user has an investment, possibly resulting in a stock trade. The time from the detection of the news report to the completion of the stock trade would be measured in hundreds of milliseconds. The user would only be notified after the actions have taken place.

Software of the kind described above is called *agent software*. An *agent* is a software artifact whose purpose is to act as a surrogate (or assistant) to a human or another software artifact. More precisely, a software agent is a software-based computer system with the following properties:

1. Autonomy. An agent operates in the "background" as an independent process with control over its actions and state.

2. Socialability. An agent can interact with other agents (and humans).

3. Reactivity. An agent examines its environment and responds to changes that occur in it.

4. Pro-activeness. An agent can engage in behavior directed to a goal.

A simple example of a software agent is an autonomous query for documents on the World Wide Web (WWW), which "crawls" around the Internet looking for documents that satisfy the query. When a document satisfying the query is found, the user is notified and the document is presented.

Agents are currently implemented as independent software systems, each of which is fully responsible for all of its functionality such as examining its environment and detecting changes that are relevant as well as finding and establishing communication with other agents. Such a design does not scale up well to millions of agents, all of whom will be independently examining the environment. For example, in a military system, one might assign an agent to represent every entity of interest in the entire theatre of operations: soldiers, weapons, targets, etc. The number of agents in such a system will be very large, yet the response time requirements of the system are severe.

The environment which the agents are examining can be structured in two ways: either the content providers provide the content as it becomes available without a specific request, or each information object is explicitly requested. Either way, the agent environment consists of a stream of information objects. Information objects can be images, sound and video streams, as well as traditional data objects such as text files and structured documents. A search engine that examines a stream of information objects to determine relevance with respect to predefined queries is called a *routing search engine*.

3 Summary of the Invention

In the present invention, the notification of agents interested in an object is based on an ontology. An *ontology* models knowledge within a particular domain. For example, a financial services ontology would be used for agents that are observing news reports for information that may be relevant to stock trading. Similarly, a military ontology would be appropriate for agents that represent entities in the theatre of operations.

An ontology can include a concept network, specialized vocabulary, syntactic forms and inference rules. In particular, an ontology specifies the features that objects can possess as well as how to extract features from objects. The extracted features are used for determining the degree of similarity between an object and the interest of a registered agent. Each feature of an object may have an associated weight, representing the strength of the feature. Current technology supports only a very limited ontology with just a few concepts. Moreover, the ontology supported by current technology is inflexibly built into the search engine. The present invention allows indexing and retrieval based on an arbitrary ontology.

Furthermore, the present invention supports very large numbers of agents, tens of millions or more, all simultaneously examining the environment. Each agent must register with the routing search engine by specifying a query that determines the changes in the environment to which the agent has an interest. Each agent is uniquely identified by an agent identifier (AID).

The invention relates to a distributed computer database system which includes one or more computer nodes interconnected by a network. The combination of computer nodes interconnected by a network operates as a routing search engine.

The main function of the routing search engine is to examine the environment. This is performed by the the examination nodes. Examination nodes can include Web "crawlers," database scanners, agent registration servers and agent communication servers. When an examination node examines an object in the environment, it extracts the features of the object as determined by the ontology of the routing search engine. The object features are then hashed. A portion of each hashed feature is used by the examination node as an addressing index by which the examination node transmits the hashed object feature to an index node on the network.

Each node on the network which receives a hashed object feature uses the hashed object feature to perform a search on its respective database. Nodes finding data corresponding to the hashed object feature return the AIDs of the agents that have registered an interest in this feature. Such AIDs are then gathered by the examination node and a similarity function is computed based on the features that are in common with the object as well as the features that are in the object but not registered by the agent. The similarity function is used to determine whether an agent is to be notified.

The AIDs of the agents to be notified are transmitted by the examination node to the agent nodes. A portion of each AID is used by the examination node as an addressing index by which the examination node transmits the AID and object information to an agent node on the network. Each agent node on the network which receives an AID and object information uses the AID to perform a search on its respective database. Nodes finding data corresponding to the AID use this data to notify the agent that an object has been encountered. The agent node also transmits the object information to the agent. The agent performs whatever processing necessary to respond to the object information, such as notifying the user who owns the agent.

An agent locates other agents by transmitting an information object to an examination node for this purpose. The examination node processes this information object as described above, except that the agents which are determined to be interested in the information object are notified that another agent wishes to establish communication.

A user wishing to register an agent with the routing search engine sends the request to an examination node. The examination node assigns an AID to the agent. The request for registration includes an information object which determines which objects encountered in the environment are of interest to the agent. In a manner similar to that described above, the agent object is fragmented, hashed and transmitted to the index nodes which store data relating the hashed fragment to the AID of the agent. The examination node also transmits any additional agent information, such as its location, to one of the agent nodes as determined by the AID of the agent.

4 Description of the Drawings

This invention is pointed out with particularity in the appended claims. The above and further advantages of the invention may be better understood by referring to the following description taken in conjunction with the accompanying drawing, in which:

FIG. 1 is a block diagram of an overview of an embodiment of the distributed computer database system of the invention;

FIG. 2 is an overview of the steps used by the embodiment of the distributed computer database system to respond to an object in the environment;

FIG. 3 is an overview of the steps used by the embodiment of the distributed computer database system to register an agent.

FIG. 4 specifies the formats of the messages transmitted between the nodes of the distributed computer database system.

The remaining diagrams are block diagrams of the modules that perform the tasks of the invention within each node.

5 Detailed Description of the Preferred Embodiment

Referring to FIG. **1**, in broad overview, one embodiment of a distributed computer database system of the invention includes a routing search engine which includes one or more computer nodes interconnected by a local area network. The individual computer nodes may include local disks, or may, alternatively or additionally, obtain data from a network disk server.

The computer nodes of the routing search engine may be of several types, including examination nodes, index nodes and agent nodes. The nodes of the routing search engine need not represent distinct computers. In one embodiment, the routing search engine consists of a single computer which takes on the roles of all examination nodes, index nodes and agent nodes. In another embodiment, the routing search engine consists of separate computers for each examination node, index node and agent node. Those skilled in the art will realize many variations are possible which will still be within the scope and spirit of the present invention.

Considering the processing of a object in the environment first, and referring also to FIG. **2**, in one embodiment when an information object is examined, the examination node extracts features from the object according to the ontology. Feature extraction may be performed using traditional techniques for associating values to attributes, such as in relational database records. Note that an object either possesses a feature or it does not. This property is distinct from the value associated with a feature when an object possesses the feature.

Features are extracted from structured documents by parsing the document to produce a data structure, then dividing this data structure into (possibly overlapping) substructures called fragments. The fragments of a structured document are the features extracted from the document. Fragments of an information object are used to find matching fragments in the database, so they are also called *probes.* This same terminology will be used for features extracted from other kinds of objects as well.

Features are extracted from unstructured documents by using knowledge extraction techniques. Knowledge extraction produces a data structure consisting of a collection of inter-related knowledge frames. The knowledge frame data structure is then divided into (possibly overlapping) substructures, as in the case of a structured document, and these substructures are the features of the unstructured document.

A large variety of feature extraction algorithms have been developed for media such as sound, images and video streams. Fourier and Wavelet transformations as well as many filtering algorithms are used. Features can also be added to an agent by manual or semi-automated means. Such added features are referred to as annotations or meta-data. Features are extracted from annotations using one of the techniques mentioned above, depending on whether the annotation is a relational database record, a structured document or an unstructured document. Each feature can have a value associated with it, and one can specify relationships between features which can also have values associated with them. For example, one feature can be contained within another feature or be adjacent to another feature. The ontology specifies the feature extraction algorithms as well as the structure of the features.

If a feature occurs very commonly in the environment, then it does not contribute to the purpose of the routing search engine; namely, distinguishing those agents that are interested in a particular object. An example is the brightness of an image. Such a feature will be partitioned into a collection of contiguous, non-overlapping ranges of the value associated with the feature rather than the feature itself. Each range of the value is then regarded as a separate feature.

The examination node then encodes each feature of the object by using a predefined hashing function. Data in the system was previously stored locally on the various index nodes using this hashing function to generate an index to the data in the local database. Thus, the use of the same hashing function to generate an index for data storage and to generate hashed probes for an information object assures that 1.) data is distributed uniformly over the index nodes of the routing search engine during the storing of data and 2.) the probes are scattered uniformly over the index nodes during the processing of an object.

In one embodiment, the hash value resulting from the use of the hashing function has a first portion which serves to identify the index node to which the data is to be sent to be stored or to which an object feature is to be sent as a probe and a second portion which is the local index value which is used to determine where data is to be stored at or retrieved from the index node. Thus, in terms of an object, the hashed object features are distributed (Step **201**) as probes to certain index nodes of the routing search engine, as determined by the first portion of the hash value.

Index nodes whose probes match the index features by which the data was initially stored on that index node respond to the object by transmitting (Step **202**) the AIDs matching the index terms of the requested information to the examination node. Thus all matches between the hashed probes and the local hash table of index terms are returned or gathered to the examination node which initially hashed the object features.

The examination node then determines the relevance of the object for each agent returned in the search. The determination of relevance is made by the examination node by comparing the degree of similarity between the object and the agents whose AIDs were returned. In one embodiment the measure of similarity between the object and the agent is a cosine measure and is given by the expression $COS(v, w)$, where the vector v denotes the object and the vector w denotes the agent. These vectors are in a space in which each feature represents one dimension of the space.

Another commonly used measure of similarity between two agents is a distance function in the same space mentioned above for the cosine measure. However, there is convincing evidence that human similarity does not satisfy the axioms of a distance function. The model that currently seems to be the most successful approach is the Feature Contrast Model of Tversky. In this model, the similarity between an object and an agent is determined by three terms:

1. The features that are common to the object and the agent.

2. The features of the agent that are not features of the object.

3. The features of the object that are not features of the agent.

The first term contributes a positive number to the similarity value, while the second and third terms have negative contributions. In addition the second and third terms are multiplied by predefined constants such that a feature in the second and third set has less effect on the similarity than one in the first set.

In one embodiment the measure of similarity between the object and the agent is a measure determined by three predefined constants that are used to multiply the three terms occurring in the Feature Contrast Model. In this embodiment, only the first two terms of the Feature Contrast Model are used to compute the measure of similarity, or equivalently, the predefined constant for the third term is set to zero.

In one embodiment all agents which generate similarity values greater than a predetermined value are considered sufficiently similar to the object to result in the agents being notified. In another embodiment each agent can specify the minimum similarity value which determines whether the agent will be notified.

Once the similarity is determined, the examination node transmits the AIDs for which the object has been determined to be relevant to the agent nodes (Step **203**) which hold information associated with the agents identified by the AIDs. The examination node also transmits object information to each agent node. In one embodiment, the object information is the URL for the object. In another embodiment, the object information is the object itself. In another embodiment, the object information is the list of all features of the object and the values of the features for those features that have associated values.

In one embodiment, each AID has a first portion which serves to identify the agent node on which the agent information is stored and a second portion which is the local index value which is used to determine where the agent information is stored in a local table at the agent node.

The agent information associated with the AID and stored in the local table is used to notify the agent that an object has been examined which may be relevant to the agent (Step **204**). Along with the notification, the agent node transmits object information to the agent. The agent may then respond to the object encountered in its environment.

An agent can communicate with other agents as a consequence of responding to changes in its environment or as a result of behavior directed toward achieving its goal. For an agent to communicate with other agents, it must either already possess the location of the other agents or else it must locate the other agents. In one embodiment, an agent wishing to interact with other agents constructs an object and transmits it to an examination node (Step **205**). The object is examined and processed as specified above to determine the agents that have an interest in the constructed object, except that after notifying the agents, a communication link is established between each of these agents and the agent that constructed the object.

Considering next the processing of a request by a user to register an agent and referring also to FIG. **3**, in one embodiment when a request to register an agent is received by an examination node (Step **301**), the request includes an object that defines which objects in the environment the agent has an interest. The examination node assigns a unique AID to the agent, then processes the object as discussed above in the case of examination of an object in the environment (Step **302**), except that data associated with the agent is stored in the index nodes and an agent node.

Considering next the message formats used in the preferred embodiment, refer to FIG. **4**. The Fragment Message has four fields: Header, Object Identifier (OID), Hashed Object Fragment (HOF) and Value. The Header field specifies that this message is a Fragment Message and also specifies the destination index node. The destination index node is determined by the first portion of the hashed object fragment. The OID field contains an object type specifier and an object identifier. The HOF field contains a fragment type specifier and the second portion of the hashed object fragment produced by the Hashing Module. The Value field contains an optional value associated with the fragment. The fragment type specifier determines whether the Fragment Message contains a Value field, and if the Fragment Message does contain a Value field then the fragment type specifier determines the size of the Value field.

The Fragment Response Message contains four fields: Header, OID, Agent Identifier (AID) and Weight. The Header field specifies that this message is a Fragment Response Message and also specifies the destination examination node. The destination examination node is the examination node from which the corresponding Fragment Message was received. The OID field contains an object type specifier and an object identifier. The AID field contains an agent type specifier and an agent identifier. The Weight field contains an optional weight associated with the agent. The agent type specifier determines whether the Fragment Response Message contains a Weight field, and if the Fragment Response Message does contain a Weight field then the agent type specifier determines the size of the field.

The Agent Communication Message has two parts: Identifier and Feature. The Identifier part has two fields: Header and AID. The Header field specifies that this message is an Agent Communication Message and also specifies a destination examination node. The destination examination node is determined by the first portion of the agent identifier. The AID field contains an agent type specifier and an agent identifier. The Feature part contains a number of features that are specified by the agent as being relevant.

The Object Message has three parts: Identifier, Feature and Auxiliary. The Identifier part has four fields: Header, AID, OID and Location. The Header field specifies that this message is an Object Message and also specifies the destination agent node. The destination agent node is specified by the first portion of the agent identifier. The AID field contains an agent type specifier and the second portion of the agent identifier. The OID field contains an object type specifier and the object identifier. The Location field contains an optional location specifier such as a URL. The object type specifier determines whether the Object Message contains a Location field, and if the Object Message does contain a Location field, then the object type specifier determines the size of the Location field. The Feature part contains a number of features associated with the object. The Auxiliary part contains auxiliary information associated with the object. The object type specifier determines whether the Object Message contains an Auxiliary part, and if the Object Message does contain an Auxiliary part, then the object type specifier determines the size and structure of the Auxiliary part.

The Insert Fragment Message has four fields: Header, AID, hashed agent fragment (HAF) and Value. The Header field specifies that this message is an Insert Fragment Message and also specifies the destination index node. The destination index node is determined by the first portion of the hashed agent fragment. The AID field contains an agent type specifier and an agent identifier. The HAF field contains a fragment type specifier and the second portion of the hashed agent fragment produced by the Hashing Module. The Value field contains an optional value associated with the fragment. The fragment type specifier determines whether the Insert Fragment Message contains a Value field, and if the Insert Fragment Message does contain a Value field then the fragment type specifier determines the size of the Value field.

The Agent Registration Message has two parts: Identifier and Program. The Identifier part has three fields: Header, AID and Location. The Header field specifies that this message is an Agent Registration Message and also specifies the destination agent node. The destination agent node is determined by the first portion of the agent identifier. The AID field contains an agent type specifier and the second portion of the agent identifier. The Location field contains an optional location specifier such as a URL. The agent type specifier determines whether the Agent Registration Message contains a Location field, and if the Insert Agent Message does contain a Location field, then the agent type specifier determines the size of the Location field. The Program part is an optional specification of the behavior of the agent. For example, it can consist of an executable program, the source code of a program or a specification of the agent.

Considering next the Communication Module contained in the computer nodes used in the preferred embodiment, refer to Fig. 5, 6 and 7. The Communication Module is responsible for transmitting and receiving messages from one node to another. The destination node for a message to be transmitted is specified in the Header field of each message. When a message is received from another node, the type of message determines which module will process the message. The message type is specified in the Header field of each message.

Considering next the modules contained in the examination nodes used in the preferred embodiment, refer to Fig. 5. The Feature Extractor extracts features from an object or agent. Feature extraction for images is performed by detecting edges, identifying the image objects, classifying the image objects as domain objects and determining relationships between domain objects. In another embodiment, feature extraction for images is performed by computing Fourier or wavelet transforms. Each Fourier or wavelet transform constitutes one extracted feature. The extracted features are transferred to the Fragmenter. In addition, when features have been extracted from an agent, the features are transferred to the Communication Module in the form of an Agent Registration Message.

The Fragmenter computes the fragments contained in each feature. Each fragment consists of a bounded set of related components in the feature. In one embodiment, the fragments of a feature consist of each attribute and each relationship in the data structure defining the feature. The fragments are transferred to the Hashing Module.

The Hashing Module computes a hash function of a fragment. In one embodiment, the hash function is the MD4 message digest function. The Hashing Module transfers either a Fragment Message or an Insert Fragment Message to the Communication Module, depending on whether the fragment is an object fragment or an agent fragment, respectively.

The Similarity Comparator receives Fragment Response Messages and produces Object Messages which are transferred to the Communication Module. The Similarity Comparator gathers all the fragment responses for an object. For each object in the responses, the Similarity Comparator determines the relevance for each agent returned in the search. This determination of relevance is made by the examination node by comparing the degree of similarity between the object and the agents whose AIDs were returned. In one embodiment the measure of similarity between the object and the agent is a cosine measure and is given by the expression $COS(v, w)$, where the vector v denotes the object and the vector w denotes the agent. These vectors are in a space in which each fragment represents one dimension of the space. The most relevant AIDs are transferred to the Communication Module using an Object Message.

The Downloader is responsible for downloading objects. Each examination node may have a different Downloader module. For example, the Downloader can be a Web crawler that finds Web pages on the WWW by extracting hypertext links from the Web pages that have been previously downloaded. For another example, the Downloader can be an Information and Content Exchange (ICE) subscriber that negotiates to obtain content from syndicators. This is the preferred mechanism for obtaining time-sensitive content such as news feeds. For yet another example, the Downloader can download information through a corporate intranet from corporate databases using a data warehousing mechanism.

Considering next the modules contained in the index nodes used in the preferred embodiment, refer to Fig. 6. The Fragment Table receives Fragment Messages and Insert Fragment Messages. In the case of a Fragment Message the Fragment Table retrieves an entry in the local hash table using the hash value in the HOF field. The type specifier in the HOF field and the entry in the local hash table are transferred to the Fragment Comparator. In the case of an Insert Fragment Message, the Fragment Table modifies the entry in the local hash table determined by the HAF field by appending the AID and Value fields of the Insert Fragment Message to the entry in the local hash table.

The Fragment Comparator receives entries from the Fragment Table. A comparison function is determined by the HOF type specifier that was transferred from the Fragment Table. The comparison function is used to determine the relevance of the AID and Value fields in the entry that was transferred from the Fragment Table. In one embodiment, the comparison function determines a similarity weight, and the AIDs having the highest similarity weight are deemed to be relevant. The relevant AIDs and their similarity weights are transferred to the Communication Module using a Fragment Response Message.

Considering next the modules contained in the agent nodes used in the preferred embodiment, refer to Fig. 7. The Agent Table receives Object Messages and Agent Registration Messages. In the case of an Object Message, the Object Table retrieves an entry in the local table using the agent identifier in the AID field of the Object Message. The Object Message and the retrieved entry are transferred to the Agent Processor. In the case of an Agent Registration Message, the Agent Table inserts a new entry in the local table. If an entry already exists for the specified agent identifier, then the existing entry is replaced. The new or replacement entry contains the information in the Agent Registration Message.

The Agent Processor receives Object Messages and entries from the Agent Table. The Agent Processor activates the agent specified by the agent identifier. Activation depends on the agent type specifier in the AID field. The agent type specifier determines how the behavior of the agent is specified in the entry obtained from the Agent Table. For example, if the entry consists of an executable program, then the Agent Processor executes the program and transfers the data contained in the Object Message to the executing program. For another example, if the entry consists of the source code of a program, then the Agent Processor interprets the program and transfers the data contained in the Object Message to the program being interpreted. For yet another example, if the entry consists of a specification of the agent, then the Agent Processor determines what actions should be performed by the agent based on the specifications and the data contained in the Object Message, and those actions are performed. If the entry only specifies the location of the agent, then the Agent Processor transfers the data contained in the Object Message to the specified location.

6 Claims

Having shown the preferred embodiment, those skilled in the art will realize many variations are possible which will still be within the scope and spirit of the claimed invention. Therefore, it is the intention to limit the invention only as indicated by the scope of the claims.

What is claimed is:

1. A method for object examination in a distributed computer database system having a plurality of examination nodes and a plurality of index nodes connected by a network, said method comprising the steps of:

   (a) extracting, by an examination node, a plurality of features from an object;

   (b) hashing, by said examination node, each said object feature of said plurality of object features, said hashed object feature having a first portion and a second portion;

   (c) transmitting, by said examination node, each said hashed object feature of said plurality of object features to a respective one of said plurality of index nodes indicated by said first portion of each said hashed object feature;

   (d) using by said index node, said second portion of said respective hashed object feature to access data according to a local hash table located on said index node;

(e) returning, by each said index node accessing data according to said respective hashed object feature, a plurality of agent identifiers corresponding to said accessed data to said examination node.

(f) transmitting, by said selected examination node, each said agent identifier of said plurality of agent identifiers to a respective one of said plurality of agent nodes indicated by said first portion of each said agent identifier;

(g) using by said agent node, said second portion of said respective agent node to access data according to a local agent table located on said agent node;

(h) using by said agent node, said accessed data to notify the agent identified by said agent identifier that an object has been examined that may be relevant to the said agent; and

(i) transmitting, by said agent node, object information to said agent.

2. The method of claim **1** further comprising the steps of:

   (a) determining, by said examination node, a measure of similarity between said accessed data and said object; and

   (b) returning to said user, by said examination node, accessed data having a predetermined degree of similarity,

   subsequent to the step of returning said plurality of agent identifiers.

3. The method of claim **2** wherein said measure of similarity is determined by a similarity function based on:

   (a) features possessed by both the said accessed data and the said object; and

   (b) features possessed only by the said object.

4. A method of storing agents or locations of agents in a manner which is conducive to routing retrieval in a distributed computer database system having a plurality of examination nodes and a plurality of index nodes connected by a network, said method comprising the steps of:

   (a) selecting, by an examination node, a unique agent identifier for an agent submitted by a user for registration, said agent identifier having a first portion and a second portion;

   (b) using the said first portion of said agent identifier to select one of said plurality of agent nodes;

   (c) transmitting, by said examination node, agent information to said selected agent node;

(d) using, by said selected agent node, said second portion of said agent identifier to store data according to a local agent table located on said agent node;

(e) extracting, by said examination node, a plurality of features from the object defining the interests of an agent submitted by a user for registration;

(f) hashing, by said examination node, each said agent feature of said plurality of agent features, said hashed agent feature having a first portion and a second portion;

(g) transmitting, by said examination node, each said hashed agent feature of said plurality of features to a respective one of said plurality of index nodes indicated by said first portion of each said hashed agent feature; and

(h) using, by said index node, said second portion of said respective hashed agent feature to store data according to a local hash table located on said index node.

5. The method of claim **4** further comprising the step of receiving, at said examination node, said agent from said user, prior to the step of extracting features from said agent.

6. A distributed computer database system having a routing retrieval tool for handling objects comprising:

(a) a plurality of examination nodes;

(b) a plurality of index nodes; and

(c) a plurality of agent nodes;

(d) said plurality of examination nodes, said plurality of index nodes, and said plurality of agent nodes connected by a network.

(e) wherein each said examination node, upon examining an object, extracts a plurality of features from said object, hashes each said object feature of said plurality of object features into a hashed object feature having a first portion and a second portion, and transmits each said hashed object feature to a respective one of said plurality of index nodes indicated by said first portion of said hashed object feature,

(f) further wherein each said index node uses said second portion of said hashed object feature to access data according to a local hash table located on said index node and returns a plurality of agent identifiers corresponding to said accessed data to said examination node,

(g) further wherein the said examination node, upon receiving a plurality of agent identifiers from each said index node, divides each said agent identifier of said plurality of agent identifiers into a first portion and a second portion, and transmits each said agent identifier to a respective one of said plurality of agent nodes indicated by said first portion of said agent identifier,

(h) further wherein each said agent node uses said second portion of said agent identifier to access data according to a local agent table located on said agent node, uses said accessed data to notify the agent identified by said agent identifier that an object has been examined that may be relevant to the said agent, and transmits object information to said agent.

7. The method of claim **6** wherein said examination node measures similarity using a similarity function determined by:

   (a) features possessed by both the said accessed data and the said object; and

   (b) features possessed only by the said object.

8. A distributed computer database system for registration of agents, comprising:

   (a) a plurality of examination nodes;

   (b) a plurality of agent nodes; and

   (c) a plurality of index nodes;

   (d) said plurality of examination nodes, said plurality of agent nodes and said plurality of index nodes connected by a network.

   (e) wherein each said examination node, upon receiving an object defining the interests of an agent from a user, selects a unique agent identifier having a first portion and a second portion, extracts a plurality of features from said agent, hashes each said agent feature of said plurality of agent features into a hashed agent feature having a first portion and a second portion, transmits agent information to a respective one of said plurality of agent nodes indicated by said first portion of said agent identifier, and transmits each said hashed agent feature to a respective one of said plurality of index nodes indicated by said first portion of said hashed agent feature,

   (f) wherein the said agent node uses said second portion of said agent identifier to store agent information in a local table, and

   (g) wherein each said index node uses said second portion of said hashed agent feature to store the agent identifier according to a local hash table located on said index node.

9. A distributed computer database system for registration of agents, comprising:

   (a) a plurality of examination nodes;

   (b) a plurality of agent nodes; and

   (c) a plurality of index nodes;

   (d) said plurality of examination nodes, said plurality of agent nodes and said plurality of index nodes connected by a network.

   (e) each said examination node, upon receiving a command from a user, enqueuing a predetermined task in response to said command,

   (f) a registration task enqueued, in response to a registration command from said user, selecting a unique agent identifier having a first portion and a second portion, for the agent contained in said registration command, extracting a plurality of features from said agent, hashing each said agent feature of said plurality of agent features into a hashed agent feature having a first portion and a second portion, transmitting an agent message containing agent information to a respective one of said plurality of agent nodes indicated by said first portion of said agent identifier, and transmitting a registration message containing each said hashed agent feature to a respective one of said plurality of index nodes indicated by said first portion of said hashed agent feature,

   (g) said agent node, upon receipt of said agent message, using said second portion of said agent identifier to store data according to a local table located on said agent node, and

   (h) said index node, upon receipt of said registration message, using said second portion of said hashed agent feature to store data according to a local hash table located on said index node.

10. The method of claim **1** further comprising the steps of:

   (a) receiving, at said examination node, an object from an agent, prior to the step of extracting features from said object; and

   (b) establishing communication between said agent and each agent in said plurality of agents for which the said object from said agent is determined to be relevant, after the step of transmitting object information to said plurality of agents.

11. The method of claim **10** further comprising the steps of:

   (a) determining, by said examination node, a measure of similarity between said accessed data and said object; and

(b) returning to said user, by said examination node, accessed data having a predetermined degree of similarity,

subsequent to the step of returning said plurality of agent identifiers.

12. The method of claim **11** wherein said measure of similarity is determined by a similarity function based on:

    (a) features possessed by both the said accessed data and the said object; and

    (b) features possessed only by the said object.

13. The method of claim **6** further comprising the steps of:

    (a) receiving, at said examination node, an object from an agent, prior to the step of examining said object; and

    (b) establishing communication between said agent and each agent in said plurality of agents for which the said object from said agent is determined to be relevant, after the step of transmitting object information to said plurality of agents.

14. The method of claim **13** wherein said examination node measures similarity using a similarity function determined by:

    (a) features possessed by both the said accessed data and the said object; and

    (b) features possessed only by the said object.

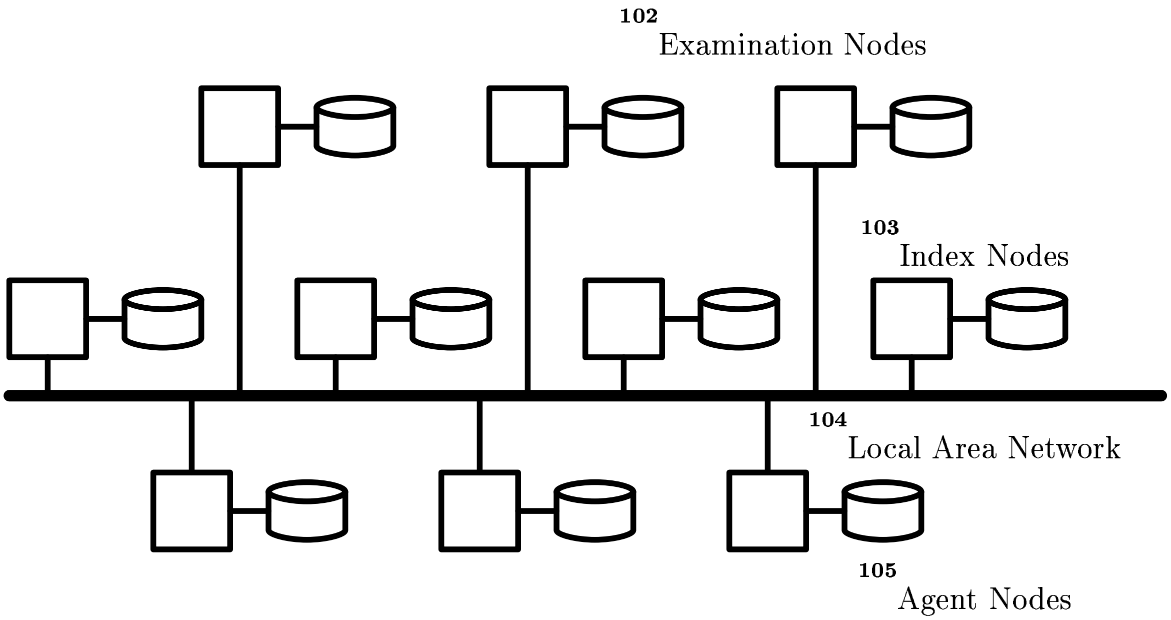
102
Examination Nodes
103
Index Nodes
104
Local Area Network
105
Agent Nodes

FIG. **1**

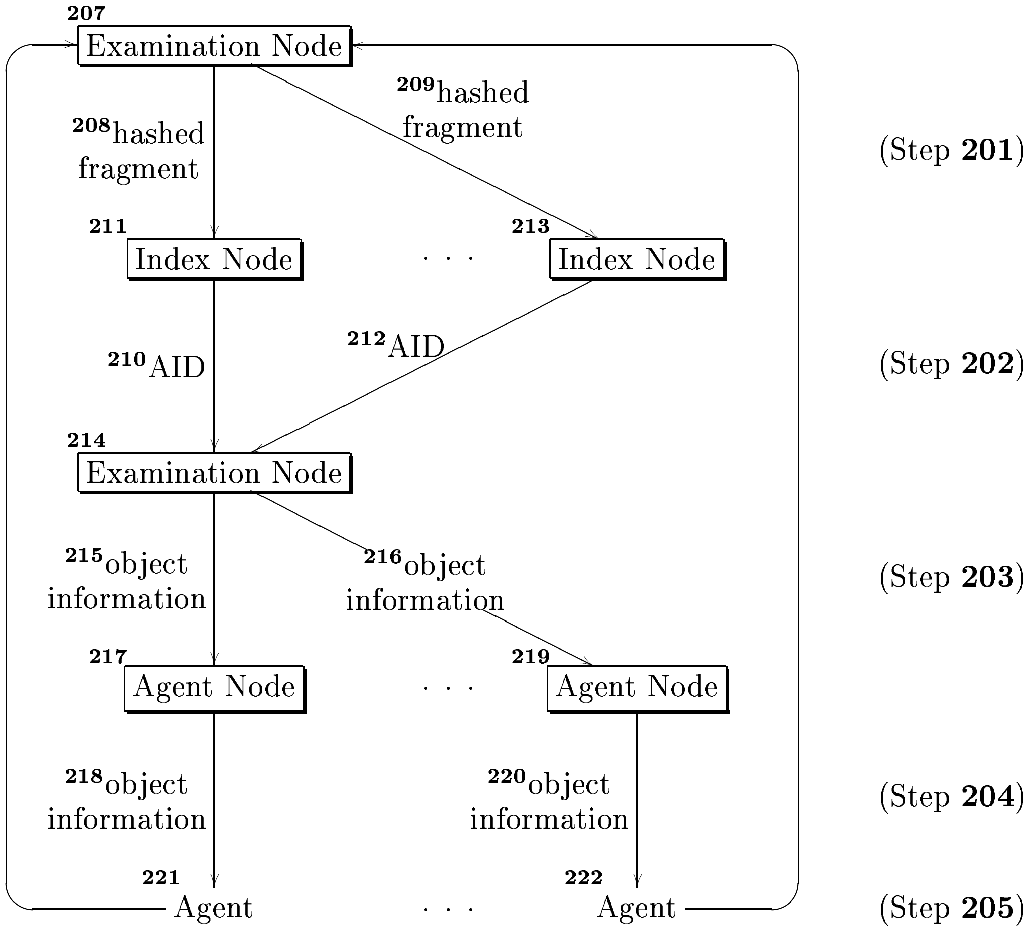
207
Examination Node
208 hashed fragment
209 hashed fragment
(Step 201)
211
Index Node
. . .
213
Index Node
210 AID
212 AID
(Step 202)
214
Examination Node
215 object information
216 object information
(Step 203)
217
Agent Node
. . .
219
Agent Node
218 object information
220 object information
(Step 204)
221
Agent
. . .
222
Agent
(Step 205)

FIG. 2

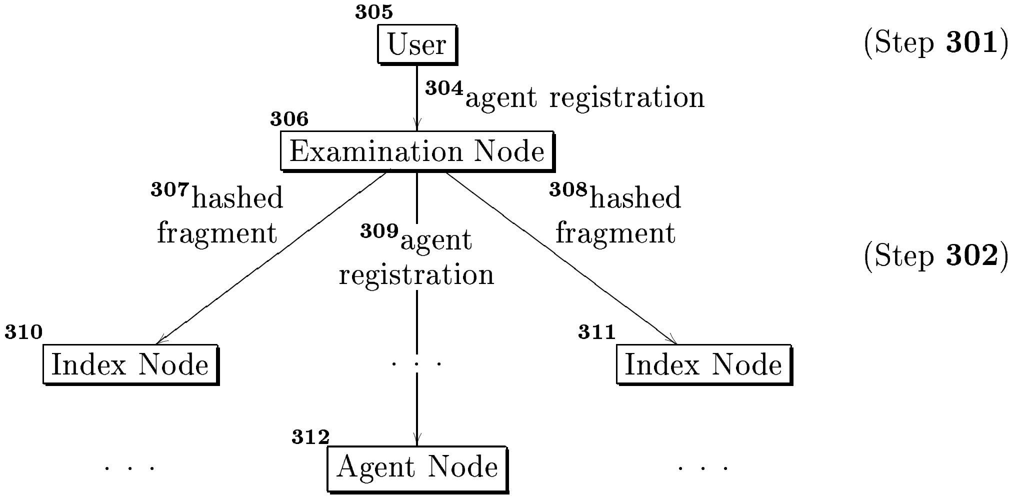
305
User
(Step 301)
304 agent registration
306
Examination Node
307 hashed fragment
309 agent registration
308 hashed fragment
(Step 302)
310
Index Node
. . .
311
Index Node
312
Agent Node
. . .
. . .

FIG. 3

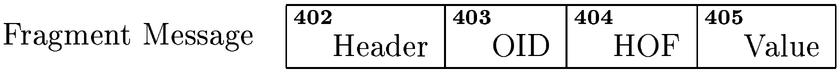

FIG. **4a**

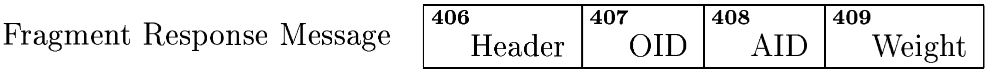

FIG. **4b**

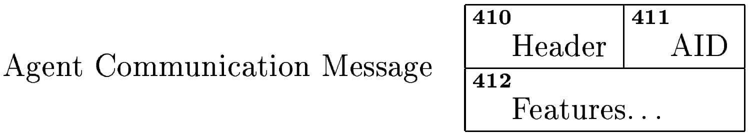

FIG. **4c**

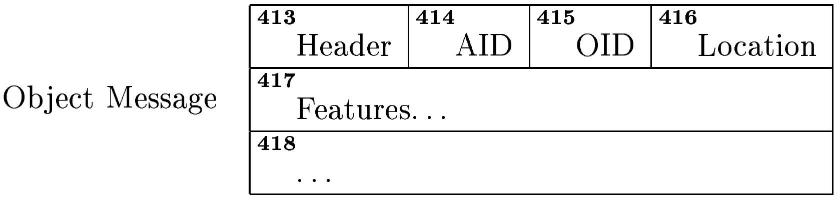

FIG. **4d**

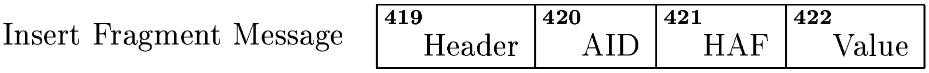

FIG. **4e**

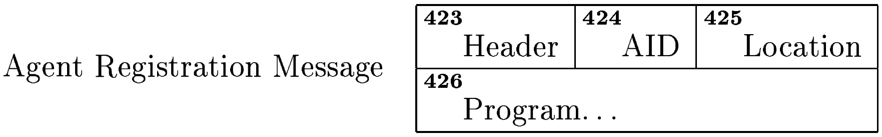

FIG. **4f**