



(51) International Patent Classification:

G06F 7/00 (2006.01) *G10L 15/22* (2006.01)
G06F 17/30 (2006.01)

(21) International Application Number:

PCT/IB2016/050593

(22) International Filing Date:

5 February 2016 (05.02.2016)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

1643/CHE/2015 30 March 2015 (30.03.2015) IN

(71) Applicant: YOKOGAWA ELECTRIC CORPORATION [JP/JP]; 9-32, Naka-cho 2-chome, Musashino-shi, Tokyo 1808750 (JP).

(72) Inventor; and

(71) Applicant (for US only): PENUMATCHA, Vijaya Rama Raju [IN/IN]; c/o Yokogawa IA Technologies India Private Limited, Cessna Business Park Umiya Business Bay, Tower 1 Outer Ring Road, Kadubeesanahalli Belandur Post, Bangalore 560103 (IN).

(74) Agents: TANAI, Sumio et al.; 1-9-2, Marunouchi, Chiyoda-ku, Tokyo 1006620 (JP).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR MACHINE BASED PROCESSING OF NATURAL LANGUAGE INPUT

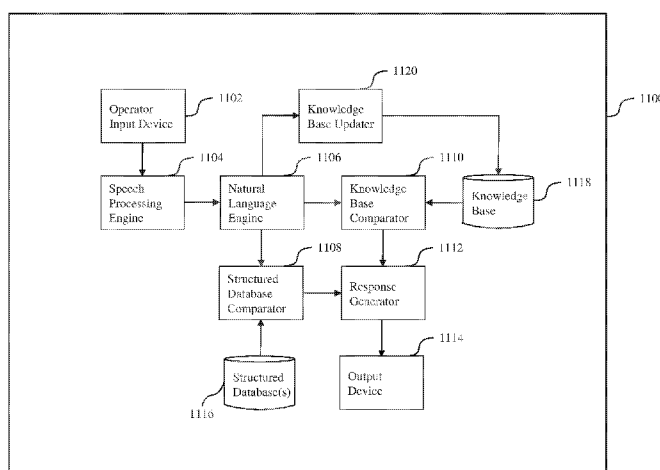


FIGURE 11

(57) Abstract: The invention provides methods, systems and computer program products for machine based processing of natural language input received from an operator within an industrial plant. The invention comprises receiving an operator input from the operator and classifying the received operator input as one of a query and a statement. Responsive to classification of the received operator input as a statement, the method proceeds to extract data from the received operator input, and to update a knowledge base by storing the extracted data as a set of interrelated concept instances within said knowledge base. Responsive to classification of the received operator input as a query, the method proceeds to search at least one of a structured database and the knowledge base for data that matches one or more query parameters extracted from, the received operator input. A query response based on a returned search result may thereafter be presented to the operator.

Methods, Systems and Computer Program Products
for Machine Based Processing of Natural Language Input

Field of the Invention

[001] The present invention relates to generation of responses to natural language queries. The invention includes methods, apparatuses and computer program products for receiving natural language inputs or queries in the form of speech or text input, from a user or operator within an industrial plant, and for processing received natural language inputs or queries using one or both of databases that provide logical storage for structured data, and knowledge bases that are based on ontological models and that provide logical storage for unstructured data.

Background

[002] Information related to events and operations within an industrial plant is stored or embedded across a number of different data sources within the plant. Such data may include (i) structured data – e.g. data stored in plant historian databases, incident management databases, work permit databases, batch databases etc. or (ii) unstructured data – e.g. remarks annexed by a user or operator in a report or shift record.

[003] Briefly, structured data refers to information with a high degree of organization, such that logical storage in a relational database is seamless, and readily searchable using straightforward keyword based search queries or operations. Unstructured data on the other hand refers to information where the lack of structure or organization makes compilation, comprehension and logical storage a time and energy consuming task, and which does not permit for straightforward keyword based search and retrieval operations.

[004] Known solutions for retrieval of information from structured data repositories rely on one or more applications or interfaces for effecting logical storage and retrieval of the structured data. The processes for storing data and developing applications or interfaces for retrieval requires knowledge of query languages, and are correspondingly time and cost intensive. As a consequence, development of applications for storage and retrieval of structured data is typically restricted to cases involving data that is accessed regularly. Further, a user or operator seeking to retrieve information from structured

databases is required to input data queries in a input format (depending on requirements of the applicable query language) prescribed by the storage and retrieval application or interface - which impacts ease of access and flexibility of the search or retrieval process.

[005] Use of unstructured data presents additional barriers to data search and retrieval. A primary reason is that unstructured data suffers from context related ambiguities – wherein data input or search terms may be interpreted in multiple different ways, depending on context that is implied by adjacent terms. As a consequence, keyword searches that may otherwise be effective in the case of structured data sources are often of limited effectiveness in the case of unstructured data.

[006] There is accordingly a need for enabling data storage and retrieval based on natural language inputs or queries raised by a user or operator, which permit (i) logical storage of natural language inputs or other unstructured data received from a user or operator, in a knowledge base and (ii) generation of meaningful responses to natural language queries received from a user or operator, based on information retrieved either from one or more structured databases or from a knowledge base that has been used to store natural language inputs or other unstructured data.

Summary

[007] The invention provides methods, systems and computer program products for machine based processing of natural language input received from an operator within an industrial plant.

[008] The invention comprises a method for machine based processing of natural language input received from an operator within an industrial plant. The method comprises receiving an operator input from the operator and classifying the received operator input as one of a query and a statement. Responsive to classification of the received operator input as a statement, the method proceeds to extract data from the received operator input, and to update a knowledge base by storing the extracted data as a set of interrelated concept instances within said knowledge base. Responsive to classification of the received operator input as a query, the method proceeds to search at least one of a structured database and the knowledge base for data that matches one or more query parameters extracted from the received operator input. A query response based on a returned search result may thereafter be presented to the operator.

[0009] The step of searching at least one of the structured database and the knowledge base may include generating a first set of query parameters based on the received operator input, and performing search of the structured database based on the generated first set of query parameters. Responsive to receiving a null result from search of the structured database, the method may proceed to generate a second set of query parameters based on the received operator input, and thereafter perform search of the knowledge base based on the generated second set of query parameters.

[0010] A query string may be generated based on the first set of query parameters. Further, the step of performing search of the structured database may comprise ascertaining whether the query string matches any predefined query pattern within a set of predefined query patterns. Responsive to identification of a matching predefined query pattern, the method may proceed to execute a set of query language instructions associated with the identified matching predefined query pattern.

[0011] The set of predefined query patterns may comprise AIML patterns. Further, each AIML pattern may have a corresponding AIML template associated therewith. In an embodiment, at least one of said AIML templates comprises a set of query language instructions encapsulated therewithin.

[0012] In an embodiment, the structured database may be selected from among a plurality of structured databases. The selection may be based on at least one of the extracted one or more query parameters, or on the query language instructions associated with the identified matching predefined query pattern.

[0013] In an embodiment of the method, an ontological query model may be generated based on the second set of query parameters. Thereafter, searching the knowledge base may include ascertaining whether the ontological query model matches any part of the knowledge base. Responsive to identification of a match within the knowledge base, the method may include extracting data from the identified matched portion of the knowledge base, and returning a search result comprising data extracted from the knowledge base. Identifying a match between the ontological query model and the knowledge base may in an embodiment be determined based on graph pattern matching.

[0014] In a method embodiment, the knowledge base may be implemented based on an ontological model having interrelated concept classes corresponding to one or more (and in an embodiment, all) of (i) one or more objects, equipment or components within an industrial plant (ii) observed events within said industrial plant (iii) causative factors related to observed events (iv) operator inputs or operator requests in response to either of an observed event or a determined causative factor (v) any action executed in response to a operator input or operator request and (vi) temporal data.

[0015] The method of the present invention may additionally include receiving operator input in the form of speech signals. The received operator input may be subjected to speech to text processing and natural language processing prior to classification as one of a query and a statement.

[0016] The invention may additionally comprise a system configured for machine based processing of natural language input received from an operator within an industrial plant. The system may include at least one processor, a data repository system, an operator input device, a natural language engine, a knowledge base updater, and at least one comparator.

[0017] The data repository system of the present invention may comprise at least one structured database and at least one knowledge base. The operator input device may be configured for receiving an operator input from the operator. The natural language engine may be configured for processing the received operator input and classifying said received operator input as one of a query and a statement.

[0018] The knowledge base updater of the system may be configured to respond to classification of the received operator input as a statement by extracting data from the received operator input, and updating a knowledge base by storing the extracted data as a set of interrelated concept instances within said knowledge base. The at least one comparator may be configured to respond to classification of the received operator input as a query by searching at least one of the at least one structured database and the knowledge base for data matching one or more query parameters extracted from the received operator input. The comparator may thereafter present to the operator, a query response based on a returned search result.

[0019] The system may be configured such that the at least one comparator comprises a structured database comparator configured to generate a first set of query parameters based on the received operator input. The structured database comparator may perform search of at least one structured database based on the generated first set of query parameters. The at least one comparator may also comprise a knowledge base comparator configured to respond to a null result arising from search of the structured database by generating a second set of query parameters based on the received operator input, and performing search of the knowledge base, based on the generated second set of query parameters.

[0020] The structured database comparator of the system may be configured to generate a query string based on the first set of query parameters, and to perform search of the structured database by ascertaining whether the query string matches any predefined query pattern within a set of predefined query patterns. Responsive to identification of a matching predefined query pattern, the structured database comparator may execute a set of query language instructions associated with the matching predefined query pattern.

[0021] In an embodiment of the system, the set of predefined query patterns may comprise AIML patterns. Each of said AIML patterns may include a corresponding AIML template associated therewith and at least one of said AIML templates may have a set of query language instructions encapsulated therewithin.

[0022] The knowledge base comparator of the system may be configured to generate an ontological query model based on the second set of query parameters. The knowledge base comparator may perform search of the knowledge base by ascertaining whether the ontological query model matches any part of the knowledge base. Responsive to identification of a match within the knowledge base, the knowledge base comparator may extract data from the matched portion of the knowledge base, and return a search result comprising data extracted from the knowledge base.

[0023] In an embodiment of the system, the knowledge base may be implemented based on an ontological model having interrelated concept classes corresponding to one or more (and in an embodiment, all) of (i) one or more objects, equipment or components within an industrial plant (ii) observed events or effects or events within said industrial plant (iii) causative factors related to

observed events (iv) operator inputs or operator requests in response to an observed event or a determined causative factor (v) any action executed in response to an operator input or operator request and (vi) temporal data.

[0024] The knowledge base comparator of the system may be configured to identify matches between the ontological query model and the knowledge base, based on graph pattern matching.

[0025] The system embodiments described above may additionally include a speech processor configured to receive speech signals from the operator input device, perform speech to text conversion on the received speech signals, and communicate the converted text to the natural language engine.

[0026] The invention may additionally include a computer program product for machine based processing of natural language input received from an operator within an industrial plant, which computer program product comprises computer readable instructions for implementing method embodiments of the present invention.

Brief Description of the Accompanying Drawings

[0027] **Figure 1** illustrates steps typically encountered during natural language processing.

[0028] **Figure 2** illustrates a data repository system.

[0029] **Figure 3** illustrates an exemplary ontology.

[0030] **Figure 4A** illustrates an ontological model

[0031] **Figure 4B** illustrates an instance of a knowledge base generated based on the ontological model of **Figure 4A**.

[0032] **Figures 4C** and **4D** respectively illustrate data structures that may be used to implement the ontological model and knowledge base of **Figures 4A** and **4B**.

[0033] **Figures 5** to **9** are flowcharts illustrating methods in accordance with the present invention.

[0034] **Figures 10A and 10B** illustrates an exemplary ontological query model.

[0035] **Figures 11 and 12** depict systems configured in accordance with the teachings of the present invention.

Detailed Description

[0036] For the purposes of the present invention, the term “operator input” may be understood to include one or both of (i) speech inputs received from a user or operator or a supervisory operator, at a speech input device (such as a microphone or other acoustic sensor) or (ii) text inputs received from a user or operator at a user interface device (such as a computer terminal, mobile phone, tablet, personal digital assistant etc) or (iii) inputs entered into reports by a user or operator or a supervisory operator.

[0037] For the purposes of the present invention, it will be understood that text inputs received directly from a user or an operator, or text received as an output of the speech-to-text conversion comprise natural language inputs, and may be subjected to natural language processing (NLP) steps, which are briefly discussed in **Figure 1**.

[0038] Steps **102** to **118** of **Figure 1** illustrate steps involved in natural language processing of speech inputs. In the event an operator input comprises text inputs, steps **102** and **104** may be omitted and the method may commence directly from step **106**.

[0039] In the event an operator input comprises speech input(s), step **102** comprises extracting a set of acoustic vectors from the input speech. Thereafter step **104** uses speech-to-text conversion, including for example Hidden Markov Models (HMMs), and appropriate grammars and dictionaries, for converting the speech representing the operator input to a set of human readable text characters.

[0040] Step **106** comprises sentence segmentation and tokenization. Sentence segmentation comprises delimiting the human readable text characters (that is either received from step **104** or directly received as text input from an operator) into one or more sentences. Tokenization comprises segmenting each sentence into a list of tokens. For the purposes of the present invention, a token may

be understood as a sequence of characters that are grouped together as a useful semantic unit for processing. Within a sentence, tokens typically correspond to discrete words or terms. In certain situations, in addition to grouping characters together, tokenization may also include removing or omitting certain characters (such as for example, certain punctuation characters or delimiters).

[0041] Step **108** comprises lexical (or morphological) analysis of sentences — where each word is tagged with its corresponding part of speech. Step **110** comprises syntactical analysis or parsing, which comprises assigning a syntactic structure or a parse tree, to a given natural language sentence. Semantic analysis at step **112** comprises translating a syntactic structure of a sentence into a semantic representation that is a precise and unambiguous representation of the meaning expressed by the sentence. Semantic analysis is achieved based on the knowledge about the structure of words and sentences — with a view to precisely stipulate the meaning of words, phrases, sentences and texts, and subsequently also their purpose and consequences.

[0042] Step **114** comprises topic identification — which ascertains the topic or subject to which a sentence or set of sentences relates. Topic identification is based on one or more topic models that are used to analyze text and classify statements or queries within the text to corresponding topics. Step **116** comprises intent identification — which seeks to ascertain the purpose or intention behind a statement or query. Step **118** comprises Named Entity Recognition which tags or labels names of things (e.g. names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages) that are found within a sentence.

[0043] By implementing some or all of natural language processing steps **102** to **118** of **Figure 1**, a disambiguated logically structured and tagged representation of human readable text may be generated, which can be used for machine processing of natural language statements or queries received from a user or operator.

[0044] In addition to natural language processing, the present invention also relies on storage and retrieval of structured data and unstructured data for the purposes of processing natural language statements or queries received from a user or operator. For the purposes of the invention, “unstructured data” refers to data that is received in the form of free text or natural language text, interpretation whereof relies on interpretation of individual terms as well as on context associated with

individual terms. For the purposes of the invention “structured data” means data that is stored or received in a predefined format which may have one or more input rules and interpretation whereof is independent of context. Structured data may be obtained in any number of ways that are well understood in the art – for example, data acquisition through sensors (for example in the case of a historian database), data that is stored while transactions are executed (for example, in database consisting of order details corresponding to a product) or data that is entered manually during execution of workflows (for example, information recordal in the course of lab analysis of a product). In an embodiment of the invention, structured data may comprise data that is already stored in a system database, and which is not a product of operator input.

[0045] The invention achieves logical storage and retrieval of both structured and unstructured data using a data repository system of the type described in **Figure 2**. **Figure 2** illustrates a data repository system comprising one or more repositories **202a**, **202b**, **202c** and **202d** configured for storing structured data, as well as a repository **204** configured for storing unstructured data.

[0046] Repositories **202a** to **202d** may comprise databases that are configured to logically store structured data. Said repositories are based on database schemas configured to store data that is received in predefined formats. Common examples of structured databases that are likely to be encountered within industrial plants include archival databases (e.g. historian databases), enterprise resource planning databases (ERP databases), laboratory information management system databases (LIMS databases), databases for storing data relevant to monitoring and resolution of service disruptions (Incident Management databases), databases for storing data relevant to access control permissions (Work Permit System databases) and databases for storing data relevant to plant management information systems (Plant MIS databases).

[0047] For the purposes of the present invention, repository **204** for storing unstructured data may be implemented by way of a knowledge base that is generated based on one or more ontological models.

[0048] An ontology comprises a set of interconnected concepts and relationships within a particular domain. Concepts may be understood as classes of instances / things, while relationships signify the relation or linkages between such classes. Ontologies are used and adopted as a way to understand,

process and apply domain specific knowledge and information. By way of example, an ontology for traffic may include concepts such as vehicles, streets lights, geographical locations, time of day, and traffic volume. The traffic ontology also includes relationships between concepts. For example, a specific vehicle may be “halted” at a particular street light that is in turn “located within” a specified geographical location. Each instance of a concept may have different attributes – for example, vehicles may be classified as cars, buses, two-wheeled vehicles etc. Attribute of concept instances may be defined in terms of corresponding attribute information – for example, in the case of vehicles, each concept instance may be defined in terms of attribute information such as size, make, model, number of wheels, public or private transport etc.

[0049] Figure 3 illustrates an exemplary ontology comprising concepts 1 to 5 that are interconnected by virtue of relationships 1 to 5.

[0050] An ontology is represented by an ontology model – which is a structure for representing an ontology. Ontology models may *inter alia* be represented using a graph structure such as a tree structure, where concepts are represented by nodes and relationships between concepts are represented by edges. In implementing an ontology, node attributes may be used to define properties of each concept instance.

[0051] When an ontology is implemented within an actual data structure, and used to store instances of specific ontology elements, the implemented data structure comprises a knowledge base (i.e. an ontological model based knowledge base). For the purposes of the present invention, repository 204 (which is configured for logical storage and retrieval of unstructured data) comprises a knowledge base of the kind described above.

[0052] For the purposes of the invention, when referring to a knowledge base that has been implemented based on an ontological model, each concept from the ontological model is implemented as a concept class, and specific concept instances corresponding to a concept class may be stored by storing attribute data specific to said concept instance. Relationships between concept instances may be defined by linking said concept instances. In an embodiment, said relationships between concept instances may be established by linking using primary keys, secondary keys, pointers or other links within the data structure being used to implement the ontological model.

[0053] In an embodiment of the invention, the ontological model which forms the basis for implementation of the knowledge base may comprise a model specifically designed or configured to accommodate or store data corresponding to concepts and relationships that are likely to be encountered within the intended domain of implementation.

[0054] **Figure 4A** provides an exemplary embodiment of an ontological model configured to accommodate concepts and relationships within data obtained within an industrial plant. Nodes within the ontological model are configured to accommodate data corresponding to the “Equipment Name”, “Rated Output”, “Equipment Class”, “Manufacturer”, “Operator ID” and “date” concept classes. Edges within the ontological model establish (i) a “generates” relation between “Equipment Name” and “Rated Output” nodes (ii) a “classified as” relation between “Equipment Name” and “Equipment Class” nodes (iii) a “preferred vendor of spare parts” relation between “Equipment Class” and “Manufacturer” nodes (iv) a “manufactured by” relation between “Equipment Name” and “Manufacturer” nodes, (v) an “operated by” relation between “Equipment Name” and “Operator ID” nodes and (v) a “last serviced on” relation between “Equipment Name” and “Date” nodes. **Figure 4C** illustrates an exemplary table data structure which may be used to implement the ontological model of **Figure 4A**.

[0055] **Figure 4B** provides an exemplary illustration of the manner in which a knowledge base that is implanted based on the ontological model of **Figure 4A** may be used to logically store unstructured data from the industrial plant. **Figure 4B** illustrates logical storage of the following set of unstructured data relating to a particular item of equipment i.e. Steam Turbine No. 3:

“Steam Turbine No. 3 has a maximum output rating of 40 KW, and is classified as heavy rotating machinery. The steam turbine is presently operated by operator 432, and was last serviced on March 3, 2015. The machine has been manufactured by Heavy Machines Inc. - who are also designated preferred vendors for obtaining spares for heavy rotating machinery.”

[0056] By parsing the above unstructured text, and using natural language processing techniques for identifying concepts and relationships therebetween, it is possible to logically store data describing said unstructured text within the knowledge base illustrated in **Figure 4B**. **Figure 4D** illustrates the

exemplary table data structure of **Figure 4C** which has been populated with the specific interrelated concept instances of **Figure 4B**.

[0057] **Figure 5** illustrates a method embodiment of the present invention. The method relies on a data repository system of the kind illustrated in **Figure 2**, which comprises at least one repository configured for storing logical data, and at least one knowledge base that is implemented based on an ontological model, and configured for enabling storage and logical retrieval of unstructured data.

[0058] Step **502** comprises receiving an operator input, which operator input may comprise speech input or text input. In the event the operator input comprises speech input, the method would include the additional steps (not illustrated) of extracting acoustic vectors from the input speech, and applying speech to text conversion for generating a set of human readable text characters, before proceeding to step **504**. If on the other hand the operator input comprises text inputs, the additional steps may be omitted and the method proceeds directly to step **504**.

[0059] Step **504** comprises application of natural language processing techniques to the generated or received set of text characters -- which natural language processing techniques may include some or all of the steps described in connection with **Figure 1**.

[0060] Step **506** comprises determining, based on the output of the natural language processing techniques, whether the operator input comprises a statement or a query.

[0061] In the event the operator input is determined to comprise a statement, the method proceeds to step **508**, wherein concepts and relationships extracted from the operator input are used to update the knowledge base within the data repository system.

[0062] Alternatively, in the event the operator input is determined to comprise a query, the method instead proceeds to step **510**, at which, at least one of a structured database(s) and a knowledge base are searched for data that matches a query string or query parameters extracted from said query. At step **510**, one or more results returned by the search operations is used to generate a response to the query -- which response may be presented to the operator either in the form of a displayed text response, or in the form of speech output generated by text-to-speech operations.

[0063] It would be understood that the method of Figure 5 permits for an operator input to be classified as a statement and dealt with in accordance with step 508, while another operator input may be classified as a query and dealt with in accordance with steps 510 and 512. The method of Figure 5 may therefore alternate between steps 508 and steps 510 and 512, depending on the contents of received operator input.

[0064] Figure 6 illustrates an exemplary embodiment of the method more generally discussed in connection with method step 508 of Figure 5, wherein concepts and relationships extracted from the operator input may be used to update the knowledge base within the data repository system. At step 602, data corresponding to concepts and relationships are extracted from the received operator input (i.e. from the set of text characters received at or generated based on step 502 of Figure 5). At step 604, the extracted concept and relationship data is used to populate the knowledge base by storing concept data, as concept instances within the knowledge base. Extracted relationships between specific concept instances may simultaneously be reflected within the knowledge base by linking interrelated concept instances using one or more keys (such primary keys or secondary keys), pointers or other links. Steps 602 and 604 accordingly enable logical storage of unstructured data that may be received by way of natural language statements made by a system operator -- which data may subsequently be subjected to logical search and retrieval operations executed on the knowledge base.

[0065] Figure 7 illustrates an exemplary embodiment of the method more generally discussed in connection with method step 510 of Figure 5, wherein at least one of a structured database(s) and a knowledge base are searched by seeking a match for a query string or query parameters extracted from said query.

[0066] At step 702, the method generates a first set of query parameters (or a first query string) based on output from natural language processing steps that have been applied (at step 504 of Figure 5) to the received operator input. Step 704 checks to ascertain whether the first set of query parameters (or the first query string) return a valid search result from one or more structured databases that form a part of the data repository system (see Figure 2). Responsive to the first set of query parameters (or first query string) returning valid search results from the one or more structured databases, the method proceeds at step 710 to generate a response to the operator input query based on returned one or more search results.

[0067] Alternatively, responsive to the first set of query parameters (or first query string) not returning valid search results (hereinafter a “null result”) from the one or more structured databases (, the method proceeds to step 706, which comprises generating a second set of query parameters (or second query string) based on output from natural language processing steps that have been applied (at step 504 of Figure 5) to the received operator input.

[0068] Step 708 thereafter checks to ascertain whether the second set of query parameters (or second query string) return a valid search result from the knowledge databases that forms a part of the data repository system (see Figure 2). Responsive to the second set of query parameters (or second query string) returning valid search results from the knowledge base, the method proceeds at step 710 to generate a response to the operator input query based on the returned search results.

[0069] In an embodiment of the method illustrated in Figure 7, the knowledge base on which the searches of step 708 are executed, may be implemented based on an ontological model having interrelated concept classes corresponding to (i) one or more objects, equipment or components within an industrial plant (ii) observed events or effects or events within said industrial plant (iii) causative factors related to observed events (iv) operator inputs or operator requests in response to an observed event or a determined causative factor (v) any action executed in response to the operator input or operator request and (vi) temporal data corresponding to any or all of the above.

[0070] Figure 8 describes the specific method steps involved in ascertaining whether a first set of query parameters return valid search results from the one or more structured databases (step 704 of Figure 7). Likewise, Figure 9 further describes specific method steps involved in ascertaining whether a second set of query parameters return valid search results from the knowledge base (step 708 of Figure 7).

[0071] In Figure 8, Step 802 comprises generating a query string based on a first set of query parameters extracted from output of the natural language processing steps that have been applied (at step 504 of Figure 5) to the received operator input. The query string may be generated based on one or more predefined rules for generation of a query string. In an embodiment of the method, the query string may comprise the entire string of text characters extracted as output of the natural language processing steps. In another embodiment of the method, the query string may comprise one or more

of topics, intents and /or named entities identified in the course of the executed natural language processing steps.

[0072] Step 804 ascertains whether the query string matches any predefined query pattern within a predefined set of query patterns. In an embodiment of the invention, the predefined set of query patterns comprises a plurality of query patterns generated in Artificial Intelligence Markup Language (AIML) syntax, and step 804 ascertains whether the query string matches any of the predefined AIML query patterns within said predefined set of query patterns.

[0073] Responsive to the query string matching a predefined query pattern, step 806 comprises executing a set of database queries instructions which correspond to the matching predefined query pattern. In the embodiment where the predefined query pattern is an AIML pattern generated using AIML syntax, step 806 comprises executing a set of database query instructions encapsulated within an AIML template corresponding to said AIML pattern. In a yet more particular embodiment of step 804, the execution of the set of database query instructions encapsulated within the AIML template may rely on one or more query parameters extracted from the query string. For example, selection of a structured database (from among a plurality of structured databases within a data repository system) may be based on one or more topics, intents or named entities extracted from the query string.

[0074] It would be understood that the set of database query instructions encapsulated within an AIML template may comprise a single database query instruction or a plurality of database query instructions. Further the database query instructions may comprise database query instructions in any query language capable of querying one or more structured databases within the data repository system. Without intending to limit the scope of the present invention, exemplary query languages that may be used to implement the present invention include SQL, .QL, CQL, CQLF, COFL, DMF, Datalog, F-logic, FQL, OQL, OCL etc.

[0075] At step 808, the method generates a response to the query string, based on results returned by the executed set of database query instructions -- which response may be presented (e.g. in text form or speech form) to the operator.

[0076] To better understand the method steps involved in **Figure 8**, an example involving implementation of AIML syntax may be considered. An operator may, by way of speech or text input

check “*Have any recent incidents been observed in connection with the distillation column*”. Applying natural language processing steps, the invention would identify the operator input as a query, and based on identified topics, intents and/or named entities, may extract the following information for search through structured databases:

- Query String: Last Similar Incident
- Equipment: The distillation column
- Equipment type: Distillation column
- Time stamp: Unknown

[0077] Responsive to generation of the query string “Last Similar Incident”, step **804** of **Figure 8** searches for a matching AIML query pattern. Given that the topic of the query string involves Incidents, in an embodiment, step **804** may search for a matching AIML query pattern only among query patterns that are associated only with the Incident Database. In the example, the following matching AIML pattern (and corresponding AIML template) is found:

```

<category>
<pattern> Last Similar Incident </pattern>
<template>
    <extract>
        <executedatabase type = “Incident Database”>
        <command try = “1”>
            Similar Incident
        <context>
            <name datatype = “text”/>
            <equipmentclass datatype = “text”/>
        <equipmentname datatype = “text”/>
        <equipmenttype datatype = “text”/>
        </context>
    </command>
    <command try = “2”>
        Similar Incident

```

```

    <context>
        <name datatype = "text"/>
        <incidenttype datatype = "text"/>
    <equipmentname datatype = "text"/>
    </context>
</command>
    </executedatabase >
</extract>
<responseText>
    The Similar Incident was found by <that> <ctxt> value0 </ctxt></that> at
    <that><ctxt>        value1        </ctxt></that><response>        value2
    </response></that>
</responseText>
</template>
</category>

```

[0078] The query language instructions encapsulated within the AIML template which corresponds to the matched AIML pattern (i.e. <pattern> Last Similar Incident </pattern>) are executed once within the Incident database, including as a search parameter, equipment name as “the distillation column”. If no result is obtained, a wider set of search instructions is executed at try 2 by eliminating the equipment name as a search parameter. Assuming a result is obtained, the values at positions 0, 1 and 2 of the returned search results may be copied into placeholder locations value0, value1 and value 2 of the defined response text, which response text is thereafter further processed for presentation to the querying operator. As will be understood from the above example, in certain embodiments, the predefined set of AIML patterns and corresponding templates may additionally encapsulate a response format into which search results that are retrieved from a structured database may be incorporated for presentation to the operator.

[0079] Figure 9 provides further detail regarding step 708 of Figure 7, which ascertains whether a second set of query parameters returns valid search results from within a knowledge base.

[0080] Step 902 of **Figure 9** comprises generating an ontological query model based on a second set of query parameters extracted from output of the natural language processing steps that have been applied (at step 504 of **Figure 5**) to operator input. The ontological query model is an ontology model comprising concept instances and relations extracted from a query string or set of query parameters. In an embodiment of the invention, an ontological query model may include a set of data instances corresponding to (i) at least two interrelated concept instances or (ii) at least one concept instance and one corresponding relationship. The ontological query model may additionally include a concept instance placeholder which identifies a concept class (for which a corresponding data value is presently unknown or is sought to be found within the knowledge base) that is related to the set of data instances that have been used to construct the ontological query model.

[0081] At step 904, the method ascertains whether the ontology query model matches or maps on to any set of concept instances and relationships that are already stored within the knowledge base. The step of matching the ontological query model with data within the knowledge base can be achieved in any number of ways including using pattern matching, graph pattern matching or query languages that implement graph pattern matching (e.g. SPARQL).

[0082] In response to an identified match between the ontology query model and a set of concept instances and relationships stored within the knowledge base, step 906 generates a response based on data extracted from the matched set of concept instances and relationships that has been identified within the knowledge base. In an embodiment of the invention, responsive to an ontology query model matching a set of concept instances and relationships stored within the knowledge base, step 906 extracts data corresponding to a concept instance within the knowledge base, which concept instance corresponds in terms of both concept class and relationships to a concept class placeholder within the ontological query model, and uses the extracted data to generate a response for presentation to an operator.

[0083] **Figure 10** illustrates an exemplary ontology query model that may be generated in accordance with the method described above in connection with **Figure 9**. **Figure 10** specifically illustrates an ontology query model that has been generated in response to a query relating to the industrial plant specific knowledge base that has been previously discussed in connection with **Figure 4B**.

[0084] Figures 10A and 10B illustrate an exemplary ontology query model generated based on the query “What is the rated output of Steam Turbine No. 3”. The ontology query model of Figure 10A includes a concept instance 1002 having data value “Steam Turbine No. 3”, which falls within concept class 1004 of the type “Equipment Name”, and has a “generates” relationship corresponding to said concept instance 1002. The ontology query model additionally includes a concept class placeholder 1008a which falls within concept class 1006 of the type “Rated Output”. In executing the method of Figure 9, the ontology query model of Figure 10A may be pattern matched against the knowledge base illustrated in Figure 4B. By virtue of pattern matching, the concept class 1006 of type “Rated Output” within the ontological query model is matched against the corresponding “Rated Output” concept class of the knowledge base. The pattern matching further establishes that the relevant concept instance of this “Rated Output” concept class is a concept instance of the class “Equipment Name” and which has the specific data value “Steam Turbine No. 3”. The invention accordingly determines (as illustrated in Figure 10B) that the relevant concept instance from the knowledge base (i.e. which most closely provides a match for the concept class placeholder within the ontological query model) is the concept instance 1008b having concept class “Rated Output” and having the data value “40 KW”. This data value may be extracted from the knowledge base and may be used to generate and present a response to the operator query for which the ontological query model of Figure 10A was generated.

[0085] Figure 11 illustrates a system 1100 in accordance the present invention, which may be configured to implement one or more of the above discussed methods for generating responses to natural language queries received by way of operator input. The system 1100 comprises a speech input 1102, a speech processing engine 1104, natural language engine 1106, structured database comparator 1108, knowledge base comparator 1110, response generator 1112, operator interface 1114, structured database(s) 1116, knowledge base 1118 and knowledge base updater 1120.

[0086] Operator input device 1102 may comprise any device configured to receive speech or text inputs from an operator. In an embodiment, operator input device 1102 may comprise any input device capable of receiving speech or sound based input signals, including by way of example a microphone, acoustic sensor or other device capable of receiving receiving sound waves and generating electromagnetic signals representing received sound waves. In another embodiment, operator input device 1102 may comprise any text input device, including by way of example a keyboard, keypad,

touch screen interface or other peripheral device configured to enable an operator to input text characters.

[0087] Speech processing engine **1104** comprises a processor implemented engine configured for extracting acoustic vectors from speech signals and to implement speech-to-text conversion based on the extracted acoustic vectors – for converting received speech input to a set of readable text characters.

[0088] Natural language engine **1106** comprises a processor implemented engine for implementing one or more of the natural language processing steps described in connection with **Figure 1**. In an embodiment, natural language engine **1106** may be configured to implement one or more of sentence segmentation, tokenization, lexical analysis, syntactical analysis, semantic analysis, topic identification, intent identification and named entity recognition. In an embodiment, natural language engine **1106** may additionally be configured to determine whether an operator input comprises a statement or a query.

[0089] Structured database comparator **1108** comprises a processor implemented comparator configured for extracting a first set of query parameters or query string from output received from natural language engine **1106**, and for determining whether the query parameters or query string return any valid search results based on data within structured database **1116**. In an embodiment, structured database comparator **1108** may be configured to implement method steps discussed above in connection with **Figures 7** and **8**.

[0090] Knowledge base comparator **1110** comprises a processor implemented comparator configured for extracting a second set of query parameters or an ontological query model from output received from natural language engine **1106**, and for determining whether a search based on the second set of query parameters or ontological query model return any valid search results from within knowledge base **1118**. In an embodiment, knowledge base comparator **1110** may be configured to implement method steps discussed above in connection with **Figures 7** and **9**. In a specific embodiment of the invention, knowledge base comparator **1110** may be configured to find and return valid search results from a knowledge base, by generating an ontological query model representative of an operator query and searching the knowledge base for a matching graph structure, using pattern matching or graph pattern matching.

[0091] Response generator **1112** comprises a processor implemented generator for generating a response to an operator query based on search results received from either of structured database comparator **1108** or knowledge base comparator **1110**. Response generator **1112** communicates the generated response to output device **1114**. In an embodiment of the invention, output device may comprise a display for presenting a text based response to an operator. In another embodiment of the invention, output device may comprise a text-to-speech convertor for converting a text based response to a speech signal and implementing playback of said speech signal to an operator.

[0092] Knowledge base updater **1120** comprises a processor implemented updater configured to respond to a determination (by natural language engine **1106**) that a received operator input comprises a statement, by updating knowledge base **1118** to include data corresponding to concept instances and relationships identified within or extracted from said statement. In an embodiment of the invention, knowledge base updater **1120** may be configured to implement method steps from the methods discussed above in connection with **Figures 5 or 6**.

[0093] In an embodiment of the system illustrated in **Figure 11**, knowledge base **1118** may be implemented based on an ontological model having interrelated concept classes corresponding to (i) one or more objects, equipment or components within an industrial plant (ii) observed events or effects or events within said industrial plant (iii) causative factors related to observed events (iv) operator inputs or operator requests in response to an observed event or a determined causative factor (v) any action executed in response to the operator input or operator request and (vi) temporal data corresponding to any or all of the above.

[0094] **Figure 12** illustrates an exemplary computing system in which various embodiments of the invention may be implemented.

[0095] The system **1202** comprises at-least one processor **1204** and at-least one memory **1206**. The processor **1204** executes program instructions and may be a real processor. The processor **1204** may also be a virtual processor. The computer system **1202** is not intended to suggest any limitation as to scope of use or functionality of described embodiments. For example, the computer system **1202** may include, but not be limited to, one or more of a general-purpose computer, a programmed microprocessor, a micro-controller, an integrated circuit, and other devices or arrangements of devices

that are capable of implementing the steps that constitute the method of the present invention. Exemplary embodiments of a system **1202** in accordance with the present invention may include one or more servers, desktops, laptops, tablets, smart phones, mobile phones, mobile communication devices, tablets, phablets and personal digital assistants. In an embodiment of the present invention, the memory **1206** may store software for implementing various embodiments of the present invention. The computer system **1202** may have additional components. For example, the computer system **1202** includes one or more communication channels **1208**, one or more input devices **1210**, one or more output devices **1212**, and storage **1214**. An interconnection mechanism (not shown) such as a bus, controller, or network, interconnects the components of the computer system **1202**. In various embodiments of the present invention, operating system software (not shown) provides an operating environment for various softwares executing in the computer system **1202** using a processor **1204**, and manages different functionalities of the components of the computer system **1202**.

[0096] The communication channel(s) **1208** allow communication over a communication medium to various other computing entities. The communication medium provides information such as program instructions, or other data in a communication media. The communication media includes, but not limited to, wired or wireless methodologies implemented with an electrical, optical, RF, infrared, acoustic, microwave, Bluetooth or other transmission media.

[0097] The input device(s) **1210** may include, but is not limited to, a touch screen, a keyboard, mouse, pen, joystick, trackball, a voice device, a scanning device, or any another device that is capable of providing input to the computer system **1202**. In an embodiment of the present invention, the input device(s) **1210** may be a sound card or similar device that accepts audio input in analog or digital form. The output device(s) **1212** may include, but not limited to, a user interface on CRT, LCD, LED display, or any other display associated with any of servers, desktops, laptops, tablets, smart phones, mobile phones, mobile communication devices, tablets, phablets and personal digital assistants, printer, speaker, CD/DVD writer, or any other device that provides output from the computer system **1202**.

[0098] The storage **1214** may include, but not limited to, magnetic disks, magnetic tapes, CD-ROMs, CD-RWs, DVDs, any types of computer memory, magnetic stripes, smart cards, printed barcodes or any other transitory or non-transitory medium which can be used to store information and can be

accessed by the computer system **1202**. In various embodiments of the present invention, the storage **1214** contains program instructions for implementing the described embodiments.

[0099] In an embodiment of the present invention, the computer system **1202** is part of a distributed network.

[00100] The present invention may be implemented in numerous ways including as a system, a method, or a computer program product such as a computer readable storage medium or a computer network wherein programming instructions are communicated from a remote location.

[00101] The present invention may suitably be embodied as a computer program product for use with the computer system **1202**. The method described herein is typically implemented as a computer program product, comprising a set of program instructions which is executed by the computer system **1202** or any other similar device. The set of program instructions may be a series of computer readable codes stored on a tangible medium, such as a computer readable storage medium (storage **1204**), for example, diskette, CD-ROM, ROM, flash drives or hard disk, or transmittable to the computer system **1202**, via a modem or other interface device, over either a tangible medium, including but not limited to optical or analogue communications channel(s) **1208**. The implementation of the invention as a computer program product may be in an intangible form using wireless techniques, including but not limited to microwave, infrared, bluetooth or other transmission techniques. These instructions can be preloaded into a system or recorded on a storage medium such as a CD-ROM, or made available for downloading over a network such as the Internet or a mobile telephone network. The series of computer readable instructions may embody all or part of the functionality previously described herein.

[00102] It will be understood that methods, systems and computer program products in accordance with the present invention provide a reliable and effective solution for natural language processing of operator inputs, and for logically storing and querying structured as well as unstructured data that are encountered within a defined knowledge domain.

[00103] While the exemplary embodiments of the present invention are described and illustrated herein, it will be appreciated that they are merely illustrative. It will be understood by those

skilled in the art that various modifications in form and detail may be made therein without departing from or offending the spirit and scope of the invention as defined by the appended claims.

We claim:

1. A method for machine based processing of natural language input received from an operator within an industrial plant, the method comprising:
receiving an operator input from the operator;
classifying the received operator input as one of a query and a statement;
responsive to classification of the received operator input as a statement:
extracting data from the received operator input; and
updating a knowledge base by storing the extracted data as a set of interrelated concept instances within said knowledge base;
and
responsive to classification of the received operator input as a query:
searching at least one of a structured database and the knowledge base for data that matches one or more query parameters extracted from the received operator input; and
presenting to the operator, a query response based on a returned search result.
2. The method as claimed in claim 1, wherein the step of searching at least one of the structured database and the knowledge base comprises:
generating a first set of query parameters based on the received operator input;
performing search of the structured database based on the generated first set of query parameters;
and
responsive to receiving a null result from search of the structured database:
generating a second set of query parameters based on the received operator input; and
performing search of the knowledge base, based on the generated second set of query parameters.
3. The method as claimed in claim 2, wherein:
a query string is generated based on the first set of query parameters; and
the step of performing search of the structured database comprises:
ascertaining whether the query string matches any predefined query pattern within a set of predefined query patterns; and

responsive to identification of a matching predefined query pattern, executing a set of query language instructions associated with the identified matching predefined query pattern.

4. The method as claimed in claim 3, wherein:

the set of predefined query patterns comprise AIML patterns, each of said AIML patterns having a corresponding AIML template associated therewith; and

at least one of said AIML templates comprising a set of query language instructions encapsulated therewithin.

5. The method as claimed in claim 3, wherein the structured database is selected from among a plurality of structured databases, and wherein said selection is based on at least one of the extracted one or more query parameters, or on the query language instructions associated with the identified matching predefined query pattern.

6. The method as claimed in claim 2, wherein:

an ontological query model is generated based on the second set of query parameters;

and wherein performing search of the knowledge base comprises:

ascertaining whether the ontological query model matches any part of the knowledge base;

responsive to identification of a match within the knowledge base, extracting data from the identified matched portion of the knowledge base; and

returning a search result comprising data extracted from the knowledge base.

7. The method as claimed in claim 1, wherein the knowledge base is implemented based on an ontological model having interrelated concept classes corresponding to all of (i) one or more objects, equipment or components within an industrial plant (ii) observed events within said industrial plant (iii) causative factors related to observed events (iv) operator inputs or operator requests in response to either of an observed event or a determined causative factor (v) any action executed in response to a operator input or operator request and (vi) temporal data.

8. The method as claimed in claim 6, wherein identifying a match between the ontological query model and the knowledge base is determined based on graph pattern matching.

9. The method as claimed in claim 1, wherein the received operator input comprises speech signals, and said received operator input is subjected to speech to text processing and natural language processing prior to classification as one of a query and a statement.

10. A system for machine based processing of natural language input received from an operator within an industrial plant, the system comprising:

at least one processor;

a data repository system comprising at least one structured database and at least one knowledge base;

an operator input device configured for receiving an operator input from the operator;

a natural language engine configured for processing the received operator input and classifying said received operator input as one of a query and a statement;

a knowledge base updater configured to respond to classification of the received operator input as a statement by:

extracting data from the received operator input; and

updating a knowledge base by storing the extracted data as a set of interrelated concept instances within said knowledge base;

and

at least one comparator configured to respond to classification of the received operator input as a query by:

searching at least one of the at least one structured database and the knowledge base for data matching one or more query parameters extracted from the received operator input; and
presenting to the operator, a query response based on a returned search result.

11. The system as claimed in claim 10, wherein the at least one comparator comprises:

a structured database comparator configured to:

generate a first set of query parameters based on the received operator input; and

perform search of at least one structured database based on the generated first set of query parameters;

and

a knowledge base comparator configured, to respond to a null result arising from search of the structured database, by:

generating a second set of query parameters based on the received operator input; and
performing search of the knowledge base, based on the generated second set of query parameters.

12. The system as claimed in claim 11, wherein the structured database comparator is configured to:

generate a query string based on the first set of query parameters; and
perform search of the structured database including by:

ascertaining whether the query string matches any predefined query pattern within a set of predefined query patterns; and

responsive to identification of a matching predefined query pattern, executing a set of query language instructions associated with the matching predefined query pattern.

13. The system as claimed in claim 12, wherein:

the set of predefined query patterns comprise AIML patterns, each of said AIML patterns having a corresponding AIML template associated therewith; and

at least one of said AIML templates comprises a set of query language instructions encapsulated therewithin.

14. The system as claimed in claim 12, wherein the structured database comparator is configured to select the structured databases from among a plurality of structured database, and wherein said selection is based on at least one of the extracted one or more query parameters or on at least one query language instruction associated with the matching predefined query pattern.

15. The system as claimed in claim 11, wherein the knowledge base comparator is configured to:
generate an ontological query model based on the second set of query parameters; and
perform search of the knowledge base including by:

ascertaining whether the ontological query model matches any part of the knowledge base;

responsive to identification of a match within the knowledge base, extracting data from the matched portion of the knowledge base; and

returning a search result comprising data extracted from the knowledge base.

16. The system as claimed in claim 10, wherein the knowledge base is implemented based on an ontological model having interrelated concept classes corresponding to all of (i) one or more objects, equipment or components within an industrial plant (ii) observed events or effects or events within said industrial plant (iii) causative factors related to observed events (iv) operator inputs or operator requests in response to an observed event or a determined causative factor (v) any action executed in response to an operator input or operator request and (vi) temporal data.

17. The system as claimed in claim 15, wherein the knowledge base comparator is configured to identify matches between the ontological query model and the knowledge base, based on graph pattern matching.

18. The system as claimed in claim 10, further comprising:
a speech processor configured to:

- receive speech signals from the operator input device;
- to perform speech to text conversion on the received speech signals; and
- communicate the converted text to the natural language engine.

19. A computer program product for machine based processing of natural language input received from an operator within an industrial plant, the computer program product embodied on a non-transitory computer readable medium, and comprising computer readable instructions for:
receiving an operator input from the operator;
classifying the received operator input as one of a query and a statement;
responsive to classification of the received operator input as a statement:

- extracting data from the received operator input; and
- updating a knowledge base by storing the extracted data as a set of interrelated concept instances within said knowledge base;

and

responsive to classification of the received operator input as a query:

- searching at least one of a structured database and the knowledge base for data matching one or more query parameters extracted from the received operator input; and
- presenting to the operator, a query response based on a returned search result.

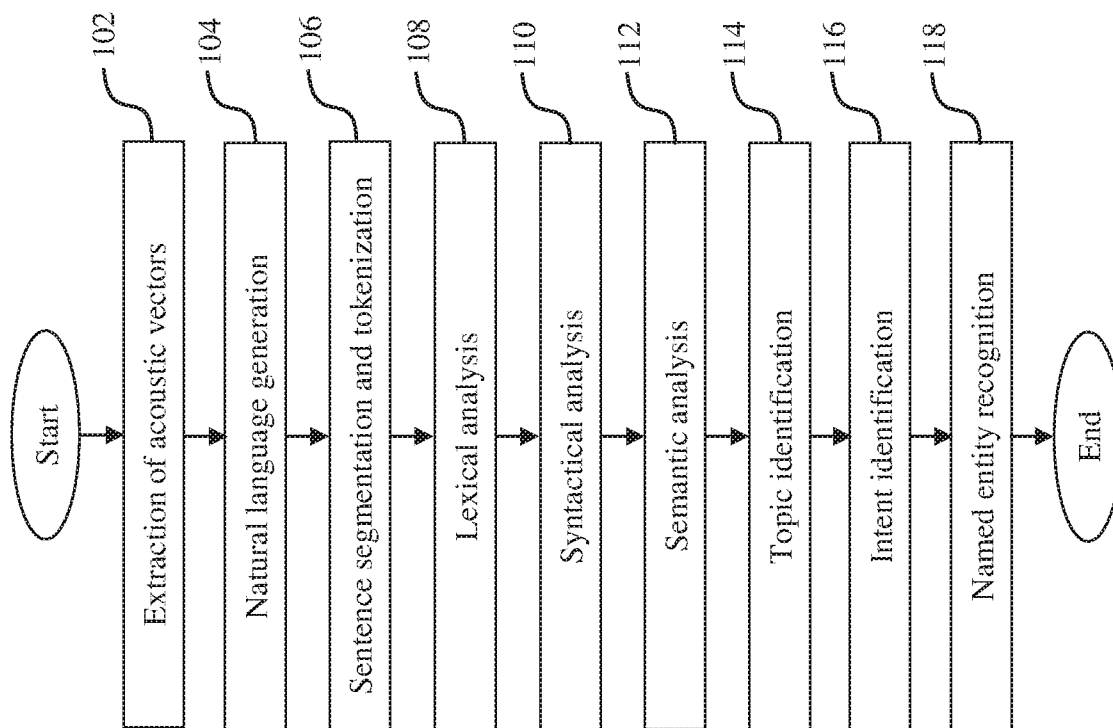


FIGURE 1

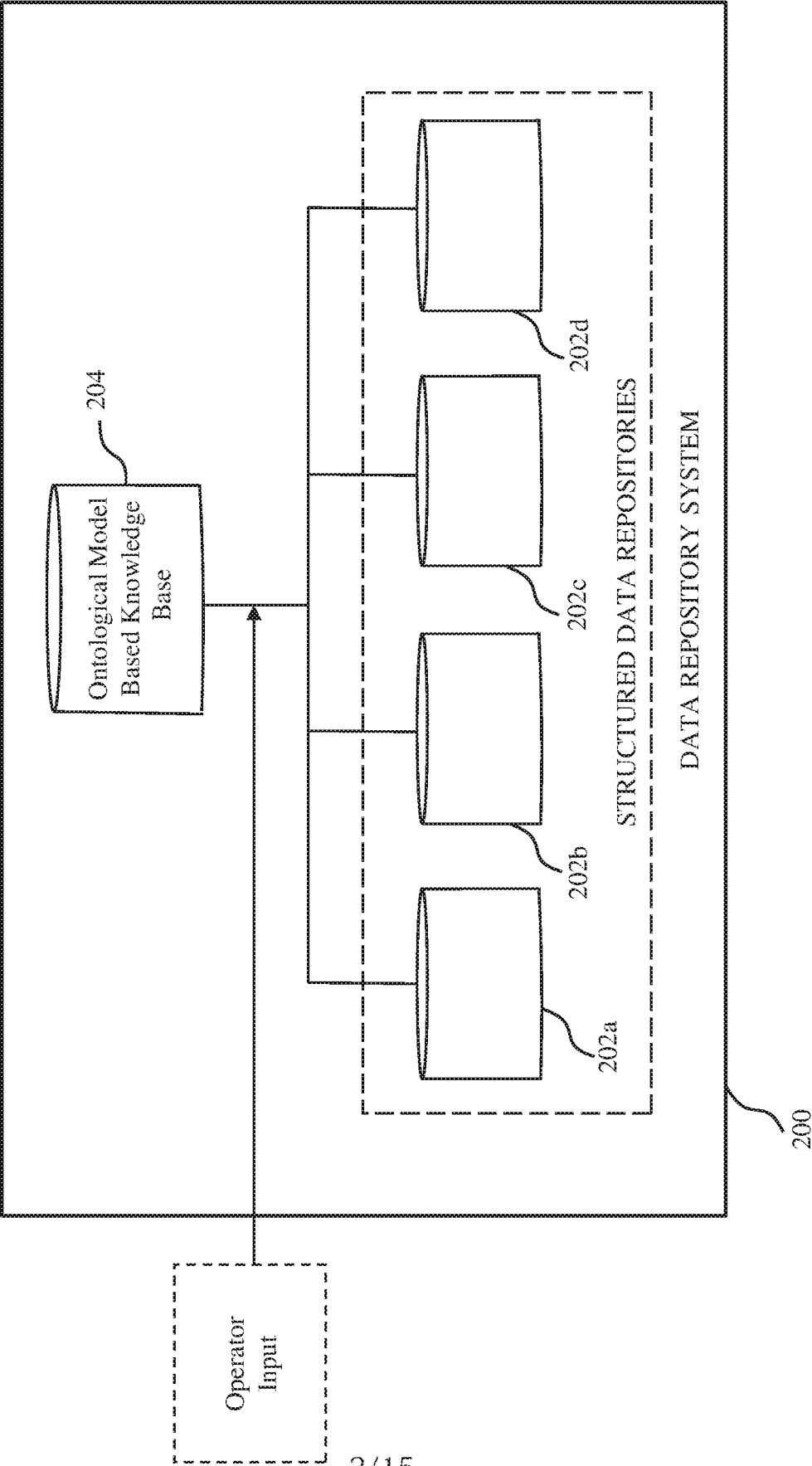


FIGURE 2

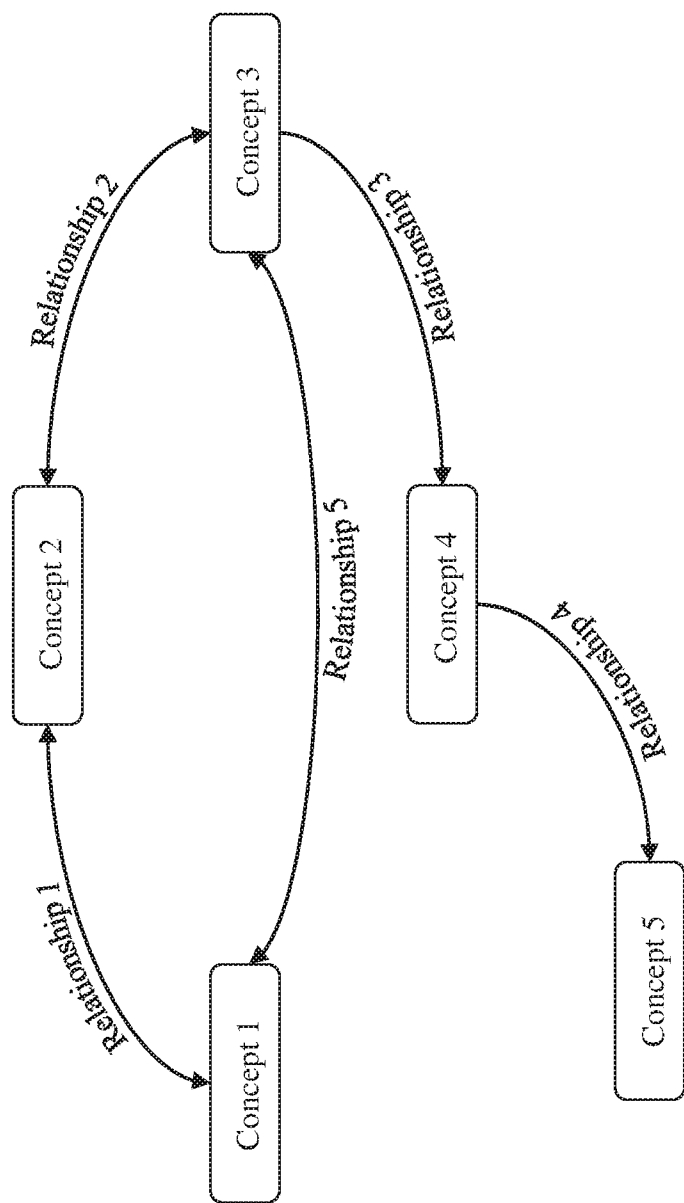


FIGURE 3

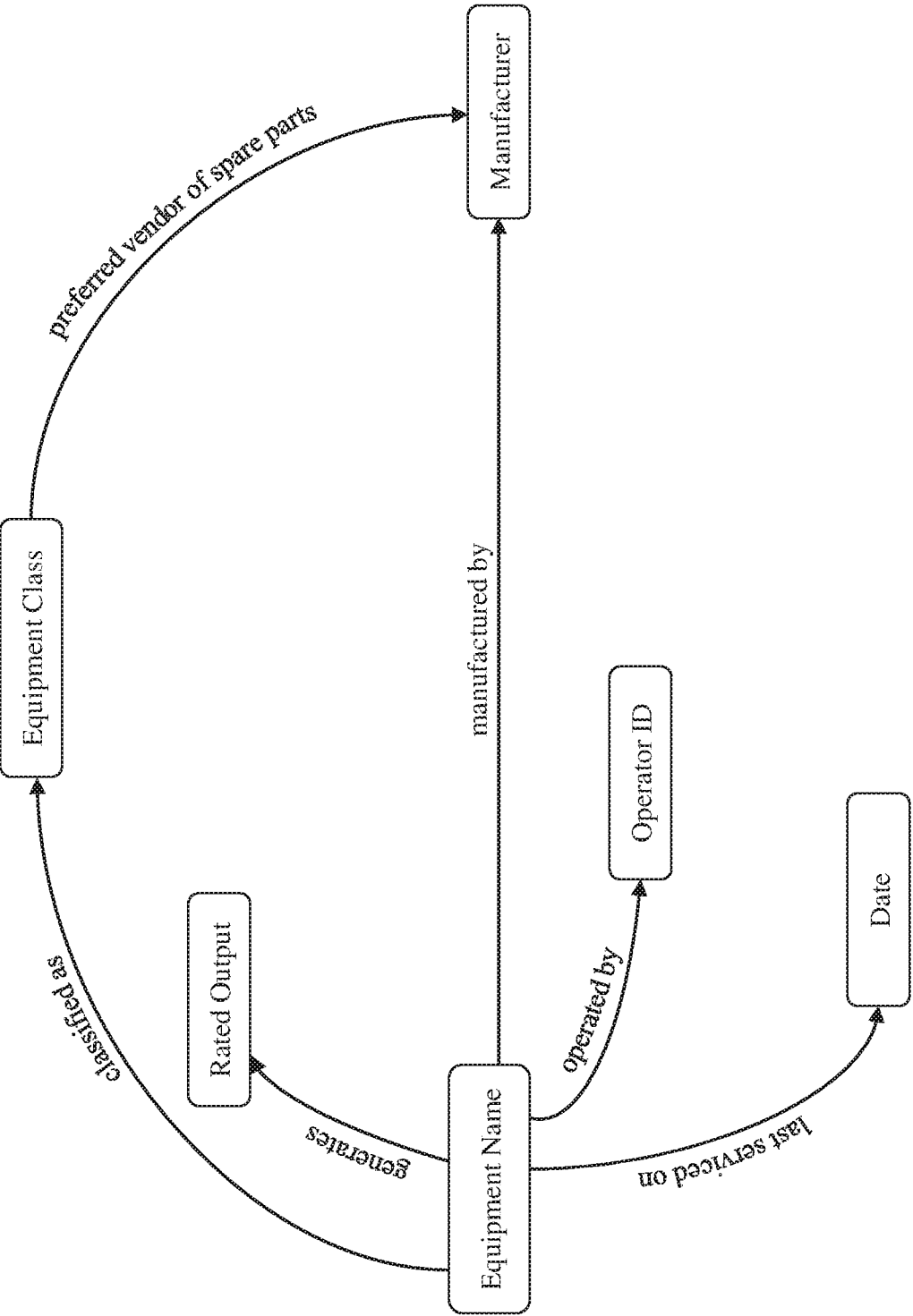


FIGURE 4A

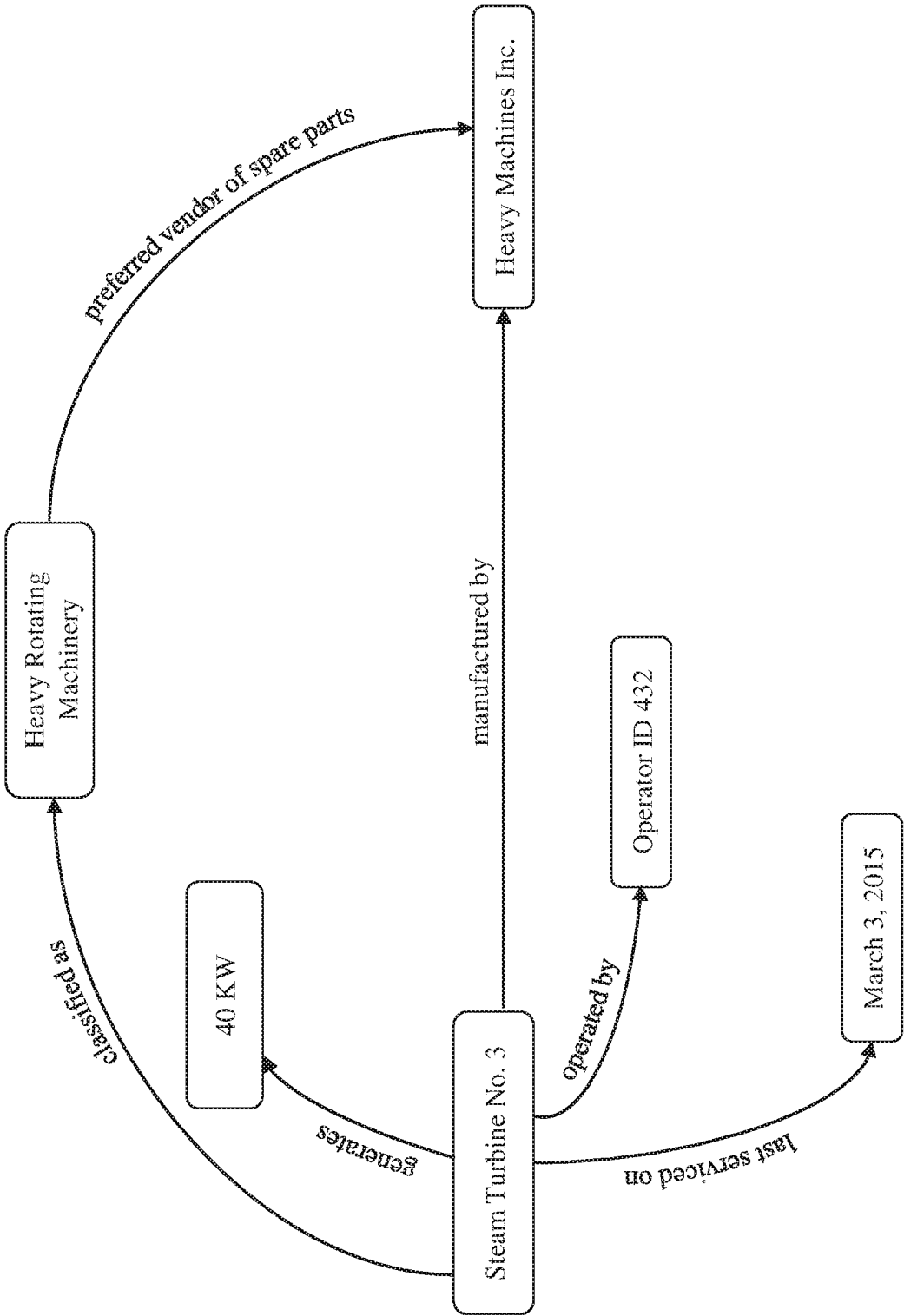


FIGURE 4B

Equipment Name	Rated Output	Equipment Class	Manufacturer	Date	Operator ID

FIGURE 4C

Equipment Name	Rated Output	Equipment Class	Manufacturer	Date	Operator ID
Steam Turbine No. 3	40 KW	Heavy Rotating Machinery	Heavy Machines Inc.	March 3, 2015	Operator ID 432

FIGURE 4D

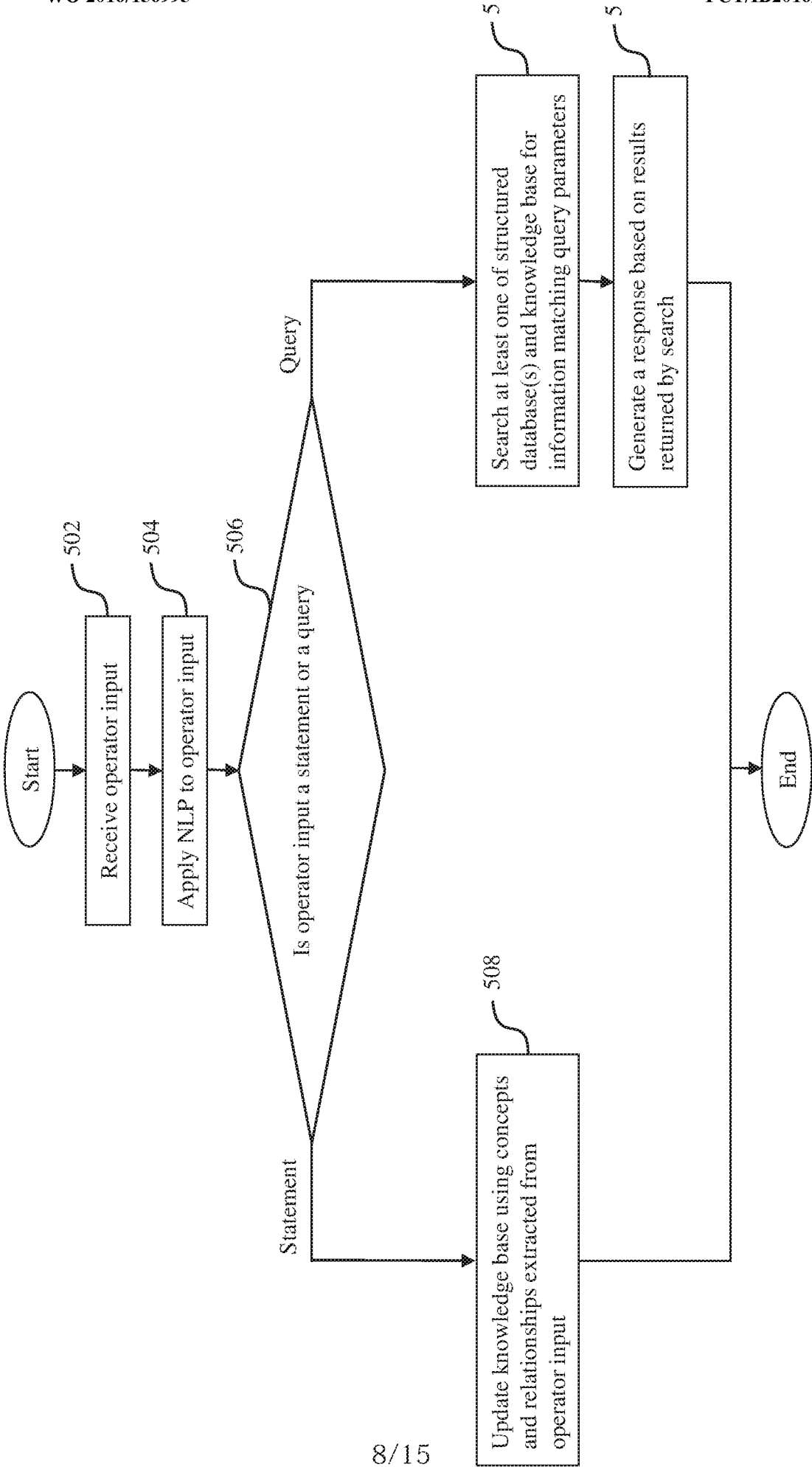


FIGURE 5

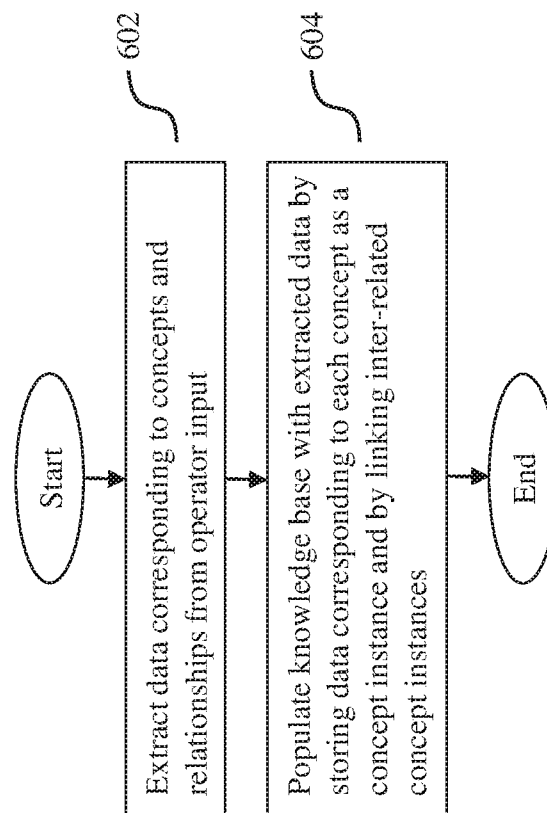


FIGURE 6

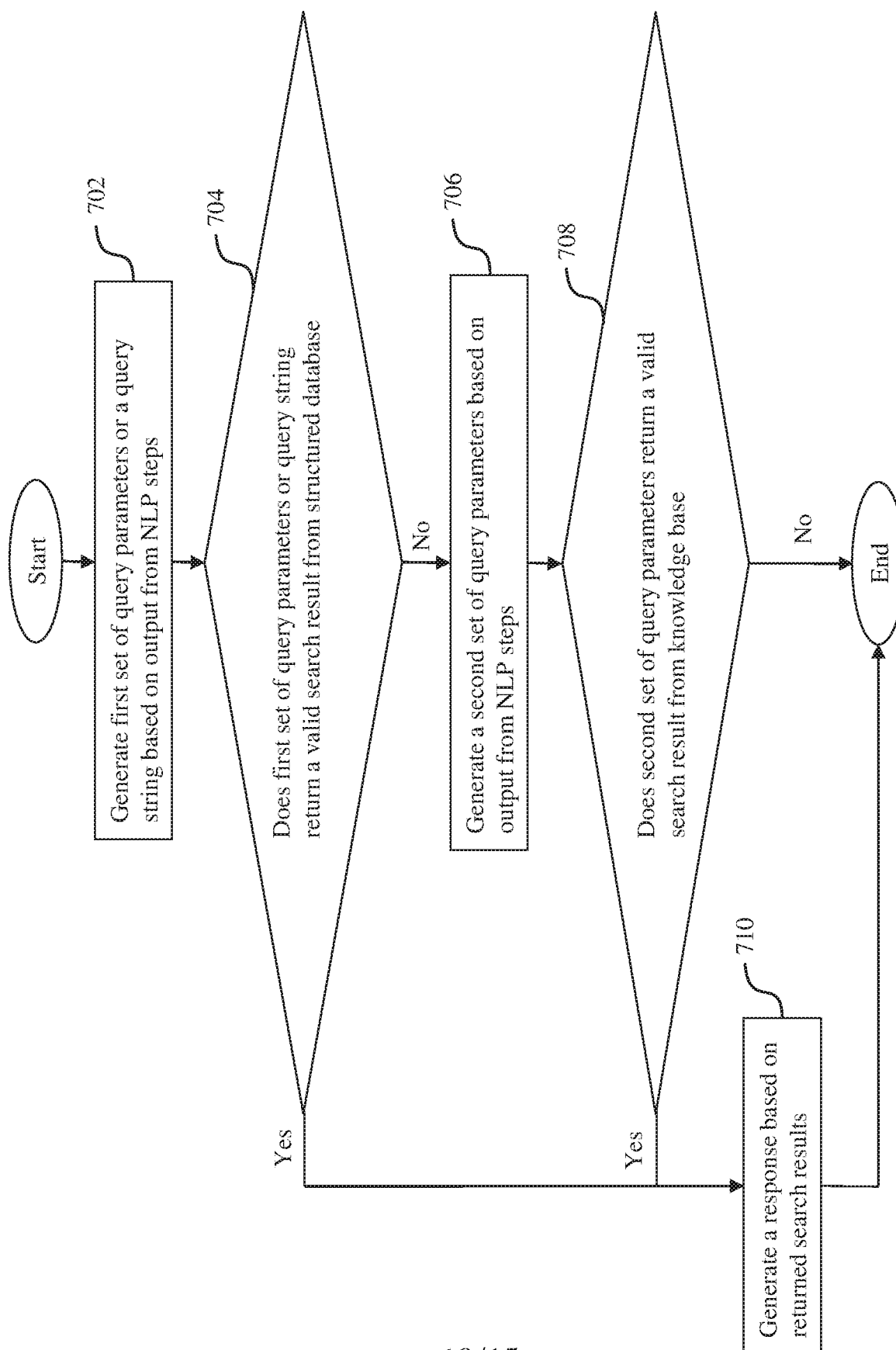


FIGURE 7

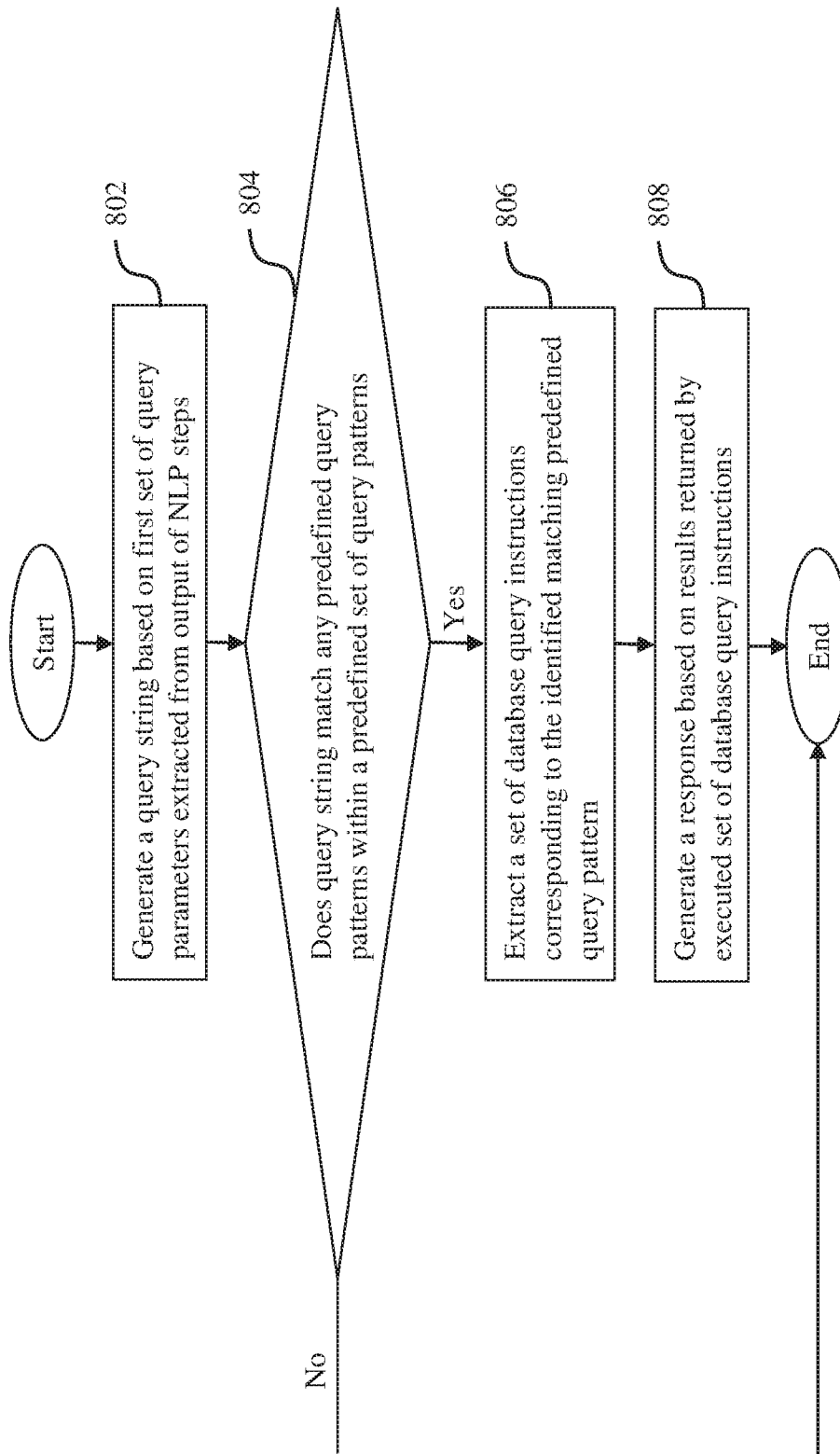


FIGURE 8

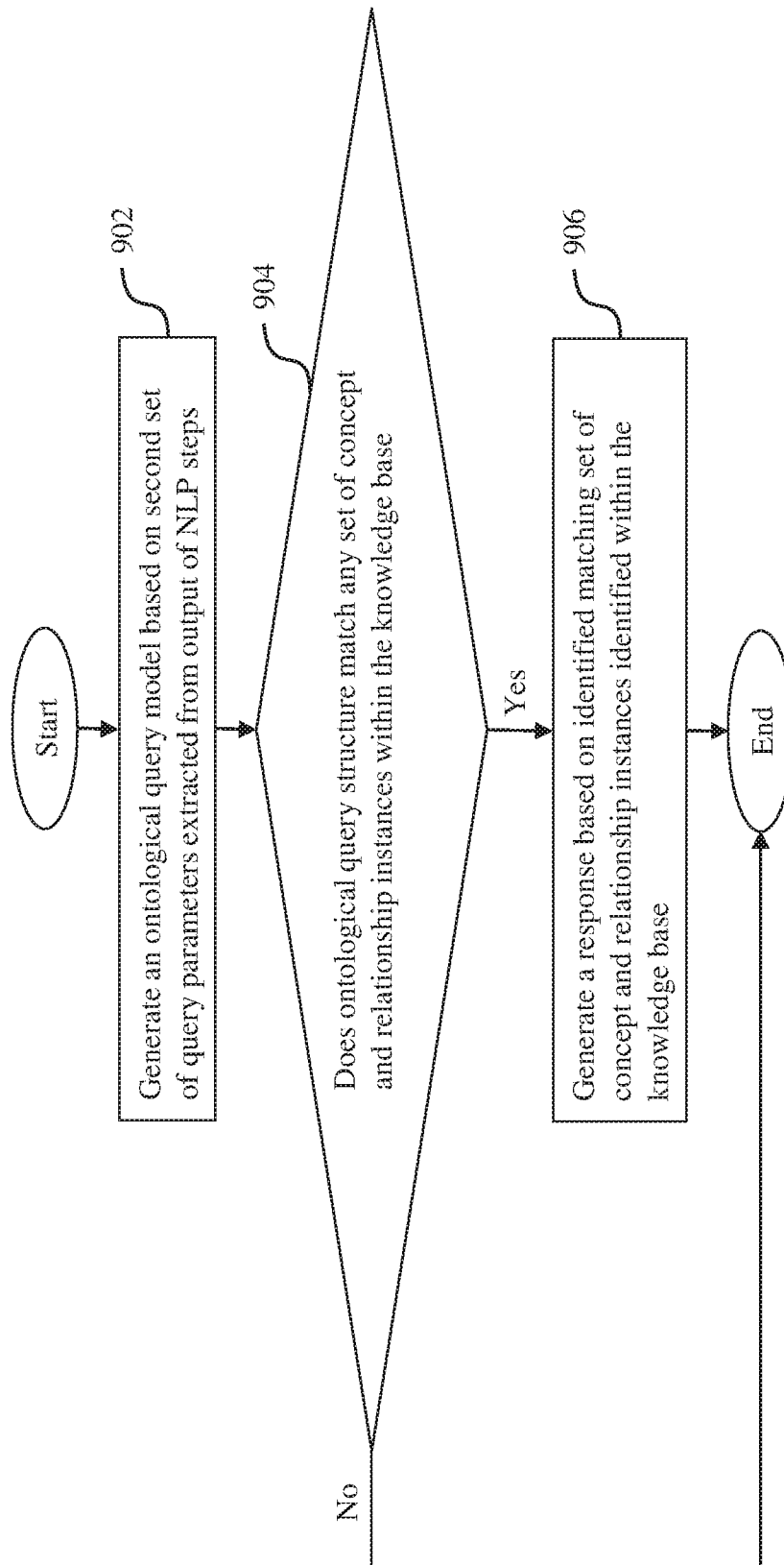


FIGURE 9

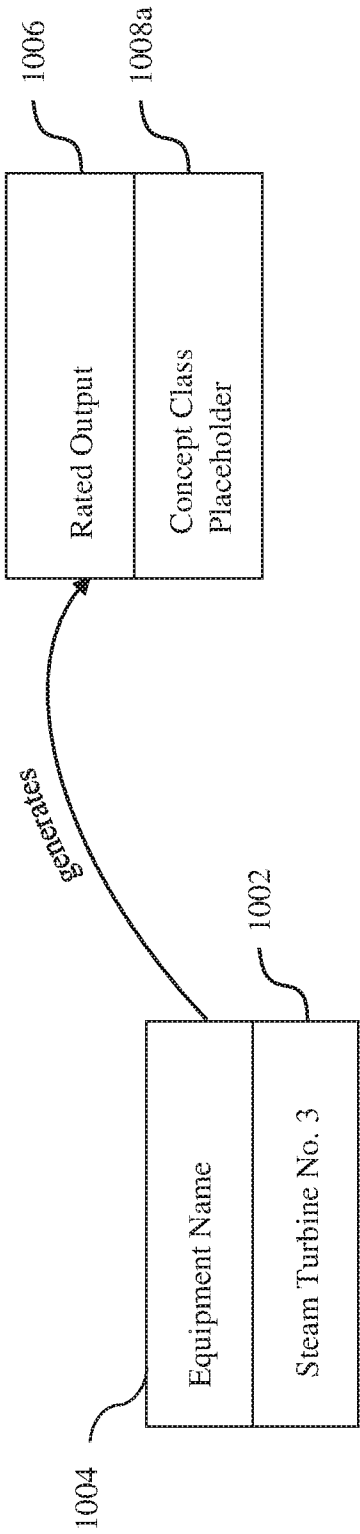


FIGURE 10A

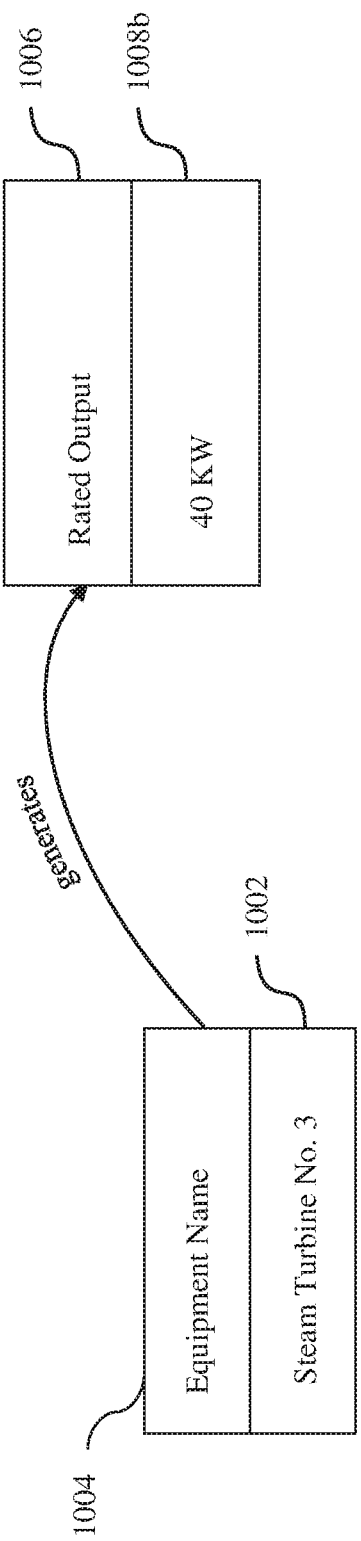


FIGURE 10B

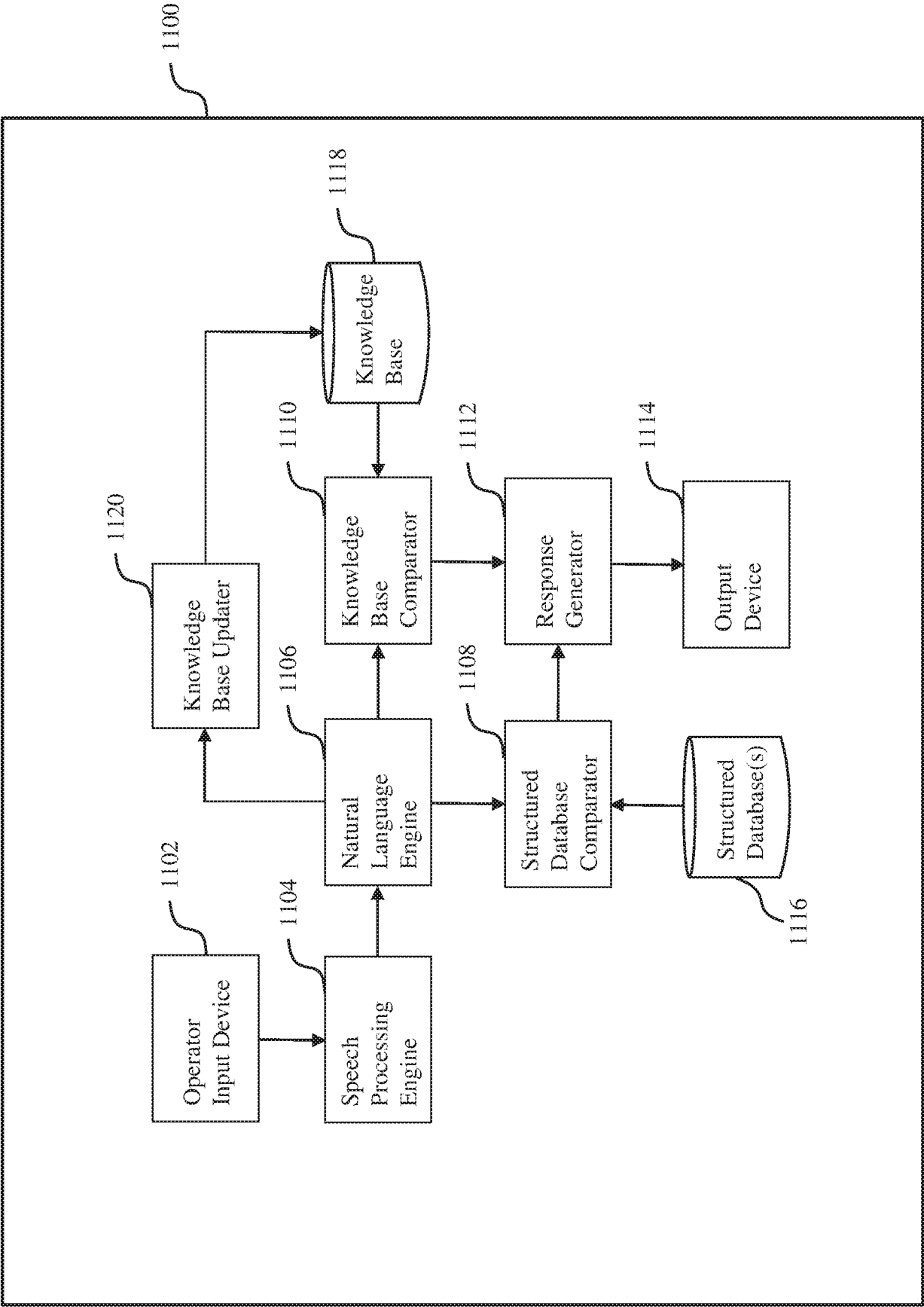


FIGURE 11

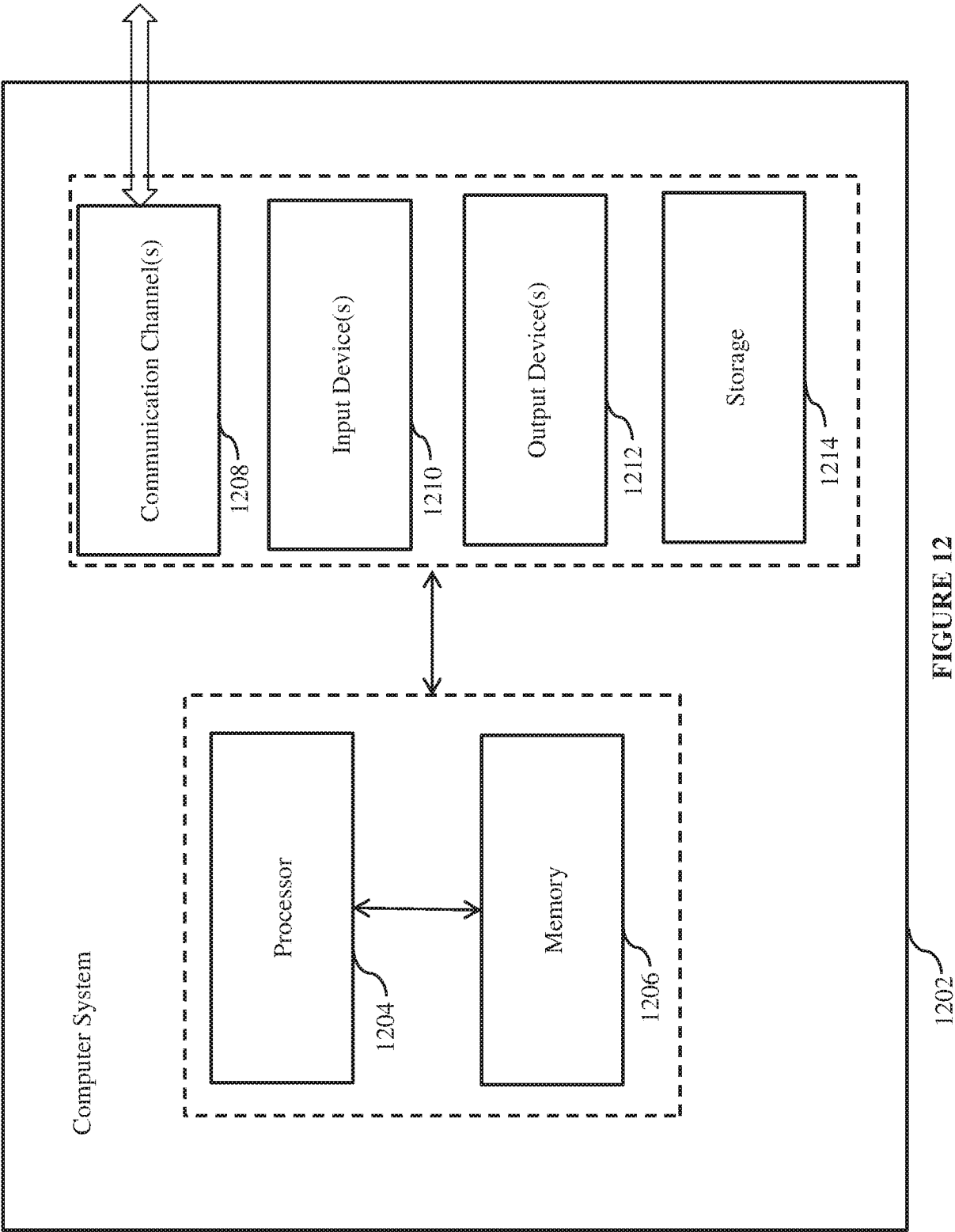


FIGURE 12

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2016/050593

A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - G06F 7/00, G06F 17/30, G10L 15/22 (2016.01) CPC - G05B 2219/31292, G05B 2219/32128, G06F 17/30654, G06F 17/30979, G10L 15/22 (2016.01) According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC(8) - G06F 7/00, G06F 17/30, G10L 15/22 (2016.01) CPC - G05B 2219/31292, G05B 2219/32128, G06F 17/30654, G06F 17/30979, G10L 15/22 (2016.01) Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched USPC - 704/270, 704/274, 704/275, 704/9, 704/E15.04, 707/999.003, 707/999.1, 707/E17.14 (keyword delimited) Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) Orbit, Google Patents, Google Scholar, Google Search terms used: machine based processing natural language input received from operator,		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2007/125108 A1 (ABB RESEARCH LTD) 08 November 2007 (08.11.2007), entire document	1-19
Y	US 2002/0010574 A1 (TSOURIKOV et al) 24 January 2002 (24.01.2002), entire document	1-19
Y	WO 2005/062202 A2 (ESKEBAEK) 07 July 2005 (07.07.2005), entire document	1-19
Y	US 2012/0254143 A1 (VARMA et al) 04 October 2012 (04.10.2012), entire document	1-19
Y	US 2008/0263006 A1 (WOLBER et al) 23 October 2008 (23.10.2008), entire document	2-6, 8, 11-15, 17
Y	US 2004/0039734 A1 (JUDD et al) 26 February 2004 (26.02.2004), entire document	2-6, 8, 11-15, 17
Y	SUN. "XML-based Agent Scripts and Inference Mechanisms." In: Diss. University of North Texas. August 2003 (08.2003). Retrieved from <http://digital.library.unt.edu/ark:/67531/metadc4288/m2/1/high_res_d/thesis.pdf>, entire document	4, 9, 13, 18
A	US 2001/0053969 A1 (HOGENHOUT et al) 20 December 2001 (20.12.2001), entire document	1-19
A	US 5,991,726 A (IMMARCO et al) 23 November 1999 (23.11.1999), entire document	1-19
A	US 2008/0059432 A1 (TAN et al) 06 March 2008 (06.03.2008), entire document	1-19
A	US 2014/0006012 A1 (ZHOU et al) 02 January 2014 (02.01.2014), entire document	1-19
☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 08 May 2016		Date of mailing of the international search report 31 MAY 2016
Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300		Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774