SYSTEM AND METHOD FOR PROCESSING LANGUAGE INDEPENDENT NATURAL LANGUAGE STATEMENTS

FIELD OF THE INVENTION

This invention relates to systems and methods for natural language processing, and more particularly a system and method for processing language independent natural language statements, expressions, rules, or policies in human-computer interface systems.

BACKGROUND OF THE INVENTION

Computer users input data and commands in the form of natural language statements in a computer interface system to instruct the computer to process the data input and perform the required tasks. A natural language processor in the computer translates the received natural language data into machine readable code. Natural language processors typically are utilized in applications such as word processing applications, and include multi-lingual functions to allow translation of input data in different languages. Although these applications commonly include processing functions such as spelling and other more abstract functions such as grammar, these functions are language specific. However, there are applications required for capturing and processing the abstract ideas conveyed in the linguistic or syntactic meaning of the input data, regardless of the specific language of the input data. Such an application where independent language natural language processing is required is described in detail below to illustrate the background of the invention, however, it will be appreciated that other applications also require the same.

In one area where independent language natural language processing is required is, for example, Business Rules Management Systems (BRMS) which enable business users to process, update, and manage business rules and policies in enterprise business systems without calling upon the expertise of an IT specialist. Business users are commonly analysts and any other employee in a business that typically have low-level technical familiarity and are non information technology (IT) specialist trained professionals. Business rules in BRMS are by nature subject to change because the extraneous factors, such as market conditions, law enactments and regulations, company policies and the like, used to form business rules are subject to change. Business rules describe the operations, definitions and constraints that apply to an organization in achieving the goals of the organization. Business rules guide the operations of an organization and are

specific to each organization. For example a business rule might state that no credit check is to be performed on return customers. Others could define a tenant in terms of solvency or list preferred suppliers and supply schedules. These rules are then used, for example, to help the organization to better achieve goals, communicate among principals and agents, communicate between the organization and interested third parties, demonstrate fulfillment of legal obligations, operate more efficiently, automate operations, perform analysis on current practices, etc.

BRMS have been developed in an attempt to allow a business user to alter, update and maintain business rules, information and policies enterprise wide without having to wait for delays due to availability of an IT specialist. An objective of BRMS is to lower customization and application maintenance costs associated with updating and maintaining business systems by improving collaboration by shifting the task from the IT specialists and empowering the business user. BRMS typically consist of a dedicated enterprise rules repository, a business user rule maintenance application, and a business rules engine.

However, in actuality, the conventional BRMS business rules engine apply a simplistic template syntax mapping which maps entire English sentences to a rule definition in proprietary rule language syntax. In such conventional BRMS, every business rule is a new rule that requires individually mapping to the proprietary rule language syntax, which often extends beyond a business user's technical ability and requires a trained IT specialist. Before business users make the rule changes, new mappings are mapped by the IT specialist. Also, since entire English sentences are mapped, converting the conventional BRMS to any other language is also expensive and time consuming. The mappings in conventional simplistic template syntax mapping BRMS are also application and system specific, and may not be reused by other business systems managed under BRMS unless the other business system has exactly the same set of rules, which is very unlikely.

There is a need for a system and method for processing language independent natural language statements, expressions, rules, or policies in human-computer interface systems. For example, there is a need for a system and method for processing language independent natural language statements to enable a business user to easily define and process business expression, rules, and policies in natural languages within a business rules management system.

SUMMARY OF THE INVENTION

In accordance with an aspect of the invention there is provided a system for processing language independent statements composed in natural language, comprising: a natural language processor for receiving a language independent natural language statement and analyzing the statement with reference to a linguistic base, and a knowledge base memory for storing elements of the natural language statement; the natural language processor for identifying within the natural language statement at least one concept element and at least one associated relation element having a logical relationship with the concept, linking the at least one relation with the associated concepts for storage in the knowledge base memory, and a code generator for generating executable logic code representation to map the code representation associated with the concept and relation element representations to the natural language statement.

In an embodiment, data of the elements stored in the knowledge base memory may comprise of the concept element having a natural language expression and an assigned concept tag, and the relation having a natural language phrase and an assigned relation tag and a concept link tag for linking the relation to the associated concept. The concept element may have at least two associated natural language expressions. The relation element may have at least two associated natural language phrases. The natural languages of either the expressions and phrases or both may be in the same or different natural languages.

In an embodiment, the system further comprises a rule engine for receiving the executable logic code representation and mapping the code representation associated with the concept and relation element to the natural language statement. The natural language statement may be a business rule, and the rule engine is a business rules management system. The rule engine may further comprise a rule base for storing the business rules.

In accordance with another aspect of the invention there is provided a method for processing language independent statements composed in natural language, comprising: receiving a language independent natural language statement with a natural language processor and analyzing the statement with the natural language processor with reference to a linguistic base, and storing elements of the natural language statement in a

knowledge base memory; identifying with the natural language processor within the natural language statement at least one concept element and at least one associated relation element having a logical relationship with the concept, linking the at least one relation with the associated concepts for storage in the knowledge base memory, and generating executable logic code representation with a code generator to map the code representation associated with the concept and relation element representations to the natural language statement.

In accordance with another aspect of the invention there is provided a computer program product comprising a computer readable medium having thereon: computer program code means, when loaded on a computer to make the computer perform a method for processing language independent statements composed in natural language, comprising: receiving a language independent natural language statement with a natural language processor and analyzing the statement with the natural language processor with reference to a linguistic base, and storing elements of the natural language statement in a knowledge base memory; identifying with the natural language processor within the natural language statement at least one concept element and at least one associated relation element having a logical relationship with the concept, linking the at least one relation with the associated concepts for storage in the knowledge base memory, and generating executable logic code representation with a code generator to map the code representation associated with the concept and relation element representations to the natural language statement.

In accordance with another aspect of the invention there is provided a computer readable medium, having a program recorder thereon, where the program when loaded on a computer to make the computer perform a method for processing language independent statements composed in natural language, comprising: receiving a language independent natural language statement with a natural language processor and analyzing the statement with the natural language processor with reference to a linguistic base, and storing elements of the natural language statement in a knowledge base memory; identifying with the natural language processor within the natural language statement at least one concept element and at least one associated relation element having a logical relationship with the concept, linking the at least one relation with the associated concepts for storage in the knowledge base memory, and generating executable logic code representation with a code generator to map the code representation associated with the concept and relation element representations to the natural language statement.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example of preferred embodiments, and not by way limitation, in the figures of the accompanying illustrative drawings and in which like reference numerals refer to similar elements and in which:

FIG. 1 illustrates an exemplary system on which various embodiments of the present invention may be implemented;

FIG. 2 illustrates a block diagram of an exemplary knowledge base and natural language processor for a system in accordance with an embodiment of the invention;

FIG. 3 illustrates a chart showing the different users roles working together in the system in accordance with an embodiment of the invention;

FIG. 4 illustrates a process in accordance with an embodiment of the invention;

FIG. 5 illustrates an exemplary anatomy of a natural language statement in accordance with an embodiment of the invention;

FIG. 6 illustrates an exemplary selection of construct, relations and parse tree within a natural language statement in accordance with an embodiment of the invention;

FIG. 7 illustrates an exemplary change of rules within a natural language statement in accordance with an embodiment of the invention; and

FIG. 8 illustrates an exemplary ambiguity associated with natural language of a business rule of a system in accordance with an embodiment of the invention; and

FIG. 9A-C are exemplary graphical user interface (GUI) displays showing contents of a domain knowledge base, business rule base, results of a rules simulation run in studio of the BRMS in accordance with an embodiment of the invention

DETAILED DESCRIPTION

A system and method in a computer interface system for capturing and processing abstract ideas conveyed in the linguistic or syntactic meaning of the input data, regardless of the specific language of the input data is described. Such an application where language independent natural language processing is required is described in detail below to illustrate an embodiment of the invention, however it will be appreciated that other embodiments of the invention may be embodied in other applications. For example, a system and method of an embodiment of the invention for defining and processing business expression, rules, and policies in natural languages within a business rules management system is described. However, the language independent natural language

processing system and method according to embodiments of the invention may be used in any human computer interface for any computer system. For example, combined with voice recognition technology, the language independent natural language processing technology of an embodiment of the invention may enable Operating Systems (such as MAC OS, WINDOWS, LINUX) to accept voice commands directly from user. MAC OS is a trademark of Apple Computer, Inc. of Cupertino, California, USA registered in the US and other countries. WINDOWS is a trademark of Microsoft Corporation of Redmond, Washington, USA registered in the US and other countries. LINUX is a trademark of Linus Torvalds in the US and other countries. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It is apparent to one skilled in the art, however, that the present invention may be practiced without these specific details or with equivalent arrangements. In some instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

Referring to FIG. 1, an exemplary system 10 is shown on which various embodiments of the invention may be implemented. FIG. 1 shows a language independent natural language processing system 10. The particular application of the system for sake of description and illustrative purposes for describing embodiments of the invention is to provide language independent natural language processing to Business Rules Management Systems (BRMS), as represented in FIG. 1 by rules engine 32, with rule base 26 and rule engine 30. The language independent natural language processor 24 resides on server 22 with the rules engine 32 and knowledge base 28. The natural language processor 24 includes registers, control and arithmetic logic unit arranged to perform the desired functions and the knowledge base includes memory for storing concepts, relations and phrases of the natural language input statements and are described in more detail with reference to FIG. 2. The rules in natural language are defined by policy manager 52 via studio interface 58 in the development area 50 indicated by dashed line of FIG. 1. The knowledge base is created by business analyst 54. During run time an end user 14 via end-user application 16 may access the server via network 12 to run the rules. A system administrator 18 via system administrator interface 20 may access to monitor the system. It will be appreciated that other connectivity arrangements may be configured, and although network 12 is shown with the development area not connected to network 12, the studio interface 58 may be arranged via network 12. Likewise, it may be possible to arrange and perform each function by system

administrator, end user, policy manager, business analyst, and software developer without a network 12 configuration, for example, by direct and separate access to the server 22. The resource library 40 includes a knowledge repository 42 and linguistic repository 44 having memory for storing, and accessible via network 12 of the knowledge base elements uploaded to the resource library 40. This configuration allows for any authorized users, such as others not shown, to access the resource library 40 for use in other applications, or other domain specific BRMS where the knowledge base may be applied.

It will be appreciated that embodiments of the invention may be applied in other areas for processing of natural language statements. With a supporting knowledge base 28, the natural language processor 24 can accept natural language statements, either as predefined statements in rule base 26 described in this embodiment or direct user input through any human computer interface device, and convert the statements into executable logics.

Referring to FIG. 2, a knowledge base 28 and a natural language processor 24 is shown of an embodiment of the invention. The knowledge base 28 and natural language processor 24 can parse natural language statements 100 input to the natural language processor by analyzing and breaking the statements into functional units or elements that can be processed and output as executable business logic 102 and mapped to executable object modules 66. The executable modules 66 and the executable logic 102 are stored in the data files (not shown) of the server 22. The processor 108 receives the natural language statements, which may be textual data strings received by a user input, at lexical analyzer 70, syntax analyzer 72, ambiguity resolver 74 and code generator 76 to output the executable logic 102. The natural language processor 24 has a capability that is extensible with linguistic base 106. As shown in FIG. 2, the natural language processor 24 is designed so that different languages may be supported by adding in language modules, for example English 80, Chinese 90, or other languages 98. The language modules of the linguistic base 106 may be memory for storing of construct 82, dictionary 84 and grammar 86 data. Furthermore, the natural language processor 24 is independent from knowledge base 28. The knowledge base 28 consists of memory for storing the functional units of natural language statements 100 of concepts 60, expressions 63, relations 62, phrases 64, and contextual knowledge 68. With properly defined knowledge base, rules with special terminologies in any business domain may be processed.

In business documents, a business rule is typically in the form of "If..., then...". Generally a business rule comprises two main parts, the first being the condition that is the "If..." portion and the second being the decision/action that is the "then..." portion. Using the credit card approval as an example, there are typically the following requirements for credit card application: 1. applicant must be at least 21 years old and less than 60 years old; and 2. applicant's annual income must be above $30,000. The business rule for above can be expressed as: IF: applicant is above 21 years old AND, applicant is less than 60 years old AND, applicant's annual income is above $30,000, THEN: applicant is eligible for credit card.

There are three conditions and one action in the business rules above. Each rule has one action, and one or multiple conditions. This is shown in FIG. 5 which illustrates an exemplary anatomy 230 of a natural language statement 210, or in this embodiment a business rule of a BRMS. Each natural language statement 210 may comprise sub-statements 212 divided by natural language constructs 214. Examples of constructs include for example and, or, and the like. A sub-statement 212 is comprised of one or more concepts 216 and a relation 218 between the concepts. The concept consists of expressions 220, where expressions are the natural language representations of the concept. For example the concept "applicant" may have expressions in English, Chinese, or other language. Each representation of "applicant" in the different languages form the different expressions 220 associated with the concept 216. Additionally, associated with each concept is an identification number, marker or tag data 232 of an assigned number or the like, being an abstract representative of the natural language concept expression. The identification number may be assigned by the server. For example, the concept "applicant" is assigned an identification number tag that is stored in the concepts 60 section of the knowledge base 28 of the system 10. In this manner, the identification tag number is associated with the concept 216, and all associated expressions. This process is the first step in a process of reduction. The aim of the process of reduction is to reduce each natural language statement to a single concept, i.e. a concept that is represented by an assigned identification number tag. A sub-statement may have one or more relations. A relation 218 may include one or more phrases 222 and a binding 224. A phrase 222 is associated with a relation 218. A phrase is the natural language representation of the relation, and an identification number, marker or tag data 234 of an assigned number or the like, being an abstract representative of the natural language relation phrase is assigned to the relation to represent the natural language phrase 222 and stored in the relations section of the knowledge base 28. Similarly, the identification number may be

assigned by the server. In this embodiment, there may be a rule set comprising of at least one rule. As indicated above, each rule or natural language statement 210 consists of at least one condition and only one action. Each condition or action is a statement in natural language. Each statement involves one or multiple concepts. Concepts are interconnected by relations 218, as shown by bi-directional arrow in FIG. 5, which define the relationship between concepts. The link between concept and relation is shown by concept identification number link 238 as part of relation. Of course, there may be other configurations of the anatomy of a natural language statement, for example, the link between concept and relation may be associated with concept 216. Each relation can be expressed in one or multiple phrases 222 in natural language. A condition and an action (not shown) may consist of at least one relation 218, and a relation consists of at least one concept 216 and is defined by a phrase 222.

For example, in the following statement: "Applicant's age is greater than 21". "Applicant" is a concept and "Age" is another concept. "Applicant" concept has a relation with the "Age" concept. Each concept may have one or multiple expressions (alias): and each relation may be expressed as one or multiple phrases. For example, "applicant's age" is one way of expressing the relation in English, while the same relation can be expressed as "age of applicant", "the age of applicant" and even other forms in other languages. Concepts and relations are abstract, but their expressions and phrases are the concrete natural language representations of the concept/relation. With the expressions and phrases defined in the knowledge base, what relations are involved may be determined in a natural language statement.

Each relation has a return type, which is the type of value that the relation refers to. For example, the "Applicant's age" relation has a return type of "Age" because the relation refers to the age value of the applicant. With the return type information, how various relations involved in the natural language statement interconnect with each other may be determined.

In order to understand business rules in natural language the system requires the user to develop a supporting knowledge base for the domain, which consist of concepts 60, expressions 63, relations 62 and phrases 64, and executable modules 66, as shown in FIG. 2. The executable modules 66, also referred to as class modules, may be JAVA classes that facilitate the business rules. JAVA is a trademark of Sun Microsystems, Inc. of Santa Clara, California, USA registered in the US and other countries. Class modules

and bindings associate relations in the class modules with JAVA classes in executable modules. The knowledge base 28 is language independent. As explained above, concepts and relations captured in the knowledge bases are abstract, that is the natural language functional units are represented by abstract representations, such as identification number tags. The system 10 allows a user to define language specific representations of concepts/relations to support different languages. This approach is flexible and extensible as a common knowledge base can be used to support rules in multiple languages.

Turning to FIG. 4, when a natural language statement is entered by the user and a natural language statement text data string is received 302, the statement is scanned 304 for identifying natural language constructs 306 such as "and" and "or", which break the statement into sub-statements with logical relationships. For example: "Applicant's age is greater than 21 and less than 60" becomes "Applicant's age is greater than 21 AND Applicant's age is less than 60". The sub-statements are scanned 312 and expressions of concepts are identified 314 and phrases of relations of concepts are identified 316. This is performed by the system 10 in the natural language processor 24 shown in FIG. 2. The natural language statement 100 is received and scanned by the lexical analyzer 70 and syntax analyzer 72 to identify the functional elements of the statement, i.e. construct, expressions and phrases. The linguistic models are relied upon to recognize the natural language constructs. For each construct, one or multiple string patterns are defined in the linguistic model. The natural language processor will search for the string patterns provided by the linguistic models in order to identify natural language constructs in the rules. Therefore, the natural language processor itself is language independent since it relies on the linguistic models to identify the natural language constructs in the rules. Each sub-statement is scanned through and concepts that are used in the sentence are identified. For example, in the natural language statement in FIG. 7 "Applicant" 274, "Annual income" 276, "Credit Card" and "Minimum Income" are the concepts used in the statement 270. After identifying the concepts, the statement is scanned for patterns that can form relations for the concepts used in the statement. The reduction process of relation to concepts 318 is performed to reduce the natural language statement to a single concept. For example, in the reduction process, in the statement "applicant's age is greater than 21." Applicant's age is a relation that refers to "Age". Therefore, the original statement may be reduced to "age is greater than 21", which in turn may be reduced to a single Boolean (true/false) concept. This reduction process forms a parse tree.

As illustrated in FIG. 6, "applicant's annual income" 252 is a relation for "Applicant" and "Annual Income" concepts and returns a "NUMBER" concept 258. Then "applicant's annual income" 262 portion is replaced with a "NUMBER" concept 258. This process is called reduction. The same process applies for the "credit card's minimum income" 256 section of the natural statement 250, which is reduced to "NUMBER" concept 260 from "credit card's minimum income" 264 portion. After the two reductions, the original statement has been reduced to "NUMBER is greater than NUMBER" 266, which is a relation 254 for two "NUMBER" concepts between "is greater than" 254 and returns a "BOOLEAN" concept. The whole sentence may now be reduced a single concept, therefore, this statement is considered fully understood by the system. The structures of the sentence of the elements grouped by dashed line 268 are known as a parse tree 268.

Returning to FIG. 4, once the expressions of concepts and phrases of relations have been identified, if any ambiguity 330 is found, a users input is requested to clarify the ambiguity 332. In breaking down the natural language statements into functional units, such as identifying constructs, concepts and relations to identify logical relationships, generic string token is considered. The string patterns of the natural language statements are language specific and are provided by the linguistic models 80,90,98 in the linguistic base 106 of the natural language processor 24. Some languages inherently have less ambiguity because there are dividers between string tokens within a statement, where string tokens are representative of a single word. String tokens separated by dividers are found in the construction of some languages like the English language. Whereas, other languages such as Chinese do not have multiple token strings in a single statement, and therefore have much more ambiguity. An example of an ambiguity 220 between two natural language statements is shown in FIG. 8, where the natural language statement 240 may be interpreted in more than one way. The concept 242 may be identified where credit is the concept, limit is treated as a verb, and "rating of consumer" 242 is a relation that refers to "rating" as concept. Viewed in a different way, "credit limit rating" 244 is a concept, which has a "has a" relationship with "customer" concept. This ambiguity is resolved by input from a user to clarify, and the clarification is stored in the knowledge base 28 as user supplied contextual knowledge 68.

After ambiguity is resolved by the ambiguity resolver 74 of FIG. 2, another part of the process of an embodiment shown in FIG. 4 is that a version number may be assigned to the statement that is stored in the rule base 324. When a statement (condition/action) is newly defined or updated, a new version number is issued such that the history of the

statement may be tracked and reviewed. The server may assign the version number. Then, at code generation, the entire natural language statement is mapped 326 at execution sequence in the engine 30 in the target engine elements 32, for example rule engine 32 with rule base 26 and engine 30 as shown in FIG. 1.

FIG. 3 illustrates a chart showing the roles in the different stages of development and run time of the system in accordance with an embodiment. A policy manager 52 defines the rules 112 and simulates 120, then publishes the rules 122. A business analyst 54 defines the concepts/expressions 114, relations/phrases 116, and defines binding 118. Although the software developer 56, shown on FIG. 1, is not shown in FIG. 3, the role of the software developer is to develop executable modules. The business analyst creates relations in the knowledge base and binds relations to executable modules. The executable modules facilitate rule execution during runtime where end user 14 accesses rules in the rule engine 32 on the server 22 through the user application 16. It will be appreciated that the titles or names chosen for the roles are chosen for illustrative purpose to describe different roles and functionality of the system and it will be appreciated that a single user may be able to operate the system which performs all of the different roles.

The order of the tasks completed between the policy manager and the business analyst may be interchanged, for example, the concepts/expressions, relations/phrases, and binding may be defined prior to the rules. As shown in FIG. 3, a policy manager 52 simulates the rules. Simulation allows user to assess and verify the behavior of rules with test data. Referring to FIG. 9A-C which show GUI 170,180,190 that may be displayed on a monitor or display apparatus that may form part of interface 20,58 or application 16 for users may interface with the system 22, for example with input via a keyboard and display for user to view results or actions of the system. Referring to FIG. 9C, for example, which shows a GUI 190, a user can key in test data on the right side of the screen 194, upon confirmation the system will execute the rules with the test data provided by the user and display the execution result on the left side of the screen 192 with executed rules and satisfied conditions highlighted. Additionally, publishing is the process in which natural language rules defined by policy manager will be converted into executable logics. FIG. 9A is a GUI 170 for the knowledge base designer for business analyst to define concepts and expressions (shown in left window 172), relations and phrases (shown in right window 174) and binding information (shown in bottom window 176) in the knowledge base. FIG. 9B is a GUI 180 for the rule designer for policy manger to define and manage business

rules in natural language by inputting statements (statement window 186), to define rules (shown in left window 182), document view 184, and properties of the highlighted rule are displayed (shown in bottom window 188). It will be appreciated that any entry of natural language statement or upon receiving a natural language statement, the natural language statement may be any language or language independent, that is not specific to a particular language since the natural language statement and natural language processor is supported by the linguistic base and the concepts, expressions, relations and phrases in whatever language that are identified in the natural language statement may be converted to executable logics. For example, rules may be presented to the system in different natural language such as English or Chinese as shown in the left and right windows 182,184 in the GUI 180 of FIG. 9B. After publishing the rules, the executable logics generated from the rules are ready to be executed upon request from end-user applications that make use of the rules. End-user application connects to the server through network and request for execution of rules. The server will execute the corresponding executable logics for the rules requested and return the execution result back to end-user application through network. Once the system enters run time, the different roles of the end user, policy manager or business analyst may be performed to run, tweak, redefine, or delete previously processed natural statements and/or generate and process new natural statements. For example, FIG. 7 shows a natural language statement 270 being changed to a new updated natural language statement 272 by simply swapping the concept of "annual income" 276 to "disposable income" 278 by amending the natural language statement rule 274. The procedure of changing the natural statement 270 is the same step taken as the policy manager to define 112, simulate 120 and publish 122 the rules as discussed above. However, the existing statement in the system is retrieved from the target application 32 rule base 26, and the user edits the statement and the new updated statement is stored in the rule base 26 replacing the previous original statement.

The devices and subsystems of the exemplary methods and systems described with respect to FIGs. 1-9A-C can communicate, for example, over a communication network 12, and can include any suitable servers, workstations, personal computers (PCs), laptop computers, handheld devices, with visual displays and/or monitors, telephones, cellular telephones, wireless devices, PDAs, Internet appliances, set top boxes, modems, other devices, and the like, capable of performing the processes of the disclosed exemplary embodiments. The devices and subsystems, for example, may communicate with each other using any suitable protocol and may be implemented using a general-purpose

computer system and the like. One or more interface mechanisms may be employed, for example, including Internet access, telecommunications in any suitable form, such as voice, modem, and the like, wireless communications media, and the like. Accordingly, network 12 may include, for example, wireless communications networks, cellular communications network, Public Switched Telephone Networks (PSTNs), Packet Data Networks (PDNs), the Internet, intranets, hybrid communications networks, combinations thereof, and the like.

It is to be understood that the embodiments, as described with respect to FIGs. 1-9A-C, are for exemplary purposes, as many variations of the specific hardware used to implement the disclosed exemplary embodiments are possible. For example, the functionality of the devices and the subsystems of the embodiments may be implemented via one or more programmed computer system or devices. To implement such variations as well as other variations, a single computer system may be programmed to perform the functions of one or more of the devices and subsystems of the exemplary systems. On the other hand, two or more programmed computer systems or devices may be substituted for any one of the devices and subsystems of the exemplary systems. Accordingly, principles and advantages of distributed processing, such as redundancy, replication, and the like, also may be implemented, as desired, for example, to increase robustness and performance of the exemplary systems described with respect to FIGs. 1-9A-C.

The exemplary systems described with respect to FIGs. 1-9A-C may be used to store information relating to various processes described herein. This information may be stored in one or more memories, such as hard disk, optical disk, magneto-optical disk, RAM, and the like, of the devices and sub-systems of the embodiments. One or more databases of the devices and subsystems may store the information used to implement the exemplary embodiments. The databases may be organized using data structures, such as records, tables, arrays, fields, graphs, trees, lists, and the like, included in one or more memories, such as the memories listed above.

All or a portion of the exemplary systems described with respect to FIGs. 1-9A-C may be conveniently implemented using one or more general-purpose computer systems, microprocessors, digital signal processors, micro-controllers, and the like, programmed according to the teachings of the disclosed exemplary embodiments. Appropriate software may be readily prepared by programmers of ordinary skill based on the

teachings of the disclosed exemplary embodiments. In addition, the exemplary systems may be implemented by the preparation of application-specific integrated circuits or by interconnecting an appropriate network of component circuits.

Advantageously, the exemplary embodiments described herein may be employed in offline systems, online systems, and the like, and in applications, such as TV applications, computer applications, DVD applications, VCR applications, appliance applications, CD play applications, and the like. In addition the signals employed to transmit the business rule expressions, the rights expression, and the like, of the exemplary embodiments, may be configured to be transmitted within the visible spectrum of a human, within the audible spectrum of a human, combinations thereof, and the like.

Although the exemplary embodiments are described of applications in business rules, analytical applications, and the like, the exemplary embodiments are applicable to any suitable application, such as digital and non-digital content, devices, software, services, goods, resources, and the like, and may be practiced with variations in technology, interface, language, grammar, content, rights, offerings, services, speed, size, limitations, devices, and the like.

While there has been described in the foregoing description preferred embodiments of the invention, it will be understood by those skilled in the technology concerned that many variations or modifications in details of design or construction may be made without departing from the scope or purview of present invention as defined by the appended claims.

CLAIMS:

1. A system for processing language independent statements composed in natural language, comprising: a natural language processor for receiving a language independent natural language statement and analyzing the statement with reference to a linguistic base, and a knowledge base memory for storing elements of the natural language statement; the natural language processor for identifying within the natural language statement at least one concept element and at least one associated relation element having a logical relationship with the concept, linking the at least one relation with the associated concepts for storage in the knowledge base memory, and a code generator for generating executable logic code representation to map the code representation associated with the concept and relation element representations to the natural language statement.
2. A system as claimed in claim 1 wherein data of the elements stored in the knowledge base memory comprise of the concept element having a natural language expression and an assigned concept tag, and the relation having a natural language phrase and an assigned relation tag and a concept link tag for linking the relation to the associated concept.
3. A system as claimed in claim 2 wherein the concept element has at least two associated natural language expressions.
4. A system as claimed in claim 2 or 3 wherein the relation element has at least two associated natural language phrases.
5. A system as claimed in claim 3 or 4 wherein the natural languages of the expressions and phrases are in different natural languages.
6. A system as claimed in any preceding claim wherein the natural language processor for identifying within the natural language statement with reference to a linguistic base at least one construct element and forming two sub-statements for each construct element within the natural language statement for processing.
7. A system as claimed in any preceding claim wherein the natural language processor comprises a lexical analyzer and a syntax analyzer for analyzing the received natural language statement.
8. A system as claimed in any preceding claim wherein the natural language processor further comprises an ambiguity resolver for resolving ambiguity between different interpretations of relation and concept elements.

9. A system as claimed in claim 8 wherein the ambiguity resolver resolves the ambiguity by updating the linking between the relation and concept elements and storing the contextual knowledge in the knowledge base memory.
10. A system as claimed in any preceding claims wherein the linguistic base memory comprises at least one language module comprising a construct source, dictionary, and grammar source.
11. A system as claimed in claim 10 wherein the at least one language module comprises an English language module and a Chinese language module.
12. A system as claimed in any preceding claim wherein the received natural language statement is parsed and reduced to a single concept element.
13. A system as claimed in any preceding claim wherein the concept and relation elements are for separate storage in the knowledge base memory.
14. A system as claimed in any preceding claim further comprising an executable module for binding with a relation to associate the relation with natural language statement.
15. A system as claimed in any preceding claim further comprising a rule engine for receiving the executable logic code representation and mapping the code representation associated with the concept and relation element to the natural language statement.
16. A system as claimed in claim 15 wherein the natural language statement is a business rule, and the rule engine is a business rules management system.
17. A system as claimed in claim 16 wherein the rule engine further comprises a rule base for storing the business rule.
18. A system as claimed in any preceding claim wherein an element of the knowledge base of the system is applied in more than one application.
19. A method for processing language independent statements composed in natural language, comprising: receiving a language independent natural language statement with a natural language processor and analyzing the statement with the natural language processor with reference to a linguistic base, and storing elements of the natural language statement in a knowledge base memory; identifying with the natural language processor within the natural language statement at least one concept element and at least one associated relation element having a logical relationship with the concept, linking the at least one relation with the associated concepts for storage in the knowledge base memory, and generating executable logic code representation with a code generator to

map the code representation associated with the concept and relation element representations to the natural language statement.

20. A method as claimed in claim 19 wherein data of the elements stored in the knowledge base memory comprise of the concept element having a natural language expression and an assigning a concept tag, and the relation having a natural language phrase and an assigning a relation tag and a concept link tag to linking the relation to the associated concept.
21. A method as claimed in claim 20 wherein the concept element has at least two associated natural language expressions.
22. A method as claimed in claim 20 or 21 wherein the relation element has at least two associated natural language expressions.
23. A method as claimed in claim 21 or 22 wherein the natural languages of the expressions and the phrases are in different natural languages.
24. A method as claimed in any of preceding claims 19-23 wherein analyzing of the received natural language statement comprises analyzing the received natural language statement with a lexical analyzer and a syntax analyzer within the natural language processor.
25. A method as claimed in any of preceding claims 19-24 further comprises resolving ambiguity between different interpretations of relation and concept elements with an ambiguity resolver with the natural language processor.
26. A method as claimed in claim 25 wherein resolving the ambiguity, updating the linking between the relation and concept elements to resolve the ambiguity with the ambiguity resolver and storing the contextual knowledge in the knowledge base memory.
27. A method as claimed in any of preceding claims 19-26 wherein the linguistic base storage comprises at least one language module comprising a construct source, dictionary, and grammar source.
28. A method as claimed in claims 27 wherein the at least one language module comprises an English language module and a Chinese language module.
29. A method as claimed in any of preceding claims 19-28 wherein the received natural language statement is parsed and reduced to a single concept element.
30. A method as claimed in any of preceding claims 19-29 wherein storing the concept and relation elements are stored in separate storage in the knowledge base memory.

31. A method as claimed in any of preceding claims 19-30 further comprises binding an executable module with a relation to associate the relation with the natural language statement.

32. A method as claimed in any of preceding claims 19-31 further comprises receiving the executable logic code representation with a rule engine and mapping the code representation associated with the concept and relation element to the natural language statement.

33. A method as claimed in claim 32 wherein the natural language statement is a business rule, and the rule engine is a business rules management system.

34. A method as claimed in claim 33 wherein the rule engine further comprises a rule base for storing the business rule.

35. A method as claimed in any of preceding claims 19-34 further comprises permitting access to an element of the knowledge base of the system from more than one application.

36. A computer program product comprising a computer readable medium having thereon: computer program code means, when loaded on a computer to make the computer perform a method for processing language independent statements composed in natural language, comprising: receiving a language independent natural language statement with a natural language processor and analyzing the statement with the natural language processor with reference to a linguistic base, and storing elements of the natural language statement in a knowledge base memory; identifying with the natural language processor within the natural language statement at least one concept element and at least one associated relation element having a logical relationship with the concept, linking the at least one relation with the associated concepts for storage in the knowledge base memory, and generating executable logic code representation with a code generator to map the code representation associated with the concept and relation element representations to the natural language statement.

37. A computer readable medium, having a program recorder thereon, where the program when loaded on a computer to make the computer perform a method for processing language independent statements composed in natural language, comprising: receiving a language independent natural language statement with a natural language processor and analyzing the statement with the natural language processor with reference to a linguistic base, and storing elements of the natural language statement in a knowledge base memory; identifying with the natural language processor within the natural language statement at least one

concept element and at least one associated relation element having a logical relationship with the concept, linking the at least one relation with the associated concepts for storage in the knowledge base memory, and generating executable logic code representation with a code generator to map the code representation associated with the concept and relation element representations to the natural language statement.