(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 March 2010 (04.03.2010)

PCT

(10) International Publication Number
WO 2010/022505 A1

(51) International Patent Classification:
G06F 17/27 (2006.01) *G06N 5/02* (2006.01)
G06F 17/28 (2006.01)

(21) International Application Number:
PCT/CA2009/001185

(22) International Filing Date:
28 August 2009 (28.08.2009)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/092,973 29 August 2008 (29.08.2008) US

(72) Inventors; and
(71) Applicants : **SWEENEY, Peter** [CA/CA]; 56 Kilbirnie Court, Kitchener, Ontario N2R 1B8 (CA). **BLACK, Alexander David** [CA/CA]; 70 Mt. Hope Street, Kitchener, Ontario N2G 2J4 (CA).

(74) Agent: **DE FAZEKAS, Anthony**; Miller Thomson LLP, 40 King Street West, Suite #5800, Toronto, Ontario M5H 3S1 (CA).

(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, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

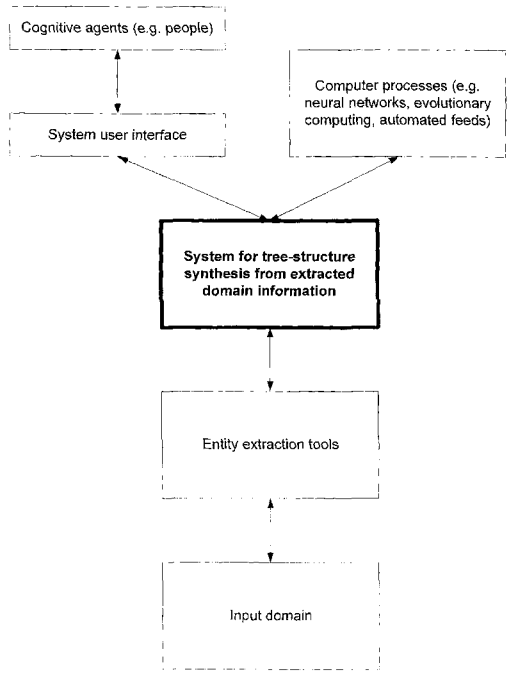
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS,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

Published:

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

(54) **Title**: SYSTEMS AND METHODS FOR SEMANTIC CONCEPT DEFINITION AND SEMANTIC CONCEPT RELATIONSHIP SYNTHESIS UTILIZING EXISTING DOMAIN DEFINITIONS

FIGURE 2



(57) **Abstract**: Computer-implemented systems and methods for synthesis of concept definitions and concept relationships from a domain of data, utilizing different semantic processing protocols such as formal concept analysis and faceted classification synthesis from existing domain concepts that have a confidence gradient built into them. A cognitive or an input agent provides an input of an active concept which is matched against existing domain concepts. The resultant pool of relevant domain concepts is then used to derive virtual concept definitions using a semantic processing protocol. The derivation is then overlaid with a concept of relative proximity of an attribute from another within an attribute set. An additional layer of coherence is given by the relative proximity measure. The end result is a pool of related virtual concept definitions in a tree structure.

WO 2010/022505 A1

SYSTEMS AND METHODS FOR SEMANTIC CONCEPT DEFINITION AND SEMANTIC CONCEPT RELATIONSHIP SYNTHESIS UTILIZING EXISTING DOMAIN DEFINITIONS

FIELD OF THE INVENTION

Embodiments of the invention relate to a computer system and computer-implemented method for processing natural language textual data to provide therefrom concept definitions and concept relationship synthesis using a semantic processing protocol in support of building semantic graphs and networks.

BACKGROUND OF THE INVENTION

A semantic network is a directed graph consisting of vertices, which represent concepts, and edges which represent semantic relationships between concepts. Semantic networking is a process of developing these graphs. A key part of developing semantic graphs is the provision of concept definitions and concept relationships. The present invention addresses this issue.

A semantic network can, in essence, be viewed as a knowledge representation. A knowledge representation is a way to model and store knowledge so that a computer-implemented program may process and use it. In the present context, specifically, knowledge representation may be viewed as a rule-based modeling of natural language from a computational perspective. The substantive value of a knowledge representation is accumulative in nature and as such increases with the amount of knowledge that can be captured and encoded by a computerized facility within a particular model.

One problem associated with an unbounded knowledge representation, is that current systems may impose significant barriers to scale. This is one reason why knowledge representations are often very difficult to prepare. Further, their technical complexity and precision may impose intellectual and time constraints that limit their generation and use. Further, existing systems are generally directed to the analysis and retrieval of knowledge representation from existing forms such as documents and unstructured text. With these analysis and retrieval systems, the amount of knowledge extracted is necessarily limited to the amount of knowledge that was captured in the

existing forms. They may not include all the potential for new knowledge that may be derivable from these documents.

As an example of these problems, consider the following application, typical of the current approach: A product support knowledge base comprising a collection of documents is made available to customers to address their questions about one or more products. The documents are annotated by the publisher with semantic data to describe in minute, machine-readable detail the subject matter of the documents. These documents are then made available through a search tool to provide the customers with the documents most relevant to their queries.

The problem with this application is that the breadth of knowledge encapsulated by the system is bounded by the documents contained within the knowledge base (as expressed through the explicit semantic representations of concept definitions and relationships). People, however, are able to create new knowledge that is inspired by the documents that they read. Continuing the example above, as customers read documents that are related to their needs, they are able to extrapolate from this existing knowledge into the very precise solutions they seek to their problems, creating new knowledge in the process. Unfortunately, there does not yet exist a technical solution that mirrors in a computer-implemented system this process of conceptual extrapolation. The publishers can only describe the knowledge they possess; they cannot provide a system of knowledge representation that encapsulates all the knowledge that might be required, or deduced, by their customers.

Therefore, great significance and associated business value for provisioning new concepts and concept relationships lies in pushing through these barriers to automate the scaling and proliferation of knowledge representations into brand new application areas. One way to distinguish between existing and new applications is that whereas existing applications might answer, "What knowledge is contained in these documents?", new applications might answer, "What knowledge can we generate next?" Among the technical barriers to achieving such knowledge creation applications is the provisioning of new mechanisms to define and capture concepts and concept relationships.

SUMMARY

There are various aspects to the systems and methods disclosed herein. Unless it is indicated to the contrary, these aspects are not intended to be mutually exclusive, but can be combined in various ways that are either discussed herein or will be apparent to those skilled in the art. Various embodiments, therefore, are shown and still other embodiments naturally will follow to those skilled in the art. An embodiment may instantiate one or more aspects of the invention. Embodiments, like aspects, are not intended to be mutually exclave unless the context indicates otherwise.

One aspect of the inventive concepts is a computer-implemented method to synthesize concept definitions and relationships, such as from a natural language data source, that comprises obtaining an active concept definition, matching the active concept definition to a plurality of extracted real concept definitions within a domain, analyzing the real concept definitions for coherence within their attributes and deriving a plurality of virtual concept definitions from the real concept definitions by semantic processing, such that the derived virtual concept definitions form a hierarchical structure.

Another aspect is a computer-implemented method to synthesize concept definitions and relationships, that comprises obtaining an active concept definition, matching the active concept definition to a plurality of extracted real concept definitions comprising attributes within a domain, analyzing the real concept definitions for coherence within their attributes and deriving a plurality of virtual concept definitions from the real concept definitions by semantic processing, such that the derived virtual concept definitions form a hierarchical structure.

Yet another aspect is a machine-readable medium containing executable computer-program instructions which, when executed by a data processing system causes said system to perform a method, the method comprising obtaining an active concept definition, matching the said active concept definition to a plural number of extracted real concept definitions comprising of attributes within a domain, the said real concept definitions analyzed for coherence within their attributes and deriving a plural number of virtual concept definitions from the real concept definitions by semantic processing such that, the derived virtual concept definitions form a hierarchical structure.

Further aspects include computer systems for practicing such methods. For example, an additional aspect is a semantic data processing computer system comprising:

at least one tangible memory that stores processor-executable instructions for synthesizing concept definitions and relationships; and at least one hardware processor, coupled to the at least one tangible memory, that executes the processor-executable instructions to: obtain an active concept definition; extract a plural number of real concept definitions that comprise of attributes from a domain and analyze them for coherence within their attributes; match the said active concept definition to the extracted real concept definitions; and derive a plurality of virtual concept definitions from the real concept definitions semantic processing such that the derived virtual concept definitions form a hierarchical structure.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates the prior art status;

FIG. 2 illustrates incorporation and insertion of tree structure synthesis within the prior art schema, in accordance with some embodiments of the invention;

FIG. 3 gives a flow diagram of the process for identifying new concepts and concept relationships, in accordance with some embodiments;

FIG. 4 gives a flow diagram of the staging and analysis phase in accordance with some embodiments of the invention;

FIG. 5 gives a flow diagram of the synthesis phase in accordance with some embodiments of the invention;

FIG. 6 gives the facet attribute hierarchy for the example where the faceted classification synthesis protocol is implemented; and

FIG. 7 is a diagram of a computer system in which some embodiments of the invention may be implemented.

DETAILED DESCRIPTION OF THE INVENTION

Visual Basic and Windows are registered trademarks of Microsoft Corporation in the United States and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

There are disclosed herein a method, system and computer program providing means for provisioning concept definition and concept relationship synthesis. These aspects of the invention capitalize on the properties of tree structures and a semantic representation that models the intrinsic definition of a concept. As such, new concepts and concept relationships may be created in a way that is not constrained by any historical or existing knowledge representation. Thus, some embodiments of the present invention provide for a new, creative and user-directed expression of semantic representation and networking (graphs). This results in an ability to synthesize forward-looking knowledge, not merely the extraction of historical knowledge.

A practical utility of this approach may comprise a whole or part of a brainstorming session, developing insights by uncovering new concepts from existing knowledge in the aid of creative writing, carving of journalistic research from a huge corpus of text documents, and in general any directed research or study which may involve developing new insights from a given corpus of text-based linguistic data. Embodiments of the inventions generate, from a domain of data, virtual concept definitions and relationships between virtual concept definitions (e.g., a hierarchy of virtual concept definitions). In some embodiments, the virtual concept definitions and their relationships may be provided to a user to aid in the activities discussed above. In other embodiments, the virtual concept definitions and their relationships may be provided to document processing/generation software which uses these definitions to aid in the automatic generation of document or to facilitate manual generation of such documents.

In some embodiments, an active concept is entered or acquired by a cognitive (e.g., human and/or software) agent and relevant real concept definitions are extracted from data representing a particular knowledge domain. The extracted definitions are computer-analyzed for their attribute set coherence within the context of the active concept definition. Attribute sets are then selected from the extracted real concept definitions and a concept synthesis process derives virtual concept definitions based upon selected attribute sets. These derived virtual concept definitions are then assembled into hierarchies. The remaining extracted real concept definitions are then computer-analyzed against the derived virtual concept definition hierarchy and if any further virtual concept definitions can be derived, then the process is repeated. The semantic protocols

exemplified in the context of the present invention are formal concept analysis and faceted classification synthesis. In addition, various overlays that affect selection of attributes such as attribute co-occurrence and relative proximity are incorporated. Further, various numerically oriented limitations in the derivations of virtual concepts are also incorporated.

One way to provide for concept definitions and concept relationships is by extraction of concept definitions from existing documents. However, this may be limited by what is already encoded in the documents and it does not provide for new concept synthesis. As such, extracted semantic representations may act only as a basis for a subsequent process of data transformation that produces a synthesis of new concept definitions and new concept relationships.

Extraction of concepts may be understood, for example, with reference to U.S. Patent Application 11/540,628 (Pub. No. US 2007-0078889 A1). In that application, Hoskinson provides for extraction of concepts from existing documents. An information extraction facility extracts text and then extracts keywords from captured text. The keywords are extracted by splitting text into a word array using various punctuation marks and space characters as separators of words, such that each element in the array is a word Subsequently, the process generates a keyword index from the word array by removing all words in the word array that are numeric, are less than two characters, or are stopwords (e.g., and, an, the an, etc). All the remaining words are included in the keyword index. Once the keyword index is generated, words in the keyword index that occur at least a threshold number of times are retained in the index, while words that occur less than the threshold hold number of times are removed from the index. The keyword index may be further identify key phrases in the text. These key phrases may be viewed as equivalent to the concepts referred to in the present disclosure. Sets of key phrases associated with keywords that provide a context for the key phrases may be viewed as equivalent to the existing concept definitions referred to in the present disclosure.

Hoskinson describes identifying key phrases using the keyword index and document text as follows. First, the document text is analyzed and punctuation symbols that are associated with phrase boundaries are replaced with a tilde character. Next, a character array is generated by parsing the document into strings that are separated by

space characters. Each element in the array is either a word or a phrase boundary character (i.e., a tilde character). Next, the process enumerates through the character array, and determines whether each element is a keyword that appears in the keyword index. If an element is not a keyword, it is replaced with a phrase boundary (i.e., tilde) character. The array elements are then concatenated into a character string, where each character string is delineated by the phrase boundary. It is then determined if each character string is a single word or a phrase. If it is a phrase, it is considered to be a keyphrase, and is added to the keyphrase dictionary.

It should be appreciated that the above-described technique for extracting concepts from documents is one illustrative technique for concept extraction. Many other techniques may be used and the invention is not limited to using this or any other particular technique.

Further, existing concept definitions that are extracted from a domain or corpus of data may be used as a measure of coherence of various attributes sets (combinations of different attributes). Inputs that are active concepts are entered by cognitive agents such as people or machine based expert systems and processed through data analysis or a semantic processing protocol in order to procure existing concepts and relationships covering the context of the active concept within a domain. The existing concepts, also known as real concept definitions, provide a basis to build virtual concepts and their subsequent relationships around the active concept. Fig. 1 represents the prior art approach, wherein a cognitive or input agent interacts with a domain date set via semantic analysis and extraction. In contrast, the at least some of the processes disclosed herein envisage, as shown in Fig. 2, the interaction of a cognitive agent (such as a person) or an input agent via a user interface through extraction of existing domain resources and the use of tree-structure synthesis to construct new concept definitions based upon existing definitions within a domain of data. The input or cognitive agent could further be computer processes like neural networks or evolutionary computing techniques. A tree-structure synthesis creates graphs of concepts and concept relationships that may be limited to a particular context.

One semantic processing protocol that may be utilizable to implement tree-structure synthesis is formal concept analysis. Formal concept analysis may be viewed as a principled way of automatically deriving a formal representation of a set of concepts

within a domain and the relationships between those concepts from a collection of objects and their properties (attributes). Other semantic processing protocols that may be used to implement tree-structure synthesis are formal concept analysis, faceted classification synthesis, and concept inferencing using semantic reasoners. All these approaches are available in the prior art.

EXPLANATION OF KEY TERMS

Domain: A domain is body of information, such as (but not limited to) a corpus of documents, a website or a database.

Attribute: A property of an object.

Attribute set coherence: Attribute set coherence is a measure of the logical coherence of concept attributes when considered as a set within a concept definition structure.

Content Node: Comprises of any object that is amenable to classification, such as a file, a document, a portion of a document, an image, or a stored string of characters.

Hierarchy: An arrangement of broader and narrower terms. Broader terms may be viewed as objects and narrower terms as attributes.

Tree Structures: Trees are like hierarchies comprising directed classes and subclasses, but using only a subset of attributes to narrow the perspective. An organizational chart can be seen as an example of a tree structure. The hierarchical relationships are only valid from perspective of job roles or responsibilities. If the full attributes of each individual were considered, no one would be related hierarchically.

Concept Definition: Semantic representations of concepts defined structurally in a machine- readable form are known as concept definitions. One such representation structures concepts in terms of other more fundamental entities such as concept attributes. A concept definition has its own hierarchy, with a concept as parent and attributes as children. Attributes may in turn be treated as concepts, with their own sets of attributes. Concepts may be associated with specific content nodes.

Concept Synthesis: Concept synthesis is the creation of new (virtual) concepts and relationships between concepts.

Confidence Gradient: The gradient refers to an ordered range of values while confidence may be referred to as a metric used in algorithms to assess the probability that one set of attributes is more coherent than others. So the composition "confidence gradient" might

refer to a declining or elevating confidence level within a group of attribute sets as well as an ordered increase or decrease of the confidence metric within an attribute set with the count of each single attribute starting from general to specific. The confidence may be calibrated using a number of properties of attributes. Two frequently used ones are relative proximity between selected attributes and co-occurrence of two attributes in a set of concept definitions. Another possible measure of confidence would involve overlaying of relative proximity over co-occurrence.

Faceted Classification Synthesis: Faceted classification synthesis allows a concept to be defined using attributes from different classes or facets. Faceted classification incorporates the principle that information has a multi-dimensional quality and can be classified in many different ways. Subjects of an informational domain may be subdivided into facets to represent this dimensionality. The attributes of the domain are related in facet hierarchies. The materials within the domain are then identified and classified based on these attributes. The "synthesis" in faceted classification synthesis refers to the assignment of attributes to objects to define real concepts.

According to one aspect of the disclosed systems and methods, there is shown a synthesis of concepts and hierarchical relationships between concepts, using relevant real (existing) concept definitions within a domain by deriving virtual concept definitions from the existing relevant real concept definitions. The act of deriving a virtual concept definition may be performed utilizing a number of semantic processing protocols that are known in the prior art, such as FCA and faceted classification synthesis, or that may subsequently become known..

With reference to Fig. 3 and Fig. 4, an active concept (AC) is entered or acquired from a cognitive agent and relevant real concept definitions are extracted from a domain. The extracted definitions are analyzed for their attribute-set coherence within the context of the AC definition. Attribute sets are selected from the extracted real concept definitions and a concept synthesis process derives virtual concept definitions based upon selected attribute sets. These derived virtual concept definitions are then assembled into hierarchies. The remaining extracted real concept definitions are then analyzed against the derived virtual concept definition hierarchy and if any can be utilized to construct further virtual concept definitions then the process is repeated again. It is of note that the initial part the overall tree synthesis process, given by Fig. 3, can be seen as a staging and

analysis phase given by Fig 4. The synthesis phase of the overall process can be seen as comprising, for example, the process of Fig 5.

Fig. 7 is a diagram of a computer system on which the processes shown in Figs. 3-5 may be implemented. In Fig. 7, a system for tree-structure synthesis from extracted domain information may receive input information from an input domain and may receive an input active concept definition from a cognitive agent (e.g., a human user) via a system user interface and/or external computer processes. The system for tree-structure synthesis from extracted domain information comprises at least one hardware processor (e.g., a central processing unit (CPU) coupled to at least one tangible storage memory. The system may also comprise an input/output interface (not shown) for receiving the information from the input domain and the cognitive agent(s)/computer processes. Once the cognitive agent and/or computer processes have provided the active concept definition to the system for tree-structure synthesis, the system for tree structure synthesis may perform the remainder of the steps in the example process of Figures 3-5.

FORMAL CONCEPT ANALYSIS

In a further aspect, a way to derive virtual concept definitions in response to an input of an active concept is by formal concept analysis (FCA). If we have real concept definitions Rα and Rβ, with sets of attributes ordered in a confidence gradient which provides a measure of the coherence of the attributes within the concept definitions, given as follows:

Rα = {K1, K3, K2}

Rβ= {K1, K3},

then we have a hierarchy Rβ→Rα. Comparably, with real concept definitions sets Rγ and Rδ, where

Rγ = {K1, K2, K3, K4}

and

Rδ = {K1, K3, K5, K6}

there is no hierarchy between these concepts. In order to construct a hierarchy out of Rγ and Rδ it is necessary to derive virtual Concept Definitions out of Rγ and Rδ using FCA such that the criteria for a hierarchical relationship are satisfied.

So we begin with an input, from an input agent or a cognitive agent, of an AC represented by

R= {K1}.

Identifying R, existing real concept definitions Rγ and Rδ are extracted such that they may have a confidence gradient that ensures integrity, where Rγ and Rδ are represented by

Rγ = {K1, K2, K3, K4}

and

Rδ = {K1, K3, K5, K6}.

Since attributes are occurring within a concept definition containing an active concept, it is assumed that the active concept and other attributes within a virtual concept definition have a contextual relationship with each other, such that the more an attribute co-occurs with an active concept across different concept definitions, the more stronger the said contextual relation. If it is possible to build a virtual concept definition set Vγ with formal concept analysis, such that Vγ has a built-in confidence gradient that may be based upon prevalence of attributes, where

Vγ = {K1, K3};

and if similarly it is possible to build Vδ, such that

Vδ = {K1, K3, K4},

then two virtual concept definitions, Vγ and Vδ, have been created that are in a hierarchical relationship between themselves, Vγ → Vδ, while each individually is in a relationship at the attribute level by virtue of sharing attributes with real concept definition sets Rγ and Rδ.

Example of formal concept analysis building a virtual concept definition with a built-in confidence gradient

Domain Input: (computers, laptop, desktop, servers, software, operating system, software application, CPU, calculators, algorithm, computer language, user interface, machine language)

Let us say that the domain includes the following real concept definitions with their composite attributes such that they have built-in confidence gradient:

R1: {computers, CPU, laptop, desktop, software, calculator}

R2: {computers, servers, software, operating system, software application, algorithm, computer language}

R3: {computers, machine language, software, algorithm}

R4: {software, user interface, software application}

AC= {software}

What is concurrent with the attribute "software"?

computers: 3 times

Algorithm: 2 times

software application: 2 times

laptop: 1 time

desktop: 1 times

servers: 1 time

operating system: 1 time

machine language: 1 time

user interface: 1 time

CPU: 1 time

calculator: 1 time

computer language: 1 time

Counting to find which attribute is concurrent the greatest number of times with the attribute "software", one finds that "computers" is the most prevalent attribute that co-occurs with "software". Thus, V1: {software, computers} is created..

Now the tree looks like the following:

AC: {software}

|

+—V1: {software, computers}

|

+—V2: {software, software application}

|

+—V3: {software, algorithm}

Continuing, recursively, one may determine what is concurrent with "software" and "computers" within the real concept definitions. In this, one finds the following:

Laptop: 1

desktop: 1

servers: 1

operating system: 1

software application: 1

CPU: 1

calculator: 1

algorithm: 2

computer language: 1

machine language: 1

So there is now the following tree:

AC: {software}

|

+—V1: {software, computers}

| |

| +—V4: {software, computers, algorithm}

|

+—V2: {software, software application}

|

+—V3: {software, algorithm}

In the result, V1 and V4 are in a hierarchy and are derived from R1, R2, R3 and R4. For a larger number of real concept definitions with additional attributes it is possible to unfold more hierarchal structures and relationships. If, for a given active concept, the system does not return a sufficient number of real concept definitions in order to derive virtual concept definitions, any number of domains can be searched to achieve the objective. The sufficient number may be considered as a minimum number of domains required to produce at least a selectable depth of one hierarchy within derived virtual concepts or may, additionally, require producing at least a selectable number of hierarchies of derivable virtual concept definitions from a domain. Further, a selectable maximum depth of a hierarchy and a selectable maximum number of hierarchies derived may cap the synthesis process.

Overlaying an additional criterion, namely relative proximity, as a confidence measure in order to build virtual concept definitions can change the virtual concepts derived from the real concept definitions using formal concept analysis. Relative proximity may be referred to as the physical separation of one attribute from another within an attribute set of a concept definition. In the example above, within R2, the attribute "software" is one attribute away from 'computers' and "software application", whereas "software" is two attributes away from "algorithm". In R3, however, "software" is adjacent to "algorithm" or zero attributes away from "algorithm". So one can consider zero as the default relative proximity for "software" and "algorithm" from the existing domain information. If more weight were given to relative proximity and relative proximity were overlaid on the above example, then the virtual concept with a higher confidence measure would come first in the tree. For example, the V1 in this case would be:

V1: {software, algorithm}

because "software" is zero attributes away from "algorithm" while "software" is one attribute away from "computers", so "algorithm" will take precedence over "computers" even though "computers" is co-occurring three times with "software". As such, all virtual concepts will change if the weight of relative proximity shifts the focus from one attribute to another with a higher relative proximity. Further, if between attributes the relative separation is equal, a higher concurrency value will give a higher confidence measure to a derived virtual concept definition. The logic behind giving more weight to relative proximity than concurrency is that relative proximity is directly observable from an existing real concept definition which is a graduated set in terms of coherence within concept definitions.

The sets R1 through R4 in the above example are associated sets. If the real concept definitions are disjoint sets, that is, if none of the attributes of the real concept definitions overlap, then the data transformation is as follows:

Let the disjoint real concept definitions sets be:

R5: {1, 2, 3, 4, 5}

R6: {6, 7, 8, 9, 10}

If the Active Concept is:

AC: {2, 8}

then, applying formal concept analysis to derive virtual concept definitions would give us the following {2, 1}, {2, 3}, {2, 4}, {2, 5}, {8, 6}, {8, 7}, {8, 9} and {8, 10}. Further, overlaying relative proximity would shorten the list to {2, 1}, {2, 3}, {8, 7} and {8, 9}. The disassociated real concept definitions give rise to separate legs (or lineages) of virtual concept definitions each representing the related part of the active concept in question. The analysis iterates over the number of times required to exhaust the list of attributes within the real concept definitions. The derivation of virtual concept definitions is bounded by the confidence as measured by concurrency and relative proximity as detailed above. It is also of note that one can tune these weighting measures in order to achieve the desired scope of a result, that is, to change relative proximity measures to expand or contract the resulting volume of virtual concept definitions.

FACETED CLASSIFICATION SYNTHESIS

In a further aspect of this disclosure, a way to derive virtual concept definitions in response to an input of an active concept may be implemented by using faceted classification synthesis (FCS) which is based on a structure of facets and attributes that exists within a domain. Fig. 6 is a good example.

Domain Input: (computer, laptop, desktop, servers, software, Windows®, Linux®, operating system, software application, CPU, calculator, algorithm, computer language, user interface, machine language, C, Visual Basic®, C++, HTML)

In this example the domain includes the following facets, built by FCS, with their composite attributes such that they have built-in confidence gradient as followed by the classification structure.

F11: {computer, servers}

F12: {computer, calculator}

F13: {computer, laptop}

F14: {computer, desktop}

F211: {software, operating system, Windows}

F212: {software, operating system, Linux}

F221: {software, software application, user interface}

F222: {software, software application, algorithm}

F2311: {software, computer language, C, C++}

F232: {software, computer language, machine language}

F233: {software, computer language, Visual Basic}

F234: {software, computer language, HTML}

All the facet attribute sets and the number indices (for example F233) listed above in the current example refer to a unique path within the facet attribute hierarchies, with any attribute inheriting all the prior attributes above it. The unique path refers to the index path with reference to Fig. 6. The index 1 at first position from left refers to computers while index 2 in the first position refers to software. Moving on, the next index number refers to inherited attribute one level below and the third index number refers to the attribute further below. The index path ensures only one path for an attribute entry in Fig. 6. Let real concept definitions based upon the facet attribute sets be the following:

IBM PC: {desktop, Windows}

ThinkPad: {laptop, Linux}

Webpage: {servers, HTML, UI}

Browser: {desktop, operating system, software application, computer language}

Web calculator: {server, HTML, software application}

Calculation: {calculator, machine language}

If an active concept is entered as following:

AC: {operating system, computer language}

then virtual concept definitions may be derived from the given real concepts using faceted classification synthesis inheritance bounds and overlaying with relative proximity (with zero and one separation). In deriving the virtual concept definitions, faceted classification synthesis rules allow the substitution of a parent attribute with a child within an attribute hierarchy. The implementation of these faceted classification synthesis substitution rules can be made optional in performing the synthesis. The substitution rule is applied in the example below. The results are as follows:

V1: {operating system, software application, computer language}

V2: {software application, computer language}

V3: {software application, HTML}

V4: {software application, C}

V5: {software application, C++}

V6: {software application, Visual Basic}

V7: {desktop, operating system, software application}

V8: {desktop, operating system, software application, computer language}

V9: {server, HTML}

V10: {server, HTML, software application}

V11: {server, HTML, UI}

V12: {desktop, Windows}

V13: {laptop, Linux}

V14: {desktop, Linux}

V15: {laptop, Windows}

V16: {calculator, machine language}

In the outcome, it is noted that many of the virtual concept definitions are arranged in a hierarchy. At all times, the confidences of the derived concept definitions remain intact, as they are in the existing domain, as the faceted classification synthesis inheritance path is strictly taken into account while deriving the virtual definitions. If the domain facet attribute sets are deeper than the example given here then one may set relative proximity greater than one. Additional virtual definitions are then derivable with deeper structures. The minimum and maximum number of derived virtual concept definitions and the attributes within are selectable in faceted classification synthesis as discussed above.

In addition, limits on the derivation of virtual concept definitions, in any form of semantic processing, may also be based on a confidence gradient or on additional qualitative aspects, such as (and not limited to) having every concept be a possible ancestor of at least one real concept or having no concept with the same descendant set as its parent.

If the domain objects defined as real concept definitions are such that a group of them is exclusively drawing attributes from a certain group of facet attribute sets and another group of real concept definitions is drawing attributes from a different group of facet attribute sets (having disjoint real concept definitions) then the active concept will go through the first group of real concept definitions and then any other disassociated group one at a time until all disjoint groups of real concept definitions are exhausted. As

always, caps are selectable based upon a number of properties or just an arbitrary number to limit the active concept going through real concept definitions.

Another interesting outcome of the synthesis process is the resulting simple and broader concepts such as "binning" which might not be readily available in the extracted real definitions. Bins, generally, are concepts that group a number of other concepts based on one or more common (shared) attributes, derived in whole from multiple real concepts such as V1:{software, computers} in the discussion of formal concept analysis.

In all aspects of the present inventions the unique combination of tree-structure classification with concept synthesis provides a far greater number of structurally pared-down virtual concept definitions and their relationships when compared to the existing real concept definitions extracted in the context of the active concept in focus. This is essentially the main objective of tree-structure synthesis.

The above-described embodiments of the present invention can be implemented in any of numerous ways. For example, the embodiments may be implemented using hardware, software or a combination thereof. When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers. It should be appreciated that any component or collection of components that perform the functions described above can be generically considered as one or more controllers that control the above-discussed functions. The one or more controllers can be implemented in numerous ways, such as with dedicated hardware, or with general purpose hardware (e.g., one or more processors) that is programmed using microcode or software to perform the functions recited above.

In this respect, it should be appreciated that one implementation of the embodiments of the present invention comprises at least one computer-readable storage medium (e.g., a computer memory, a floppy disk, a compact disk, a tape, and/or other tangible storage media.) encoded with a computer program (i.e., a plurality of instructions), which, when executed on a processor, performs the above-discussed functions of the embodiments of the present invention. The computer-readable medium can be transportable such that the program stored thereon can be loaded onto any computer system resource to implement the aspects of the present invention discussed herein. In addition, it should be appreciated that the reference to a computer program

which, when executed, performs the above-discussed functions, is not limited to an application program running on a host computer. Rather, the term computer program is used herein in a generic sense to reference any type of computer code (e.g., software or microcode) that can be employed to program a processor to implement the above-discussed aspects of the present invention.

It should be appreciated that in accordance with several embodiments of the present invention wherein processes are implemented in a computer readable medium, the computer implemented processes may, during the course of their execution, receive input manually (e.g., from a user), in the manners described above.

Having described several embodiments of the invention in detail, various modifications and improvements will readily occur to those skilled in the art. Such modifications and improvements are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description is by way of example only, and is not intended as limiting. The invention is limited only as defined by the following claims and the equivalents thereto.

What is claimed is:

CLAIMS

1. A method of operating a computer to perform a computer-implemented process for synthesizing concept definitions and relationships comprising:

obtaining an active concept definition;

extracting a plurality of real concept definitions comprising attributes from a domain and analyzing them for coherence within their attributes;

matching the said active concept definition to the extracted real concept definitions; and

deriving a plurality of virtual concept definitions from the real concept definitions by semantic processing, such that the derived virtual concept definitions form relationships between themselves.

2. The method of claim 1, deriving additional possible virtual concept definitions using the derived virtual concept definitions.

3. The method of claim 1, wherein the relationships are hierarchical structures.

4. The method of claim 1, wherein a depth of a hierarchy of one derived virtual concept definitions is selectable.

5. The method of claim 4, wherein the selection of the depth of the hierarchy is based upon a confidence gradient.

6. The method of claim 1, wherein a derivable number of virtual concept definitions is limited by quantity.

7. The method of claim 1, wherein the derivable number of virtual concept definitions is based upon a qualitative aspect.

8. The method of claim 7, wherein the qualitative aspect is determined by a confidence gradient.

9. The method of claim 1, further comprising searching a plurality of domains to build a selectable quantity of virtual concept definitions.

10. The method of claim 1, further comprising searching a plurality of domains to build selectable depths of hierarchies of virtual concept definitions.

11. The method of claim 1, wherein the derived virtual concept definitions are in a poly-hierarchal relationship within themselves.

12. The method of claim 1, wherein the existing real concept definitions are used as a measure of a coherence of various attribute sets.

13. The method of claim 1, wherein derived virtual concept definitions are part of a tree structure.

14. The method of claim 1, wherein the derived virtual concept definitions are in a poly-hierarchal relationship with the real concept definitions.

15. The method of claim 1, wherein a scope of the derived virtual concept definitions is variable with respect to a change in a relative proximity measure between attributes in an attribute set.

16. The method of claim 1, wherein the derived virtual concept definitions comprise bins.

17. A method of claim 1, wherein the semantic processing is based upon faceted classification synthesis.

18. The method of claim 17, wherein an attribute substitution rule of replacing parent attributes with child attributes is made optional to synthesize virtual concept definitions.

19. A method of claim 1, wherein the semantic processing is based upon formal concept analysis.

20. A computer-implemented method to synthesize concept definitions and relationships comprising:

obtaining an active concept definition;

extracting a plural number of real concept definitions comprising attributes from a domain and analyzing them for coherence within their attributes;

matching the said active concept definition to the extracted real concept definitions; and

deriving a plural number of virtual concept definitions from the real concept definitions by semantic processing, such that the derived virtual concept definitions form relationships between themselves.

21. The method of claim 20, wherein the relationships are hierarchical structures.

22. The method of claim 20, comprising a further step of a final overlay of a concept of relative proximity that further affects selection of attributes.

23. A machine-readable medium containing executable computer program instructions which when executed by a data processing system causes the said system to perform a method, the method comprising:

obtaining an active concept definition;

extracting a plural number of real concept definitions comprising attributes from a domain and analyzing them for coherence within their attributes;

matching the said active concept definition to the extracted real concept definitions; and

deriving a plural number of virtual concept definitions from the real concept definitions by semantic processing, such that the derived virtual concept definitions form a hierarchical structure.

24. The machine readable medium containing executable computer program instructions of claim 23, wherein the relationships are hierarchical structures.

25. The machine readable medium containing executable computer program instructions of claim 23, wherein the method within comprises a further step of a final overlay of a concept of relative proximity that further affects selection of attributes.

26. A semantic data processing computer system comprising:

at least one tangible memory that stores processor-executable instructions for synthesizing concept definitions and relationships; and

at least one hardware processor, coupled to the at least one tangible memory, that executes the processor-executable instructions to:

obtain an active concept definition;

extract a plural number of real concept definitions that comprise of attributes from a domain and analyze them for coherence within their attributes;

match the said active concept definition to the extracted real concept definitions; and

derive a plural number of virtual concept definitions from the real concept definitions semantic processing such that the derived virtual concept definitions form a hierarchical structure.

27. The system of claim 26, wherein the relationships are hierarchical structures.

28. The semantic data processing system of claim 26, wherein the at least one hardware processor executes the processor-executable instructions to overlay a concept of relative proximity that further affects selection of attributes.

FIGURE 1

PRIOR ART

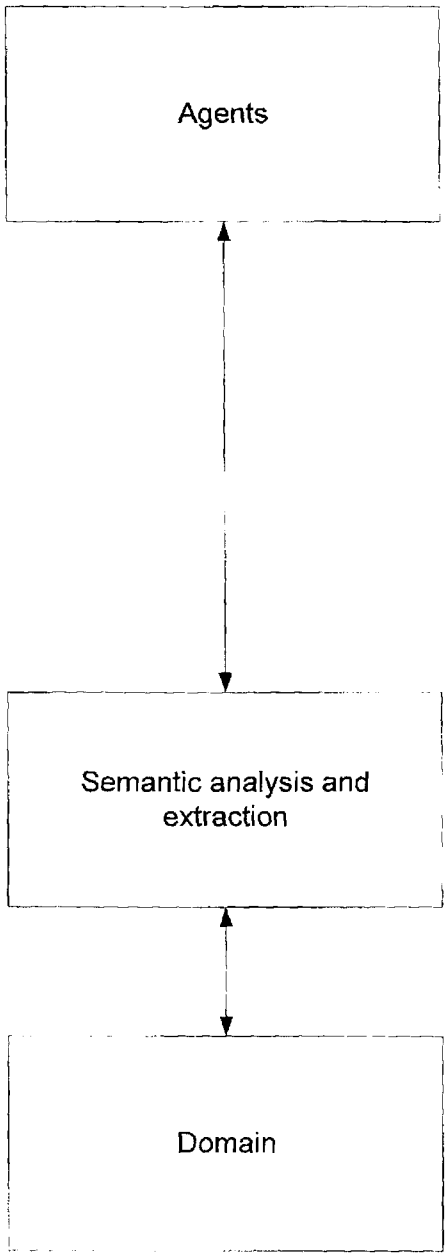


FIGURE 2

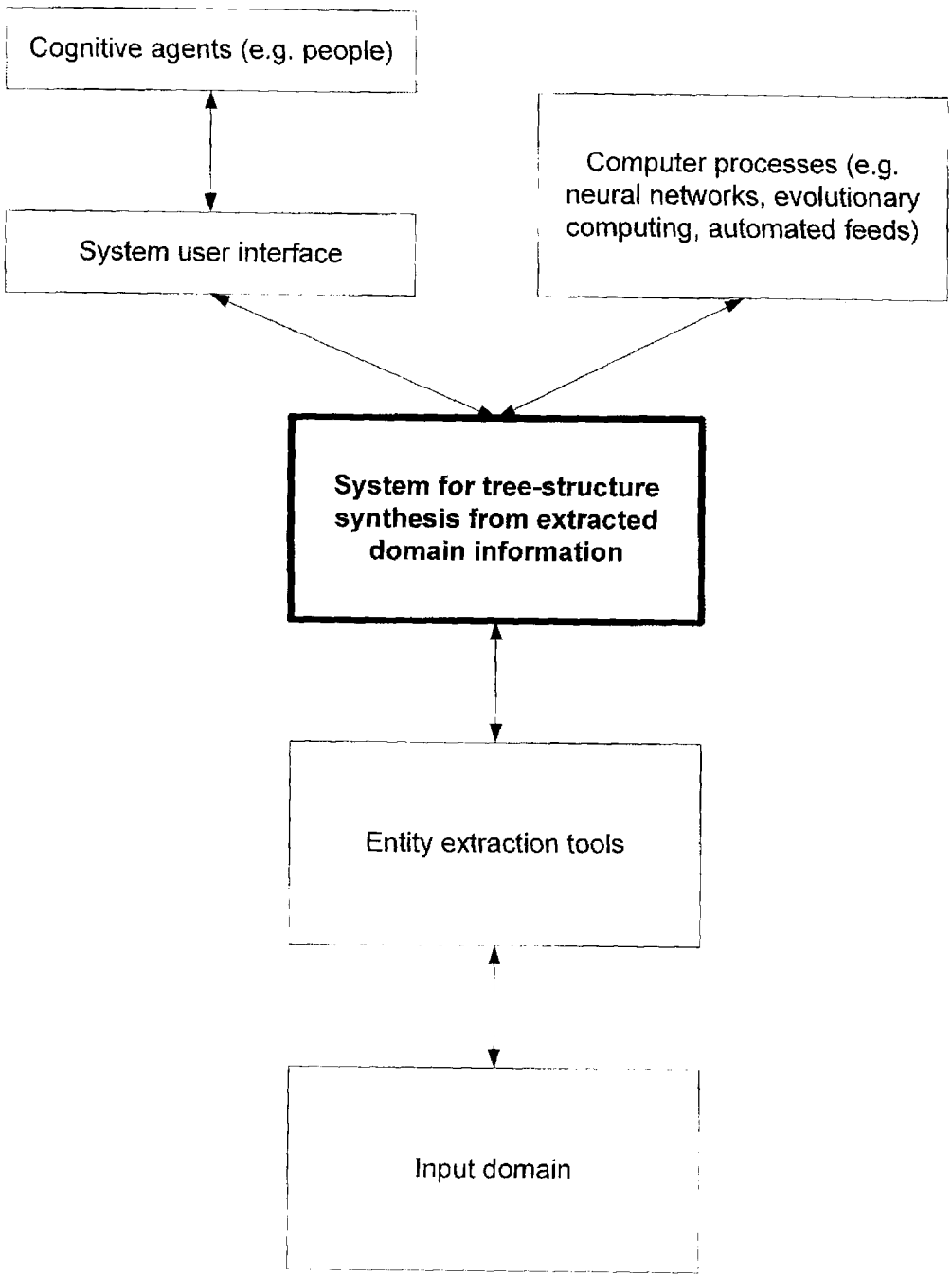

Cognitive agents (e.g. people)
System user interface
Computer processes (e.g. neural networks, evolutionary computing, automated feeds)
System for tree-structure synthesis from extracted domain information
Entity extraction tools
Input domain

FIGURE 3

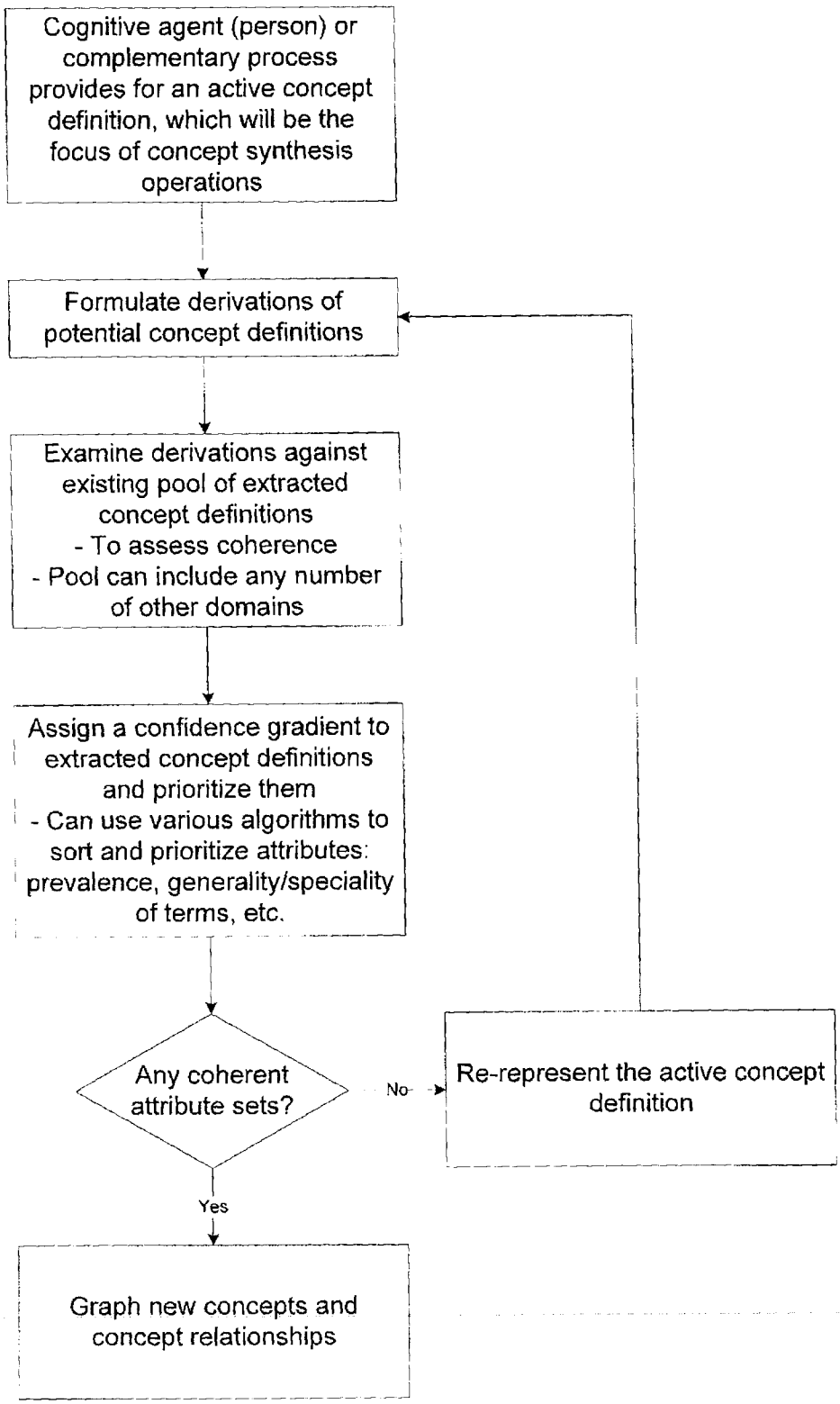

Cognitive agent (person) or complementary process provides for an active concept definition, which will be the focus of concept synthesis operations
Formulate derivations of potential concept definitions
Examine derivations against existing pool of extracted concept definitions
- To assess coherence
- Pool can include any number of other domains
Assign a confidence gradient to extracted concept definitions and prioritize them
- Can use various algorithms to sort and prioritize attributes: prevalence, generality/speciality of terms, etc.
Any coherent attribute sets?
No
Re-represent the active concept definition
Yes
Graph new concepts and concept relationships

FIGURE 4

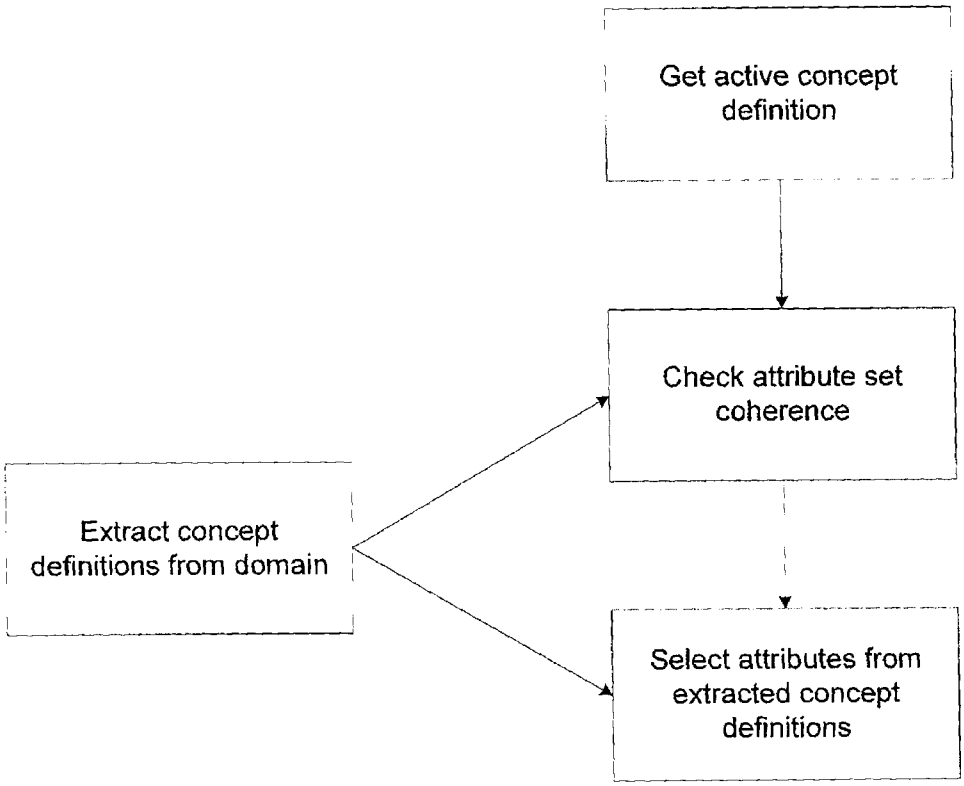

Get active concept definition
Check attribute set coherence
Extract concept definitions from domain
Select attributes from extracted concept definitions

FIGURE 5

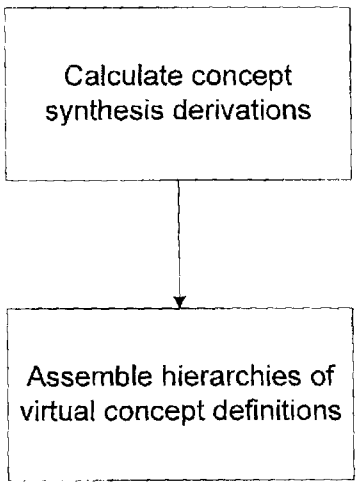

Calculate concept synthesis derivations
Assemble hierarchies of virtual concept definitions

FIGURE 6

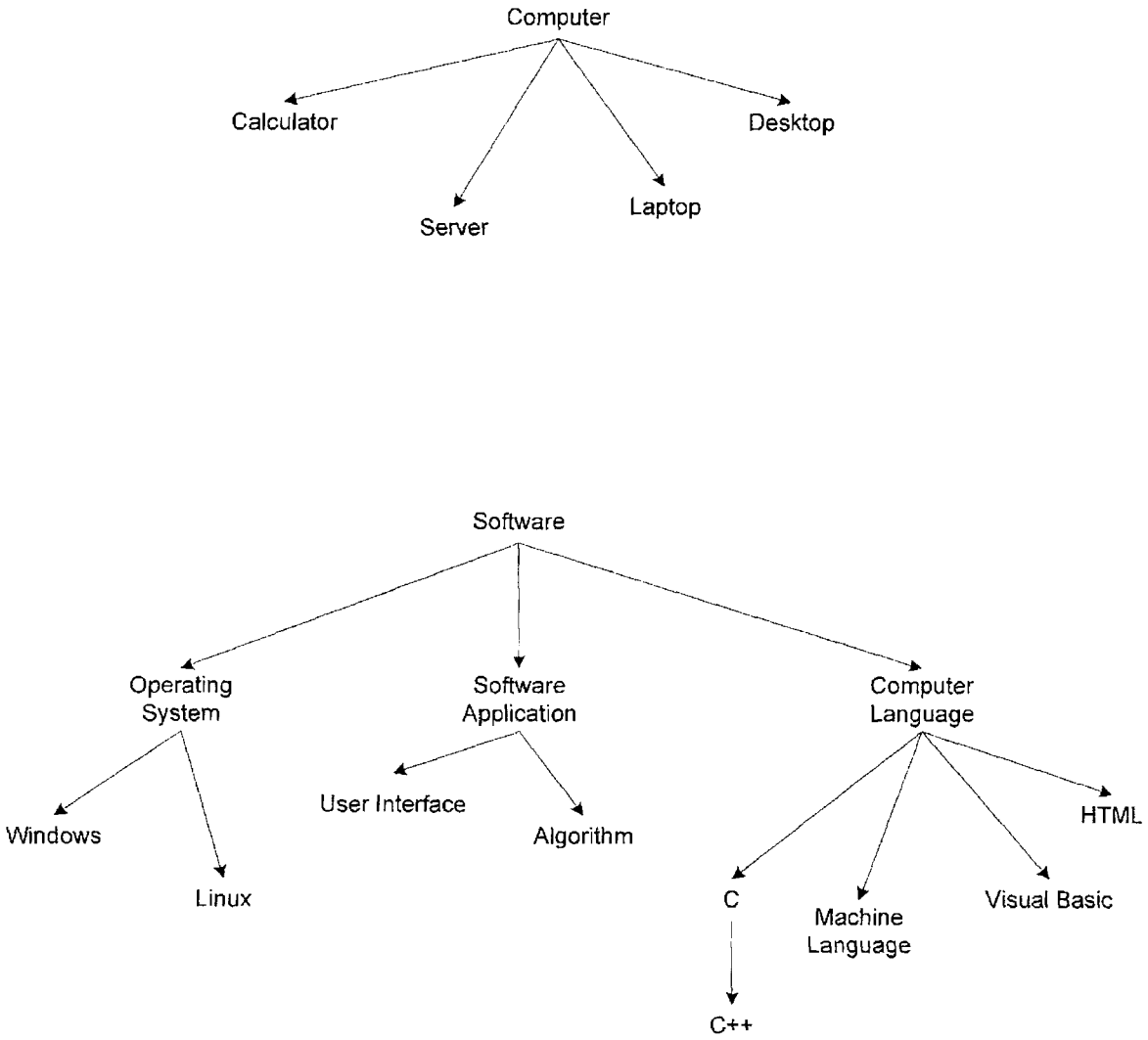
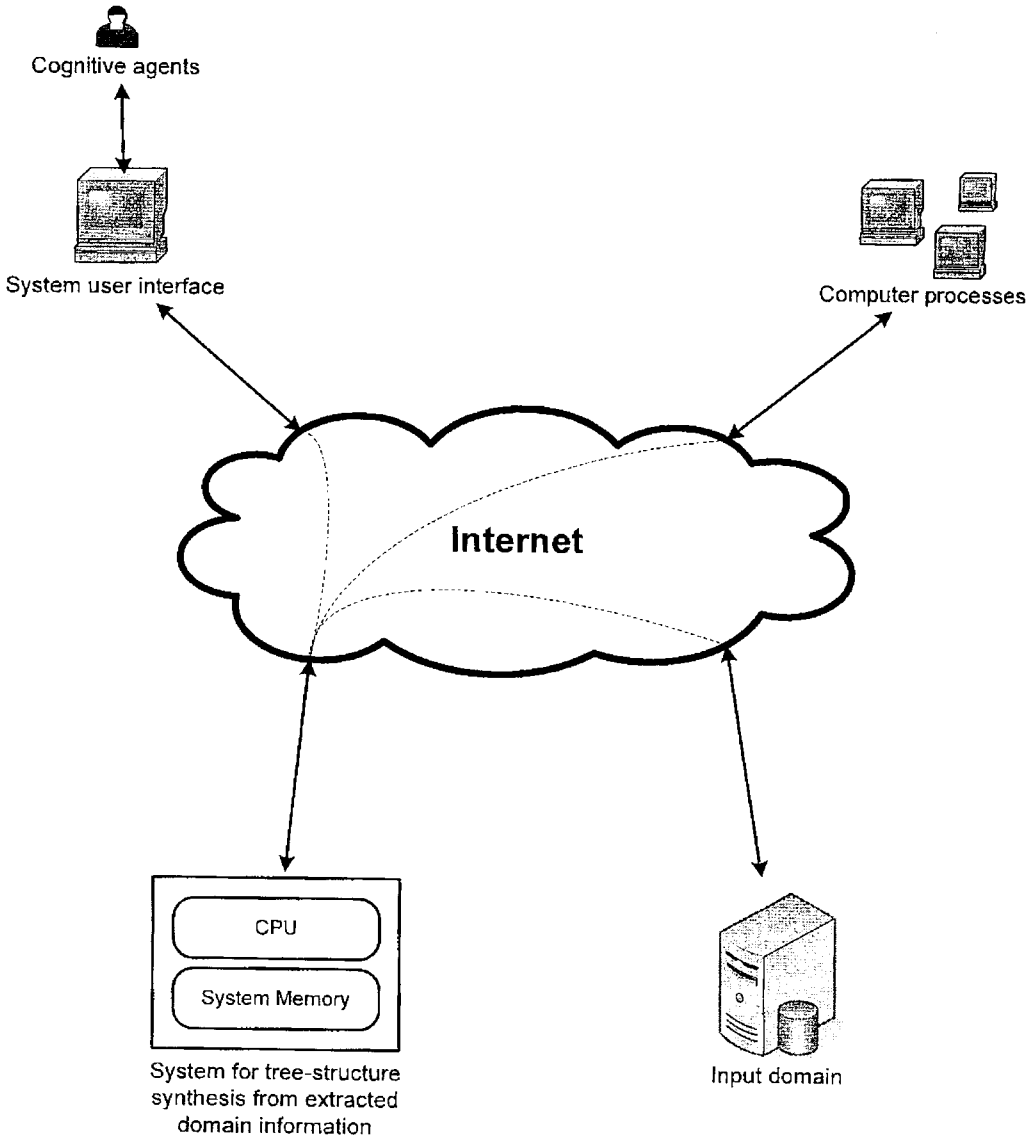

Computer
Calculator
Desktop
Server
Laptop
Software
Operating System
Software Application
Computer Language
Windows
Linux
User Interface
Algorithm
C
C++
Machine Language
Visual Basic
HTML

FIGURE 7


Cognitive agents
System user interface
Computer processes
Internet
CPU
System Memory
System for tree-structure synthesis from extracted domain information
Input domain

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2009/001185

A. CLASSIFICATION OF SUBJECT MATTER
IPC: ***G06F 17/27*** (2006.01) , ***G06F 17/28*** (2006.01) , ***G06N 5/02*** (2006.01)
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: ***G06F 17/27*** (2006.01) , ***G06F 17/28*** (2006.01) , ***G06N 5/02*** (2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Deplhion, IEEE, Epoque, CPD, QPAT (using keywords): concept definition, systhesiz*, semantic, relation*, concept, faceted classification

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2007/0208764 A1 (Grisinger) 6 September 2007 (06-09-2007) Entire document	1-28
A	US 2007/0118542 A1 (Sweeney) 24 May 2007 (24-05-2007) Entire document	1-28
A	US 2007/0174041 A1 (Yeske) 26 July 2007 (26-07-2007) Entire document	1-28

[ ] Further documents are listed in the continuation of Box C.

[X] 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	Date of mailing of the international search report
19 November 2009 (19-22-2009)	3 December 2009 (03-12-2009)

Name and mailing address of the ISA/CA	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001-819-953-2476	Kristy Hyam (819) 934-2673

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2009/001185

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US 2007208764A1	06-09-2007	WO 2007103461A2	13-09-2007
		WO 2007103461A3	13-12-2007
US 2007118542A1	24-05-2007	AU 2007291867A1	06-03-2008
		CA 2662063A1	06-03-2008
		EP 2062174A1	27-05-2009
		US 7596574B2	29-09-2009
		US 7606781B2	20-10-2009
		US 2007136221A1	14-06-2007
		US 2008021925A1	24-01-2008
		WO 2008025167A1	06-03-2008
US 2007174041A1	26-07-2007	CA 2523586A1	11-11-2004
		EP 1623339A2	08-02-2006
		WO 2004097664A2	11-11-2004
		WO 2004097664A3	24-11-2005