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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 February 2006 (02.02.2006)

PCT

(10) International Publication Number
WO 2006/011102 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/IB2005/052360

(22) International Filing Date: 15 July 2005 (15.07.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
04103504.9 22 July 2004 (22.07.2004) EP

(71) Applicant *(for all designated States except US)*: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: ALEKSOVSKI, Zarko [MK/NL]; c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL). TEN KATE, Warner, R., T. [NL/NL]; c/o Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(74) Agents: GRAVENDEEL, Cornelis et al.; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

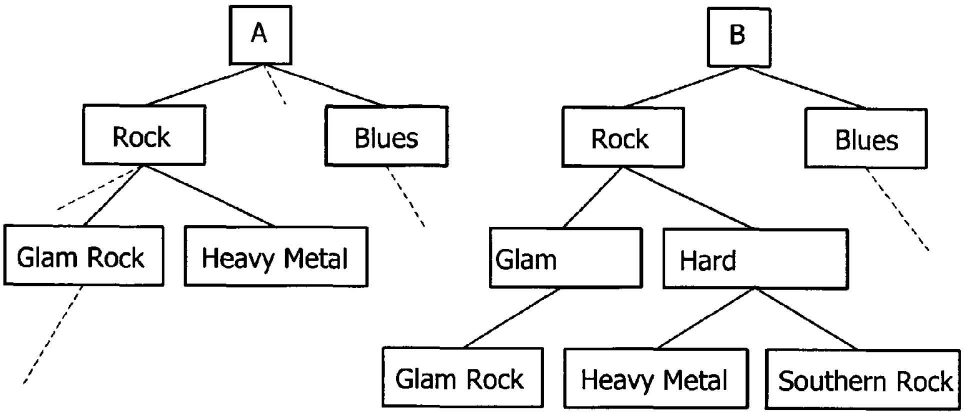
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, 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, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DETERMINING A SIMILARITY BETWEEN ONTOLOGY CONCEPTS



(57) Abstract: A system with a first device 200 storing a first concept A from a first ontology 202 and a second device 210 storing a second concept B from a second ontology 212. Each ontology describes concepts and relations between the concepts. Similarity between the concepts A and B is determined by expressing the similarity as a logical relation between the concepts; converting the first concept A to an expression E_A with sub-expressions $E_{A'}$ each representing respective literal concepts A_i ($i = 1 .. I; I >1$) related to the first concept A according to the first ontology; converting the second concept B to an expression E_B with sub-expressions E_{Bj} each representing respective literal concepts B_j ($j = 1 .. J; J >1$) related to the second concept according to the second ontology; converting the logical relation to logical sub-relations between the subexpressions $E_{A'}$ and E_{BJ} ; and determining the similarity in dependence on how many of the logical sub relations evaluate positively.

Determining a similarity between ontology concepts

FIELD OF THE INVENTION

The invention relates to a method of determining a similarity between concepts of different ontologies.

The invention also relates to a system for performing the method.

BACKGROUND OF THE INVENTION

The advent of coding media in digital format has brought new forms of storing, exchanging and accessing content. In particular, it has introduced multimedia, which combines all sorts of media and provides all sorts of interaction with the content. New digital communications systems, such as the Internet and DVB, have introduced the integration of content from different sources, such as different content providers. It has also eased the way for any party to publish content, both professionals and consumers. Aside from this expansion in opportunities, the number of information items is also growing. The navigation through content is increasingly challenged with the increasing amount of available content, the heterogeneity of content types, and the scale of distribution. Even tracing back some known piece of content can be cumbersome.

Searching a content item using keyword search is a first step, but it requires the user to know the keyword schema or to creatively modify the entered keyword sequences to find the content of interest. Technically, the problem relates to the mismatch between the system which operates at the syntactical level, while the user's cognition is at the semantic level. An approach to bridge this gap would be the introduction of semantics in the machine processes. The Semantic Web is heading towards this direction, see W3C, The Semantic Web, http://www.w3.org/2001/sw/ , [W3CsemWeb].

A main challenge lies in the so-called content management, i.e. the way to store media content so that it can easily, at the semantic level, be found and accessed. For example, it is desired to realize a user interface on a consumer electronics device, such as a set-top box, digital TV or PC, that unites heterogeneous content listings from different providers. Preferably, it integrates listings from the home DVD Disc Manager, Home HDD, Yahoo, and iM Networks. It should present the listings homogeneously, using a single vocabulary, and only those parts that match the user's preference and interest profile. The vocabulary it uses is that of the user, also allowing the user to express his preferences and requests in his own language. A first step in that direction is the use of metadata in the search and selection processes. For example, the content items can be classified according to the metadata they share. It requires that the keywords denoting the metadata are structured in a schema, or at least are part of a shared vocabulary, upon which the search application can base its classification algorithm. Metadata are available from the content through feature extraction, or from supply by an external source.

However, it is unlikely that on the Internet all users and providers will make use of one single metadata schema, if only for the problem of maintaining the schema updated and shared consistently, not to mention the problem of incomplete or erroneous information. A second step, therefore, is to establish an ontology that sufficiently spans the domains of users and providers, such that it can support a mediator which relates user preferences (queries) to the provider's metadata. Fig.1 shows a possible system wherein a mediator 100 has an ontology 102 that is filled with information supplied by/obtained from the content providers 130, 131. The user preferences are reflected as queries 120, 121 to the ontology 102. The mediator answers the queries based on the ontology. The user profiles can be stored and maintained at the user's premises.

As an example scenario, assume that a provider offers music labeled "Evergreens". The songs in the collection are annotated with title and artist name. For example, it includes "Yesterday"/"The Beatles" and "Bridge over Troubled Water"/"Simon and Garfunkel". A device of a user may have an ontology with a class called "Golden Hits". Using the ontology the class is defined as containing songs that were "hits" in the "60s". Further assume that a site exists that publishes the weekly top ten listings. The ontology makes use of them by defining its "hits" concept as the collection of items listed on that site. Finally, the ontology defines the concept "60s" in terms of its concept "compositionDate". (The instance values are determined by additional relations with the same site or with other repositories.) So, the user preference "Golden Hits" is known in terms of the ontology as "listed on site" and "composed in 60s". The "Evergreens" class is known in terms of the ontology as "collection of title/artist pairs". Based on these class definitions, it can be determined whether "collection of title/artist pairs" is a subclass of "listed on top-ten site", and, in a similar fashion, whether it is a subclass of "composed in 60s". If so, it is a subclass of "Golden Hits" and the content is of interest to the user.

The background knowledge, stored in the ontology in Fig.1, is an essential tool in the process. On the Internet, the W3C is standardizing a set of languages that provide, among others, the exchange of ontologies. The set is collectively known as the Semantic Web. One of its interesting features is that it supports the construction of an ontology by including others from the Internet. OWL is the language for ontology exchange, of which the variant OWL-DL is based on Description Logics (DL), see "http://dl.kr.org/". This logic foundation enables the use of standard (DL) reasoning services for inferring implicit statements from the given ontologies.

An ontology is an explicit formal specification that describes a human knowledge domain (such as audio content in the example given above) in terms of concepts (classes) and relations between those concepts. Each concept represents at least one human understandable term. Concepts can be defined in terms of other concepts, using logic constructs as conjunction, disjunction and negation, as well as by specifying restrictions on relationships with other classes. The semantics of the constructs is defined in a model theory, which includes the definition of the entailments that can be made. An example entailment is to infer a subclass relationship between two concepts that is not explicitly modeled in that way. Another example is the following. Assume the ontology contains family relations, such as “brothers share the same father”, and it is further known (asserted) that “Al is the father of John” and “John is the brother of Mike”. Then, the reasoner can infer that “Al is the father of Mike”.

Although the ontology in Fig.1 provides for the background knowledge to find semantic matches between a user's request and the provider's content, the approach still assumes that the user and provider vocabularies are integrated with that ontology. Of course, this is not necessarily the case, apart from the problems of covering all accessible vocabularies and of keeping track of their evolutions. Currently, human intervention is needed to define the mapping between the vocabularies and the ontology. In order to bring the system to its full value, it is needed to introduce an automated manner of (dynamically) integrating vocabularies.

One approach is proposed in P. Bouquet, L. Serafini, and S. Zanobini, Semantic Coordination: A New Approach and an Application, Proc. ISWC2003, Springer, LNCS 2870, 130-145, 2003, [trento]. This article describes an algorithm to find the relationships between terms from different vocabularies. In the first step of the algorithm, each term in the vocabularies is expressed as a logical formula. Subsequently, the so-obtained formulas are compared pairwise by a theorem prover (SAT solver) to decide whether they are related, and, if so, in what manner (equal, more general, more specific, etc.). WordNet is used to generate the logical expressions in the first step, see G.A. Miller et al., WordNet - A Lexical Database for the English Language, “http://www.cogsci.princeton.edu/~wn/”. WordNet is a publicly available taxonomy of the English language. For each term in the vocabulary WordNet is queried for the explanations of that term. The returned explanations are combined as predicates so as to form the logical expression. Under the assumption that the terms in the vocabularies are words that reflect the term’s meaning in English language, the logical formulas indeed capture that meaning in a logical sense. If the assumption does not hold, one needs to use another knowledge base to generate the formulas. For example, in the case of matching music genres, WordNet might be less optimal, or partly contributing to the formula generation, and an additional background knowledge base is needed. The formula generation process is refined in several ways. For example, the position of each term in its vocabulary’s schema is accounted for by logically combining the term’s formula with those of its parents and siblings in the schema. This can be done by bounding (logical $\wedge$) the term to the scope of its parent, and by setting it disjoint (logical $\neg$) from its siblings. The precise way of using this type of position information may depend on the particular domain. For example, in the music domain it is not uncommon that an artist is classified in different, sibling genres. Extending the logic formulas for being disjoint with the sibling classes (genres) would be a too strong statement.

The second step of the algorithm is to decide whether two terms are related to each other, say to decide whether term A is a subClassOf B, $A \subseteq B$. The relation holds if the logical formula $A \Rightarrow B$ is satisfiable, i.e. if A evaluates to True, then B also evaluates to True. (If A evaluates to False, the expression is considered True, since it cannot be said to be False.) Here a problem arises, since the generated logical formulas are not necessarily fully correctly representing the intended meaning, or, as in the case of music, the formula is roughly correct, say, in human use of language, but not fully in its strict (mathematical) interpretation.

Summary of the invention

It is an object of the invention to provide a system and method of the kind set forth that is better capable of dealing with differences in ontologies, in particular in differences in vocabularies that define the concepts in the ontologies.

To meet an object of the invention, a method of determining a measure of similarity between a first concept **A** and a second concept **B** from a first and second ontology relating to a same human knowledge domain, where each ontology describes a plurality of concepts and relations between the concepts of the ontology and where each concept represents at least one human understandable term, includes:

expressing the similarity as a logical relation between the first and second concept;

converting the first concept **A** to an expression $\mathbf{E_A}$ with at least one sub-expressions E_{A_i} including respective literal concepts $\mathbf{A_i}$ ($i = 1 .. I; I \geq 1$), where the literal concepts $\mathbf{A_i}$ are related to the first concept **A** according to the first ontology; and converting the second concept **B** to an expression $\mathbf{E_B}$ with at least one sub-expressions E_{B_j} representing respective literal concepts $\mathbf{B_j}$ ($j = 1 .. J; J \geq 1$), where the literal concepts $\mathbf{B_j}$ are related to the second concept **B** according to the second ontology; at least the expression $\mathbf{E_A}$ or the expression $\mathbf{E_B}$ including a plurality of subexpressions;

converting the logical relation to a plurality of logical sub-relations between the sub-expressions E_{A_i} and E_{B_j};

determining the measure of similarity in dependence on how many of the logical sub-relations evaluate positively.

According to the invention, a concept is 'expanded' to an expression with a plurality of sub-expressions that include literal concepts. For example, in a hierarchical ontology a concept may be expanded to an expression that represents its children concepts. This can be repeated until the concepts are reached that are no longer to be expanded further (hereinafter referred to as literal concepts). A literal concept may be a concept that can not be expanded further (e.g. in a hierarchical ontology form the end nodes) or as treated as such literal concepts in the sense that a concept that could be expanded further is not expanded, for example for computation efficiency. For example, in a hierarchical ontology it may be decided not to expand a subtree of concepts of a certain concept (node in the tree). Thus, if a concept C can be represented as C=A AND D, where D=B OR E, then in testing an expression containing C one may treat D as a literal (e.g. if B and E will not appear in another way than through expansion (elimination) of D).

The two concepts are now compared generating a plurality of sub-relations between the two sets of literal concepts represented in the expressions. A measure of similarity can be generated based on the evaluation of all the sub-relations. The measure may, for example, be based on an absolute number of positive evaluations or on a relative number of positive evaluations (e.g. a percentage of positive evaluations). In this way, it is not required that the similarity between concepts must be a 'full match'. A high level of similarity can still be detected even if the ontologies are not based on a same vocabulary or if the ontologies have a different structure. The measure of similarity can be seen as a measure of relationship between the ontologies. In most systems, the ontologies will be different. As a special case, one of the ontologies may be a sub-part of the other (e.g. integrated by reference). In this case, the method may for example be used to find out if there is an overlap between a concept of the main part and a concept of the sub-part.

According to the measure of dependent claim 2, the method further includes assigning respective weights to the logical sub-relations and the step of determining the measure of similarity includes weighing a contribution of a positive evaluation of a logical sub-relation with the respective weight. In this way the measure can be made more accurate. Any suitable form of weights may be used. For example, based on a position of the involved concept in the ontology (e.g. concepts that are lower in a hierarchical ontology may be assigned a higher weight), number of occurrences of a concept in base (e.g. a concept 'rock' with a 1000 audio titles assigned to it may be assigned a higher weight than a concept 'southern-rock' with only six titles assigned to it), number of children concepts in hierarchically arranged ontology.

According to the measure of dependent claim 3, the method further includes:

receiving a human-understandable term representing a human understandable multi-word;

representing the human understandable term as a concept that is related to plurality of literal concepts that each represent a respective single word of the multi-word. In this way, a concept representing a multi-word can be matched more accurately. For example, the multi-word "glam-rock" of the first ontology may now give a positive evaluation result if the second ontology only has the concept "rock" or "glam". It will be appreciated that then the measure of similarity will not be 100%, but for example only 50% which may depend on the match of related concepts.

According to the measure of dependent claim 4, the step of converting the first concept **A** to an expression $\mathbf{E_A}$ includes using a predetermined form in a expression language, in particular a Disjunctive Normal Form. According to the measure of dependent claim 5, the step of converting the second concept **B** to an expression $\mathbf{E_B}$ includes using a further predetermined format in the same expression language, in particular a Conjunctive Normal

Form. $\mathbf{E}_A$ and $\mathbf{E}_B$ are expressed in the same expression language. Using these two different predetermined forms makes the evaluation straightforward.

According to the measure of dependent claim 6, the method further includes determining that the first and second ontology match if the similarity measure is above a predetermined threshold that is less than a positive evaluation of all logical sub-relations. In this way, the system can determine that there is a match even if both ontologies do no fully use the same concepts or relationships. A user may set the threshold, or the system may propose a threshold.

According to the measure of dependent claim 7, the method further includes reporting a representation of the measure of similarity to a user. The reporting may take any suitable form, such as in word or graphical showing a percentage of matching sub-relations. As an alternative to reporting the measure to the user, the measure may determine further actions to be taken by a system using the method. For example, a high enough similarity may mean that the involved concept of the second ontology is presented to a user, the involved concept may also be integrated into the first ontology forming a unified ontology, etc.

According to the measure of dependent claim 8, the human knowledge domain is audio and/or video content; one of the first and second ontologies representing a user preference profile, hereinafter referred to as "preference ontology"; the other of the first and second ontologies representing a listing of content offered by a content provider, hereinafter referred to as "provider ontology". This is an effective way of checking whether a content provider has titles to provide that meet the preferences of a user. The user may use his own vocabulary to build a preference ontology that needs not be exactly the same as used by the service provider. In this way also content can be selected from several providers, each with their own ontology based on a single preference ontology for example as described by the measure of dependent claim 9.

An object of the invention is also met by a system including a first device storing a first concept **A** from a first ontology and a second device storing a second concept **B** from a second ontology, where the first and second ontology correspond to a same human knowledge domain and each ontology describes a plurality of concepts and relations between the concepts of the ontology, where each concept represents at least one human understandable term; the system further including means for determining a measure of similarity between the first concept **A** and the second concept **B** by:

expressing the similarity as a logical relation between the first and second concept;

converting the first concept **A** to an expression $\mathbf{E_A}$ with a plurality of sub-expressions E_{A_i} each representing respective literal concepts $\mathbf{A_i}$ (i = 1 .. I; I >1), where the literal concepts $\mathbf{A_i}$ are related to the first concept **A** according to the first ontology;

converting the second concept **B** to an expression $\mathbf{E_B}$ with a plurality of sub-expressions E_{B_j} each representing respective literal concepts $\mathbf{B_j}$ (j = 1 .. J; J >1), where the literal concepts $\mathbf{B_j}$ are related to the second concept according to the second ontology;

converting the logical relation to a plurality of logical sub-relations between the sub-expressions E_{A_i} and E_{B_j}; and

determining the measure of similarity in dependence on how many of the logical sub-relations evaluate positively.

These and other aspects of the invention are apparent from and will be elucidated with reference to the embodiments described hereinafter.

Brief description of the drawings

In the drawings:

Fig. 1 shows a block diagram of an exemplary prior art system;

Fig. 2 shows a block diagram of a system according to the invention;

Fig. 3 shows two exemplary ontologies;

Fig. 4 shows main components of the system according to the invention;

Fig. 5 illustrates the method for unifying ontologies;

Fig. 6 shows an exemplary user device; and

Fig. 7 shows how the matching process can be used as a Web Service.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

According to the invention, some form of approximating relationships between concepts of ontologies is provided, of which the subclass relation is the predominant one. The approximation is particularly useful for situations wherein the ontologies represent the human knowledge domain of audio/video content. Music/video genres are not black-and-whitely bounded concepts, which calls for some form of tolerance in the decision. The method and system according to the invention express the tolerance in terms of how well the black-and-white relation is approximated. This can enable a user of the system to control the tolerance in a sensible manner. For example, a user may accept different tolerances on different aspects of the comparison. The system may also use such tolerance as guidance in automatic decisions (e.g. whether or not to merge two ontologies into one).

Fig. 2 shows a block diagram of a system in which the invention may be used. In this exemplary system two devices 200 and 210 are shown, each with a storage for storing a respective first and second ontology 202, 212. The ontologies may be stored in any suitable storage, preferably a non-volatile memory, such as a hard disk, optical storage (e.g. DVD+RW) or flash memory. The ontologies correspond to a same human knowledge domain, preferably audio and/or video content. Each ontology describes a plurality of concepts and relations between the concepts of the ontology. Each concept represents at least one human understandable term. Fig.3 illustrates two ontologies A and B. For ontology A, concepts are explicitly shown that represent the human terms "Rock", "Blues", "Glam Rock" and "Heavy Metal". More concepts and relationships may be present. In Fig.3, relationships are shown as lines between the concepts. In the example of Fig.3, the ontology is strictly hierarchical (starting from a root, parent concepts can have children concepts). This is not required but is used for simplicity of the explanation. Other ontologies are well-known.

In a preferred embodiment, one of the ontologies represents a user preference profile, hereinafter referred to as "preference ontology"; the other one of the ontologies represents a listing of content offered by a content provider, hereinafter referred to as "provider ontology". One or both of the ontologies may have been created by incorporating concepts of other ontologies, that using the method according to the invention have shown to be relevant.

In a preferred embodiment, if a concept of the provider ontology appears relevant, the system represents that concept in a listing of content titles offered to the user for rendering. As will be described in more detail below, determining whether a concept is relevant can be done by comparing the measure of similarity to a threshold.

The description gives details on a method of determining a measure of similarity between a first concept **A** from the first ontology and a second concept **B** from the second ontology. The concepts being compared may be any of the concepts on the ontology. For example, in a hierarchical ontology it may be the root concept, a leaf concept or any intermediate level concept. The comparison is done by a comparison unit. It is sufficient that the system has one comparison unit; data from the ontologies may be loaded through a network 220, such as Internet or a local area network. Fig.2 shows that each device has a respective comparison unit 208 and 218. The first device 200 also has a user I/O device 206 for interaction with a user. For example, the user may provide input for assisting in determining the preferences of the user, such as favorite types of music/video. The first device 100 also has a unit 204 for representing the user preferences as a preference ontology.

According to the invention, the measure of similarity is determined by:

1. expressing the similarity as a logical relation between the first and second concept;
2. converting the first concept **A** to a corresponding expression $\mathbf{E_A}$ with a plurality of sub-expressions E_{A_i} each representing respective literal concepts $\mathbf{A_i}$ (i = 1 .. I; I >1), where the literal concepts $\mathbf{A_i}$ are related to the first concept **A** according to the first ontology;
3. converting the second concept **B** to a corresponding expression $\mathbf{E_B}$ with a plurality of sub-expressions E_{B_j} each representing respective literal concepts $\mathbf{B_j}$ (j = 1 .. J; J >1), where the literal concepts $\mathbf{B_j}$ are related to the second concept according to the second ontology;
4. converting the logical relation to a plurality of logical sub-relations between the sub-expressions E_{A_i} and E_{B_j}; and
5. determining the measure of similarity in dependence on how many of the logical sub-relations evaluate positively.

Ad.1 For example, the logical relationship could be that concept A is a sub-class of concept B (or vice versa) represented in a corresponding logical form $E_A \Rightarrow E_B$. Other similarities between A and B (e.g. A and B are equivalent) may also be evaluated, giving rise to other logical relationship that need to be checked. The example given below will demonstrate the method for checking whether A is a sub-class of B (A $\subseteq$ B). The method according to the invention provides information on how well $E_A \Rightarrow E_B$. This is reflected by the measure of similarity. In the remainder also "sloppiness" measure will be used to express that no full match is required.

Ad.2 The first concept **A** is converted to an expression $\mathbf{E_A}$ with a plurality of sub-expressions, each representing respective literal concepts $\mathbf{A_i}$ (i = 1 .. I; I >1), where the literal concepts $\mathbf{A_i}$ are related to the first concept **A** according to the first ontology. In a preferred embodiment, this rewriting is done by using a predetermined form in an expression language, in particular a Disjunctive Normal Form (DNF), which is a disjunction of conjunctions: $E_A = E_{A_1} \vee E_{A_2} \vee \ldots$, E_{A_i} where each $E_{A_i} = E_{A_{i_1}} \wedge E_{A_{i_2}} \wedge \ldots$. If so desired, other forms than DNF may be used. Such forms are well-known from Mathematical Logic.

Ad.3 The second concept **B** is converted to an expression **E**

$$\forall j \; E_{A_1} \vee E_{A_2} \vee \ldots \Rightarrow E_{B_j}$$

where J equals the number of E_{B_j}. Taken together the problem (*) can be rewritten as a set of I x J subproblems that all need to be satisfied (to evaluate to True):

$$\forall i, j \; E_{A_i} \Rightarrow E_{B_j} \quad (**)$$

As explained above, each of the E_{A_i} and E_{B_j} are CNF and DNF themselves, the members of which are (treated as) literals. Because of this, the form of expression of the subproblems is called clauses. In other words, the problem (*) is rewritten in a set of satisfiability tests on a set of I x J clauses, problem (**).

According to the invention, instead of requiring that all I x J subproblems evaluate to True, it is allowed that some of them evaluate to False. The similarity need thus not be full. This measure of similarity is 'the sloppiness value'. In a preferred embodiment, the sloppiness value is used for determining whether two concepts of different ontologies match. This is done by deciding that there is a match if the similarity measure is above a predetermined threshold that is a fraction of a positive evaluation of all logical sub-relations. For example, $E_A \Rightarrow E_B$ is said to be True if less than 0.3 of the $E_{A_i} \Rightarrow E_{B_j}$ is False (the threshold is thus 0.7).

Note, that the use of sloppiness by itself may be used in implementing the evaluation of the main problem (*). For example, as soon as the sloppiness threshold has been reached, the remaining subproblems (**) do not require evaluation. Other dependencies can also be taken into account to optimize the computation. For example, if one of the pairs E_{A_i}, E_{B_j} is disjoint, or, less extreme, the corresponding subproblem (**) is False up to a considerably large sloppiness value, one may decide to not accept the main problem (*) at all, and hence can discard the other remaining subproblems. Yet another refinement is to assign weights to each of the subproblems. The weights express the relative contribution of the corresponding subproblem to accepting the main problem, or, similarly, their relative contribution to sloppiness consumption. These weights may follow from the size of the classes that are evaluated in the subproblem, for example. Further, they can be used in optimizing the total evaluation in providing an order of evaluation, together with the described mechanism of discarding further evaluation if a threshold has been reached.

Experiments have revealed that this sloppiness measure is very effective in finding the proper matches between the terms in the two bases (ontologies) A and B. For example, E_A is an expression for “Contemporary Bluegrass” and E_B is an expression for “Contemporary Country”. Both are known to be subclasses of “Country”, which fact is also expressed in the formulas. The question to the system is to decide whether “Contemporary Bluegrass” is a subclass of “Contemporary Country”. In the experiment, a strict evaluation of $E_A \Rightarrow E_B$ concludes False. When sloppiness is set to 0.3, the more useful True is obtained (threshold is then 0.7).

Instead of setting the sloppiness threshold to find matches, one can also let the system find that sloppiness value at which terms starts to become related. Experiments revealed that at a certain threshold the number of matches steeply increases (while not including them all). This threshold could serve as the intended measure (threshold used for the matching).

Some of the disjuncts E_{A_i} and some of the conjuncts E_{B_j} can be more discriminating than the others in deciding the truth value of $E_{A_i} \Rightarrow E_{B_j}$. This can be reflected by assigning different weights to each of the conjuncts/disjuncts in the way they add to the cumulative sloppiness. For example, the weight can be inversionally proportional to the number of instances that is known to belong to the conjunct/disjunct. Similarly, the contribution of each literal can be weighted for a more accurate sloppiness value. For example, Cajun or Zydeco should not be equally weighted as Blues or Jazz.

Detailed embodiments of the invention method

By way of example, Fig.3 displays part of the schemas of two knowledge or data bases A and B, along which music is structured and offered to users. The base A contains, among others, the genres “Rock” and “Blues”, of which “Rock” distinguishes, among others, “Glam Rock” and “Heavy Metal”. Likewise, the base B contains the genres “Rock” and “Blues”, and the subgenres “Glam Rock” and “Heavy Metal”, which are part of the genres “Glam” and “Hard”. “Hard” also contains a subgenre “Southern Rock”. Note that, although names in A and B are identical, that doesn’t necessarily mean both bases contain, or intend to contain, the same music songs. Since the base B has another structure than base A, “Heavy Metal” in A might fit with “Hard” or another subgenre in base B, for example.

In this example the bases A and B are considered to represent the schemas of two different providers offering music. It is clear that, instead of music, the method also applies to other types of content, such as video, text, hypertext, photos and other pictures, multimedia, hypermedia, and mixtures thereof. The content may also be medical data, e.g. of different departments in a hospital, or of different medical parties, such as a hospital, a general practitioner, a chemist, etc. It is also noted that instead of representing the schemas of two providers, the two bases can also represent two (temporary) parts of a same base. In that case the method is applied to enrich or clean-up the base, for example. Enrichment results from the discovery of additional relationships, clean-up results from merging classes that actually are representing the same concept. Yet another view on the two bases A and B is that one, or both, represent a user's profile, such as the user's preferences. Say, base A represent a user's preference profile, e.g., the music genres the user likes. The task, then, is to find the matching classes in base B that respond to the user's preferences. In case the two bases both represent a user profile, from a same or two different users, the task is to enrich or clean-up the user's profile as above, or to find the merged profile of both users, so that a query can be created that optimizes the preferences of both users. Finally, instead of considering two bases the method can be involved with multiple bases. For example, the purpose is to find a unified view over multiple providers, where the unification is limited to those parts of the provider bases that match within the scope of the profiles from some set of users. It is clear there are many other contexts than the one described here in detail to apply the method.

Returning back to the example of Fig.3, the task is to decide which of the listed genres are related to each other. In this example, the problem of "termOfA equivalentTo termOfB" is considered only. Other problems are treated similarly. The problem is True, if both the two problems "termOfA subClassOf termOfB" and "termOfB subClassOf termOfA" are found to be True. The discussion is limited to comparing the two Glam_Rock classes.

First the terms, Rock, Blues, Heavy Metal, etc. are expressed as logical formulas. For the sake of this example, those expressions are kept simple, but in realistic implementations they might become lengthy and complex. In the formulas below, the terms themselves are expressed as a logic formula, bound it to their parent and set it disjoint from their sibling. Each word in a term is transformed in a logical proposition (literal). A term that is a multiword is transformed into the union of that multiword with the intersection of all its constituents. Words that are meaningless, such as "Heavy", are omitted.

Rock_A ::= Rock

Glam_Rock_A ::= ((Glam_Rock) ∨ (Glam ∧ Rock)) ∧ (Rock) ∧ ¬(Metal)

$\text{Heavy\_Metal\_A} ::= ((\text{Heavy\_Metal}) \vee (\text{Metal})) \wedge (\text{Rock})$
$\wedge \neg((\text{Glam\_Rock}) \vee (\text{Glam} \wedge \text{Rock}))$
$\text{Rock\_B} ::= \text{Rock}$
$\text{Hard\_B} ::= (\text{Hard}) \wedge (\text{Rock}) \wedge \neg(\text{Glam})$
$\text{Heavy\_Metal\_B} ::= ((\text{Heavy\_Metal}) \vee (\text{Metal})) \wedge (\text{Hard}) \wedge (\text{Rock})$
$\wedge \neg(\text{Southern\_Rock})$
$\text{Southern\_Rock\_B} ::= (\text{Southern\_Rock}) \wedge (\text{Hard}) \wedge (\text{Rock})$
$\wedge \neg((\text{Heavy\_Metal}) \vee (\text{Metal}))$
$\text{Glam\_B} ::= (\text{Glam}) \wedge (\text{Rock} \wedge \text{Music}) \wedge \neg(\text{Hard})$
$\text{Glam\_Rock\_B} ::= ((\text{Glam\_Rock}) \vee (\text{Glam} \wedge \text{Rock})) \wedge (\text{Glam}) \wedge (\text{Rock})$

As said, the evaluation is limited to the two Glam_Rock classes. Further, for the sake of simplicity, from the formulas the disjoint sibling nodes are omitted: the logical formula for "Glam Rock" in A is simplified by ignoring the intersection with the complement of "Heavy Metal".

The A and B formulas are rewritten in DNF and CNF and simplified by using equivalence relations, such as absorption. Rewrite techniques are well-known from the literature. This gives:

$\text{Glam\_Rock\_A\_DNF} ::= (\text{Glam\_Rock} \wedge \text{Rock}) \vee (\text{Glam} \wedge \text{Rock})$
$\text{Glam\_Rock\_B\_DNF} ::= (\text{Glam} \wedge \text{Rock})$
$\text{Glam\_Rock\_A\_CNF} ::= (\text{Glam\_Rock} \vee \text{Glam}) \wedge (\text{Rock})$
$\text{Glam\_Rock\_B\_CNF} ::= (\text{Glam}) \wedge (\text{Rock})$

This gives the desired logical formulas. Recall, that the names at the right-hand sides represent logical expressions, i.e. the propositions "Rock" and "Glam" can take the logical values True and False. In the second step, the terms derived from base A are compared with those derived from base B for subClassOf relationships, and vice versa. In other words, it must be decided whether the formulas term_A ⇒ term_B and term_B ⇒ term_A are satisfiable (take the value True), where term_A and term_B are to be substituted by one of the expressions above.

In the case of the subsumption term_B ⇒ term_A it is preferred to have the formulas of B in DNF form, and A in CNF:

$\text{term\_B} ::= (\text{Glam} \wedge \text{Rock})$

term_A ::= (Glam_Rock $\vee$ Glam) $\wedge$ (Rock)

term_B consists of only one disjunct, and term_A consists of two conjuncts. So, it is then required to check for two pairs, namely:

(Glam $\wedge$ Rock) $\Rightarrow$ (Rock)

(Glam $\wedge$ Rock) $\Rightarrow$ (Glam $\vee$ Glam_Rock)

The first subformula is always True, since Rock is on both sides. Similarly, the second term is always True, since Glam is on both sides. So, both the subformulas are satisfied and hence the total relation, and Glam_Rock_B $\subseteq$ Glam_Rock_A holds with sloppiness 0%.

In the case of the subsumption term_A $\Rightarrow$ term_B, it is preferred to have the formulas of A in DNF form, and B in CNF:

term_A ::= (Glam_Rock $\wedge$ Rock) $\vee$ (Glam $\wedge$ Rock)

term_B ::= (Glam) $\wedge$ (Rock)

term_A consists of two disjuncts, and term_B consists of two conjuncts. So, it is required to check for four pairs, namely:

(Glam $\wedge$ Rock) $\Rightarrow$ (Rock)

(Glam $\wedge$ Rock) $\Rightarrow$ (Glam)

(Glam_Rock $\wedge$ Rock) $\Rightarrow$ (Rock)

(Glam_Rock $\wedge$ Rock) $\Rightarrow$ (Glam)

The first three subformulas are always True, since either Rock or Glam is on both sides. The fourth subformula is not always satisfied, and hence it can not be concluded that the total relation holds. However, if a sloppiness of 25% is allowed, the total relation can be accepted, i.e. Glam_Rock_A $\subseteq$ Glam_Rock_B holds with sloppiness 25%.

When assessing the sloppiness in the equivalence relation between Glam_Rock_A and Glam_Rock_B, the maximum of the sloppiness calculated in the two subsumptions is taken, i.e. Glam_Rock_B $\equiv$ Glam_Rock_A is inferred with sloppiness 25%.

It should be noted that the given example is simplified for the ease of explanation. In realistic situations the formulas can be more complicated and the number of subformulas can be larger than that in the example.

System

Fig.4 depicts schematically exemplary system components. Content is stored at and offered by several sources. Fig.4 shows 2 sources 402 and 404, but this can be a virtually infinite number. Also, as said before, it can also be a single source, or represent a user profile instead of provider's schema. The content can concern audio data (music), but it can also be other forms of content, such as video, pictures, photos, graphics, text, multimedia, lighting and games. For example, the lighting may relate to ambient lighting that is controlled to reflect a mood of a user or to match a 'mood' of media content being rendered, where the matching is according to a user preference. The content can also concern data from another domain, such as the health care domain, where the possible types of data include electronic patient records (EPRs), medical terminology, knowledge about the human anatomy, reasons of admission to the intensive care unit (ICU) etc. The content is organized according to some schema. The schema is known or accessible by the matching process 410. In practical situation the schema might be indirectly visible. For example, the content is listed in a set of web pages (HTML pages), and the navigation path sequences through them, together with the listed names on each of the pages, define the schema. In such cases, it is assumed the schema is created off-line and offered to the matching process. The matching process performs the integration as described in the previous section. On the one hand, this may concern the relationship discovery between the genres offered by the sources A and B, such that a unified view can be generated that structures the content from all sources in a single, compact schema. On the other hand, this may concern matching between user preferences, or queries, that indicate which part of the offered content the user is interested in, and wishes to see on his display. User preferences and queries can be treated in a similar way as the terms from any of the sources. For example, after rewriting in a logical formula, the preferences are tested for equality or subClassOf relationships with the genres in the sources (their logical rewrites), and matching pairs are selected for presentation to the user, possibly in a unified way over the different sources.

At the other side of the matching process is the user interface 420. This concerns the input and output devices, or device in case these functions are combined. A typical output device is a display, which could be part of a handheld device. Typical input devices are a keyboard and a remote control. Obviously, in case the content is music, a sound reproduction device is also present (receiver, player, decoder, amplifier, loudspeaker, etc.).

Fig.4 only depicts the main system components that are relevant for the current invention. For example, next to the matching process other functions can be executed between user interface and sources. For example, as depicted in Fig.1, an ontology might reside with the matching process, which is used to refine the matching process. An example is to use the ontology as background knowledge to create the logical formulas: a term is expanded in a logical formula based on the concept definitions provided by the ontology, similar to the manner that WordNet is used in [trento]. Other examples of functions include the discovery of the sources and the communication processes with them.

As another example, in creating a unified view over providers, the main purpose of the matching process is to prevent that a user who is looking for content of interest does not need to go to each provider separately and has to navigate that provider's interaction schema, causing the user to repeatedly going up and down navigation trees, switching between sites (with another navigation model), and by that getting lost in keeping track where the content of interest resides. So, the purpose is to find, filter out, and integrate matching classes from different sources, and, for that purpose, it is not necessarily required to create a unified schema over all sources. The method could be part of such a process, however. When unifying two (or more) schemas, one needs to (re-)classify the classes from the two (or more) schemas relative to each other's schema. For each pair of classes it is needed to decide whether they are equivalent, a subclass of each other or unrelated, so that a new taxonomy tree builds up. The described method obviously allows to discover such relàtionships. After completion of the discovery process, other steps might be needed to arrive at a complete unification. For example, it might happen that some inconsistencies have arisen. Then, for a complete unification, these inconsistencies need to be solved.

An example of an inconsistency is the following as illustrated in Fig.5. Let's assume base A contains two classes, C_A and D_A, of which D_A is declared as subclass of C_A, and base B contains two classes, D_B and C_B, of which C_B is declared as subclass of D_B. Now, let's assume the matching process finds that C_B is equivalent to C_A, and that D_B is equivalent to D_A. Having the four classes, C_A, C_B, D_A, D_B, together with the original and found relationships in the unified schema yields an inconsistency. That is, the total set of classes and relationships would imply that the four classes are equivalent. This is opposing the information from both the original bases that they have a subclass relationship. One way to solve this inconsistency is to use the sloppiness measure. For example, assume that the matching process had found the equivalence between C_B and C_A at a sloppiness 0, while between D_B and D_A it was at 0.5. In that case, the unification process could decide to discard the latter equivalence, concluding to a unified schema where D_A is a subclass of the equivalent classes C_A and C_B, which in turn are a subclass of D_B. As another example, it might be that within one equivalence relation different sloppiness figures are found for each of its constituting subclass relations. This can also be used to arrive at a unified schema. For example, say C_B was found as subclass of C_A with a sloppiness 0, while C_A was found as subclass of C_B with a sloppiness 0.4. Likewise, D_A was found as subclass of D_B with a sloppiness 0, while D_B was found as subclass of D_A with a sloppiness 0.4. In that case, it is reasonable to discard the subclass relations with the larger sloppiness and to conclude that the unified schema is that where D_A is a subclass of C_B, which is a subclass of D_B, which is a subclass of C_A. Next to considering the mentioned relationships, the other relationships with their sloppiness figure can also be included: the relationship between C_A and D_B, and the relationship between C_B and D_A. Moreover, the given (subclass) relationships between C_A and D_A, and between D_B and C_B, can be evaluated for their sloppiness value and be used in the unification process. This can be illustrated by the following example, again concerning the case depicted in Fig.5. Assume that the matching process has found subclass relations only, no equivalence: C_A as subclass of C_B, and D_B as subclass of D_A. Again, this would lead to the conclusion that the four classes are equivalent, since they are subclasses of each other in a circular way. If all four subclass relations have the same sloppiness figure, it is reasonable to conclude this equivalence (at that sloppiness). However, if one of the subclass relations requires a significant larger sloppiness figure, it is more reasonable to discard that subclass relation, and to defeat the inferred equivalence amongst the four classes.

Of course, the sloppiness values can be contradicting themselves, needing more sophisticated rules then the ones described, or even not providing sufficient bias for resolving the inconsistency. For example, if all sloppiness values are equal they can only serve as an indication that the difference between the encountered inconsistencies is balanced, and hence arbitrary. Other approaches to solve for inconsistencies include the use of background knowledge, or to assign one or some of the bases as master (being decisive) over the others. This assignment can be based again on some trust value derived from observed sloppiness patterns and statistics.

An important aspect of the invention concerns the use of a sloppiness measure, which is a measure of similarity. This is shown in Fig.4 as an additional input 430 to the matching process. The sloppiness measure can be set, including being hard coded, in the matching process. It can be controlled by a system operator, it can be made dependent on observed interactions with the user, it can be user dependent, it can be policy dependent, e.g. related to service contract of operator to user, etc. Finally, it can be set by the user. The other way around, not shown in Figure 4, is that the sloppiness value at which a relationship holds is communicated back to the user or process using the system. For example, as described above, the sloppiness threshold can be used to resolve inconsistencies when unifying schemas. Its combination is also feasible: the user sets a sloppiness value, e.g., as initial or limit value, while the used or found threshold is communicated back. It will be clear that many other possibilities exist to use, control, and inform about the sloppiness setting.

It is not required that a single sloppiness value is used. Sloppiness can be set per source, even made dependent on the level depth of the concepts in their hierarchy, can be adapted based on the size of the response (number of found matches), the quality of the response, or the profile of the found items. There are many ways to extend the usage.

It is important to note that the diagram in Fig.4 abstracts from the actual realization. One form is the situation where the User Interface is implemented in a user device, such as a handheld device or PC, the matching process resides on some server in the network, such as the Internet, and the sources are other servers on that same network, such as Internet radio stations, Internet music providers, and Internet portals. Another form is an in-home network, where, for example, the matching process resides on a home-server or residential gateway, and the sources appear as hard discs on devices like PCs, DVD-players, and in-car players. The matching process is typically implemented using a suitable program executed in a processor after being loaded from a non-volatile storage.

Fig.6 shows an example of parts of the user device 600. In this example, the user enters request for music through the keyboard 620, the results of which are shown on the display 610. A next action is to select music items from such a result for reproduction. The result could also be a playlist that will initiate reproduction automatically. The display also informs the user about the used sloppiness figure, 15% in Fig.6 indicated with item 612. The number could be a derived number. For example, presented as a "reliability" measure, or "trust" value. Instead of a number, the sloppiness figure could be represented graphically, like an extending bar, or it could be a colored field, etc. Likewise the user could enter a sloppiness figure, through a dedicated dialog or immediately by placing focus at the display field. This figure is forwarded to the matching process, Fig.3, which will subsequently use it. Again, the number used by the matching process might be a number derived from the figure entered by the user. All sorts of interaction styles are conceivable.

A possible form of implementing the matching process resides in the use of Web Services [W3CwebServices] W3C, Web Services, http://www.w3.org/2002/ws/. The matching process is implemented stand-alone and offered as a service to be used by other processes. Fig.7 illustrates this. The Web Services framework consists of many functional parts, some of which are still developing or even require development. For example, the framework offers a way to register the service, e.g. through UDDI, so that users (people and/or machines) can find the service, a way to describe the usage of the service, e.g. through

WSDL, and a way to concatenate services, e.g. through WSFL. A description of UDDI can be found at OASIS, Universal Description, Discovery and Integration of Web Services, “http://www.uddi.org”. A description of WSDL can be found at:
[WSDL] W3C, Web Services Description Language (WSDL) 1.1, W3C Note, “http://www.w3.org/TR/wsdl.html”.
Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, “http://www.w3.org/TR/wsdl20”
Web Services Description Language (WSDL) Version 2.0 Part 2: Message Patterns, “http://www.w3.org/TR/wsdl20-patterns”
Web Services Description Language (WSDL) Version 1.2 Part 3: Bindings, “http://www.w3.org/TR/wsdl12-bindings”.

The service in which the matching process 712 resides, the “Integration Service” 710- in Fig.7, is a service that integrates knowledge bases for “you”. “you” can be a human being, who is accessing the service through an interface as shown in Fig.4 and 6, but it can equally well be another machine, that is running another service that can benefit from using the Integration Service. For example, the Integration Service is part of a chain of services, as described in a WSFL document. A description of WSFL can be found at:
[WSFL] F. Leymann, Web Services Flow Language (WSFL) 1.0, “http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf”.

The actual communication is described in a WSDL or alike document, but a conceivable scenario is the following. The user (human or machine) sends a list of knowledge bases (including databases or other forms of content bases) to the Integration Service. The form of the list is as specified in the WSDL document. The names and/or addresses of the bases are known to the user, or have been acquired from another service, such as a UDDI registry. Upon receiving the request, the Integration Service runs its matching routine over the bases 720 as specified in the list. (For the sake of simplicity, we omit additional communications like security/trust measures and payments.) The matching can be used to create unified view 730 over the bases, or over a particular selection from the bases. The view can be according to a schema also provided by the user. This is not shown in the figure. In addition to the list of bases to be unified, the user can submit a sloppiness figure 740 that the Integration Service has to use in creating the view. As before, the figure can be a set of figures, each element of which relates to some aspect of the matching process, such as depth of term in source tree, or number of subformulas, etc. The format in which the sloppiness figure is exchanged is described in the WSDL (or alike) document. The used sloppiness figure can be communicated back in the returned view. Also, as described above, the matching process may generate a sloppiness grade itself and return that with its response, possibly distinguishing parts in the view by labeling them with their sloppiness measure. For example, the first part in the view contains matches with sloppiness 0%, the next part those with 10%, etc. Of course, the service, and its interface description, may call the figure differently from sloppiness, using names like probability or acceptance threshold, or, more important, may use a figure that is derived from the sloppiness as described here.

It will be appreciated that the invention also extends to computer programs, particularly computer programs on or in a carrier, adapted for putting the invention into practice. The program may be in the form of source code, object code, a code intermediate source and object code such as partially compiled form, or in any other form suitable for use in the implementation of the method according to the invention. The carrier may be any entity or device capable of carrying the program. For example, the carrier may include a storage medium, such as a ROM, for example a CD ROM or a semiconductor ROM, or a magnetic recording medium, for example a floppy disc or hard disk. Further the carrier may be a transmissible carrier such as an electrical or optical signal, which may be conveyed via electrical or optical cable or by radio or other means. When the program is embodied in such a signal, the carrier may be constituted by such cable or other device or means. Alternatively, the carrier may be an integrated circuit in which the program is embedded, the integrated circuit being adapted for performing, or for use in the performance of, the relevant method.

It should be noted that the above-mentioned embodiments illustrate rather than limit the invention, and that those skilled in the art will be able to design many alternative embodiments without departing from the scope of the appended claims. In the claims, any reference signs placed between parentheses shall not be construed as limiting the claim. Use of the verb "comprise" and its conjugations does not exclude the presence of elements or steps other than those stated in a claim. The article "a" or "an" preceding an element does not exclude the presence of a plurality of such elements. The invention may be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer. In the device claim enumerating several means, several of these means may be embodied by one and the same item of hardware. The mere fact that certain measures are recited in mutually different dependent claims does not indicate that a combination of these measures cannot be used to advantage.

CLAIMS:

1. A method of determining a measure of similarity between a first concept **A** and a second concept **B** from a first and second ontology corresponding to a same human knowledge domain, each ontology describing a plurality of concepts and relations between the concepts of the ontology, where each concept represents at least one human understandable term; the method including:

expressing the similarity as a logical relation between the first and second concept;

converting the first concept **A** to an expression $\mathbf{E_A}$ with at least one sub-expressions E_{A_i} including respective literal concepts $\mathbf{A_i}$ (i = 1 .. I; I ≥ 1), where the literal concepts $\mathbf{A_i}$ are related to the first concept **A** according to the first ontology; and converting the second concept **B** to an expression $\mathbf{E_B}$ with at least one sub-expressions E_{B_j} representing respective literal concepts $\mathbf{B_j}$ (j = 1 .. J; J ≥ 1), where the literal concepts $\mathbf{B_j}$ are related to the second concept **B** according to the second ontology; at least the expression $\mathbf{E_A}$ or the expression $\mathbf{E_B}$ including a plurality of subexpressions;

converting the logical relation to a plurality of logical sub-relations between the sub-expressions E_{A_i} and E_{B_j};

determining the measure of similarity in dependence on how many of the logical sub-relations evaluate positively.

2. A method as claimed in claim 1, including assigning respective weights to the logical sub-relations and the step of determining the measure of similarity includes weighing a contribution of a positive evaluation of a logical sub-relation with the respective weight.

3. A method as claimed in claim 1, further including:

receiving a human-understandable term representing a human understandable multi-word;

representing the human understandable term as a concept that is related to plurality of literal concepts that each represent a respective single word of the multi-word.

4. A method as claimed in claim 1, wherein the step of converting the first concept **A** to an expression $\mathbf{E_A}$ includes using a predetermined form in an expression language, in particular a Disjunctive Normal Form.

5. A method as claimed in claim 4, wherein the step of converting the second concept **B** to an expression $\mathbf{E_B}$ includes using a further predetermined form in the expression language, in particular a Conjunctive Normal Form.

6. A method as claimed in claim 1, further including determining that two concepts from respectively the first and second ontology match if the similarity measure is above a predetermined threshold that is a fraction of a positive evaluation of all logical sub-relations.

7. A method as claimed in claim 1, further including reporting a representation of the measure of similarity to a user.

8. A method as claimed in claim 1, wherein the human knowledge domain is audio and/or video content; one of the first and second ontologies representing a user preference profile, hereinafter referred to as "preference ontology"; the other of the first and second ontologies representing a listing of content offered by a content provider, hereinafter referred to as "provider ontology".

9. A method as claimed in claims 6 and 8, further including representing concepts of the provider ontology that match the preference ontology into a listing of content titles offered to the user for rendering.

10. A computer program product for causing a processor to execute the method of claim 1.

11. A system including a first device (200) storing a first concept **A** from a first ontology (202) and a second device (210) storing a second concept **B** from a second ontology (212), where the first and second ontology correspond to a same human knowledge domain and each ontology describes a plurality of concepts and relations between the concepts of the ontology, where each concept represents at least one human understandable term; the system

further including means for determining a measure of similarity between the first concept **A** and the second concept **B** by:

expressing the similarity as a logical relation between the first and second concept;

converting the first concept **A** to an expression $\mathbf{E_A}$ with a plurality of sub-expressions E_{A_i} each representing respective literal concepts $\mathbf{A_i}$ ($i = 1 .. I; I > 1$), where the literal concepts $\mathbf{A_i}$ are related to the first concept **A** according to the first ontology;

converting the second concept **B** to an expression $\mathbf{E_B}$ with a plurality of sub-expressions E_{B_j} each representing respective literal concepts $\mathbf{B_j}$ ($j = 1 .. J; J > 1$), where the literal concepts $\mathbf{B_j}$ are related to the second concept according to the second ontology;

converting the logical relation to a plurality of logical sub-relations between the sub-expressions E_{A_i} and E_{B_j} ; and

determining the measure of similarity in dependence on how many of the logical sub-relations evaluate positively.

12. A system as claimed in claim 11, wherein the human knowledge domain is audio and/or video content; one of the first and second ontologies representing a user preference profile, hereinafter referred to as "preference ontology"; the other of the first and second ontologies representing a listing of content offered by a content provider, hereinafter referred to as "provider ontology".

13. A system as claimed in claim 12, further including means for determining that two concepts from respectively the first and second ontology match if the similarity measure is above a predetermined threshold that is a fraction of a positive evaluation of all logical sub-relations.

14. A system as claimed in claim 13, further including means for representing concepts of the provider ontology that match the preference ontology into a listing of content titles offered to the user for rendering.

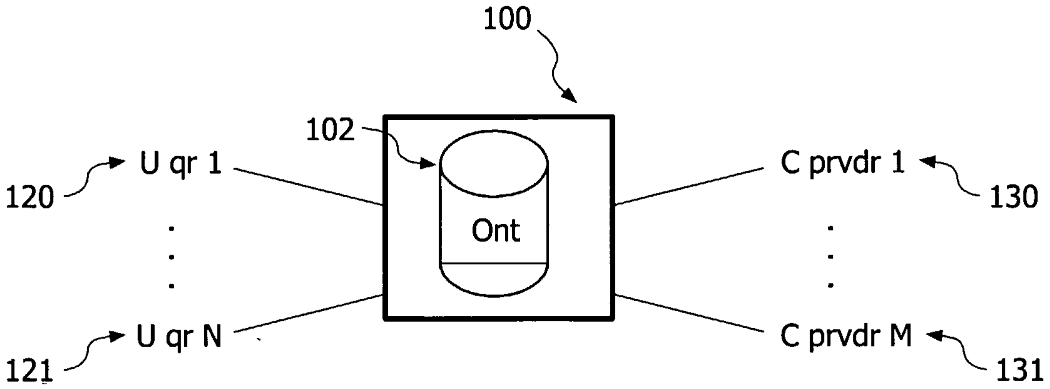

100
102
Ont
U qr 1
U qr N
120
121
C prvdr 1
C prvdr M
130
131

FIG. 1

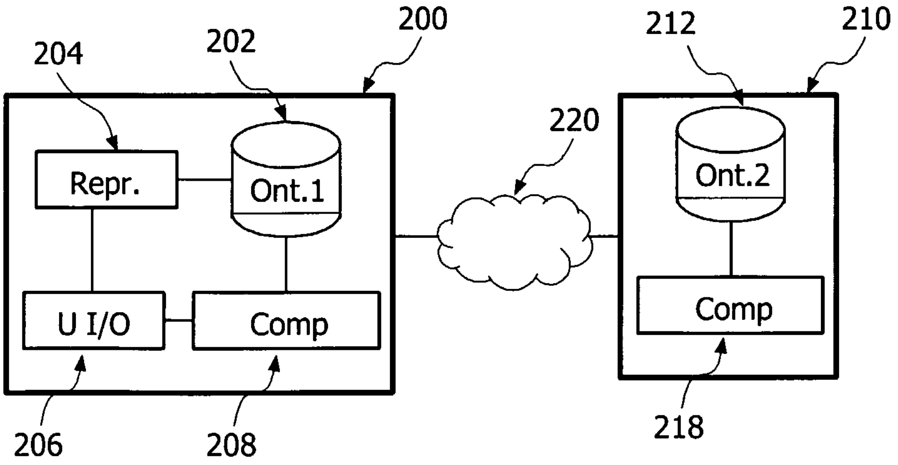

200
202
204
Repr.
Ont.1
U I/O
Comp
206
208
220
210
212
Ont.2
Comp
218

FIG. 2

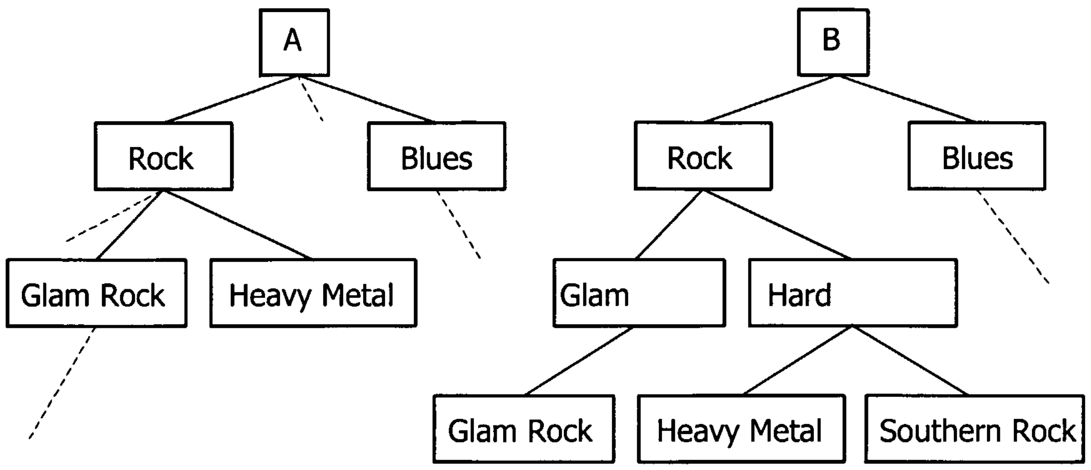

A
Rock
Blues
Glam Rock
Heavy Metal
B
Rock
Blues
Glam
Hard
Glam Rock
Heavy Metal
Southern Rock

FIG. 3

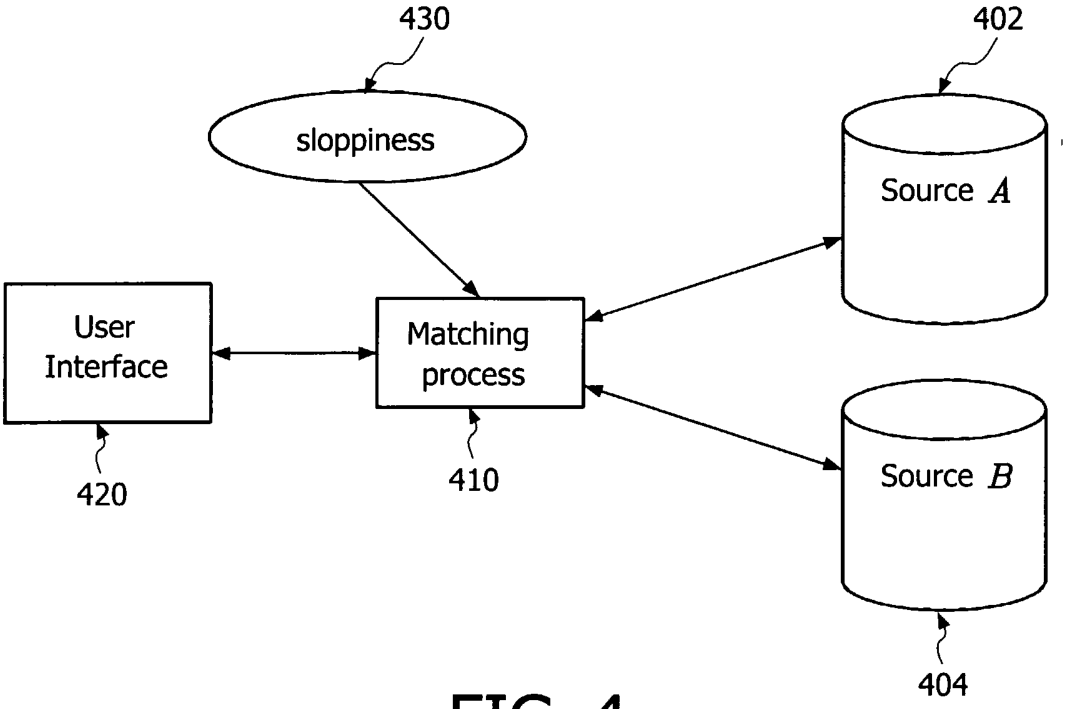

430
sloppiness
402
Source A
User Interface
Matching process
Source B
420
410
404

FIG. 4

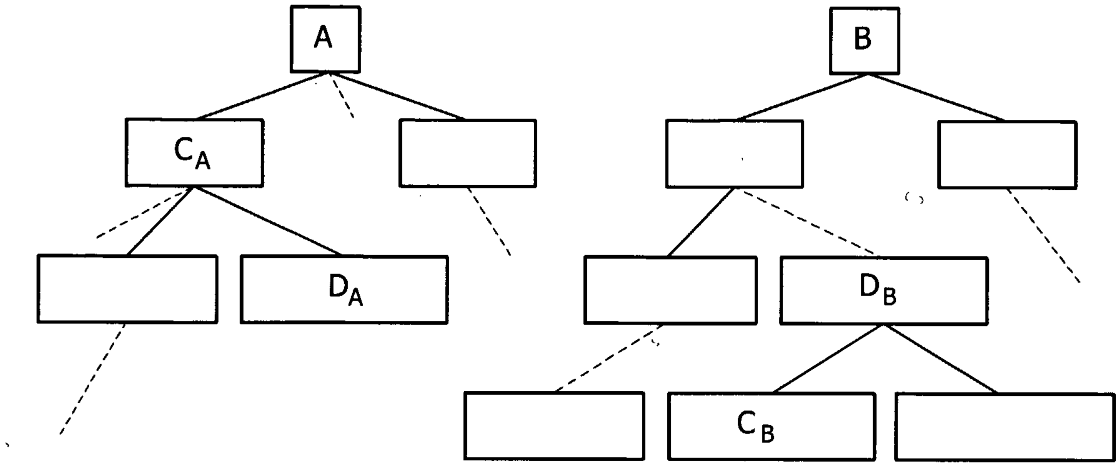


FIG. 5

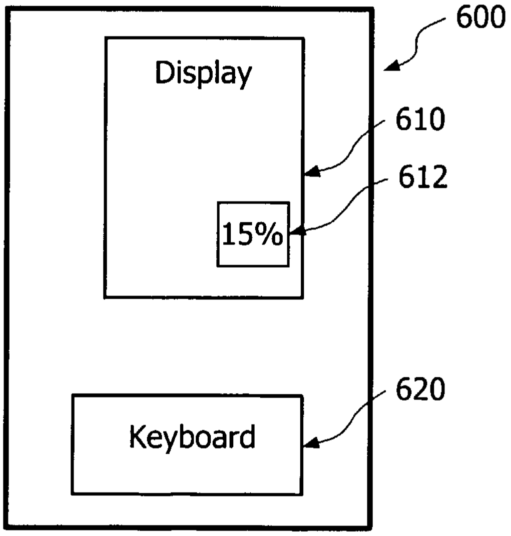


FIG. 6

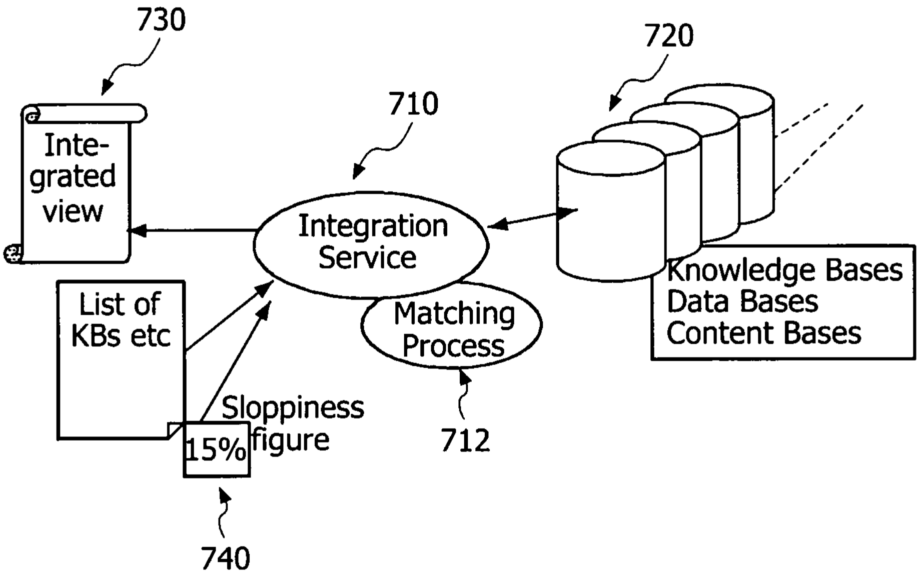

730
720
710
Inte-
grated
view
Integration
Service
Knowledge Bases
Data Bases
Content Bases
List of
KBs etc
Matching
Process
Sloppiness
figure
15%
712
740

FIG. 7

INTERNATIONAL SEARCH REPORT

IB2005/052360

A. CLASSIFICATION OF SUBJECT MATTER
G06F17/30

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)
G06F

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, BIOSIS, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Y. ZHONG ET AL.: "A general method for tree-comparison based on subtree similarity and its use in a taxonomic database" BIOSYSTEMS, vol. 42, no. 1, 1997, pages 1-8, XP002352428 Ireland the whole document -----	1-14
X	US 2003/120651 A1 (BERNSTEIN PHILIP A ET AL) 26 June 2003 (2003-06-26) paragraph ‘0058! - paragraph ‘0060! paragraph ‘0129! - paragraph ‘0160! ----- -/--	1-14

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in 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 document 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 3 November 2005	Date of mailing of the international search report 22/11/2005
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Eichenauer, L

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	NOY N F: "Tools for Mapping and Merging Ontologies" January 2004 (2004-01), HANDBOOK ON ONTOLOGIES, SPRINGER, HEIDELBERG, DE, PAGE(S) 365-384 , XP002341366 the whole document -----	

INTERNATIONAL SEARCH REPORT

IB2005/052360

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2003120651 A1	26-06-2003	US 2005027681 A1	03-02-2005
		US 2005060332 A1	17-03-2005