



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

(21) International Application Number:
PCT/US2011/021108

(22) International Filing Date:
13 January 2011 (13.01.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/294,663 13 January 2010 (13.01.2010) US

(71) Applicant (for all designated States except US): **AB INITIO TECHNOLOGY LLC** [US/US]; 201 Spring Street, Lexington, Massachusetts 02421 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **SCHON, Andrew** [US/US]; 291 Otis Street, Newton, Massachusetts 02465 (US).

(74) Agents: **HENNESSEY, Gilbert H.** et al.; Fish & Richardson P.C., P.O. Box 1022, Minneapolis, Minnesota 55440-1022 (US).

(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, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

- with international search report (Art. 21(3))
- with amended claims (Art. 19(1))

(54) Title: MATCHING METADATA SOURCES USING RULES FOR CHARACTERIZING MATCHES

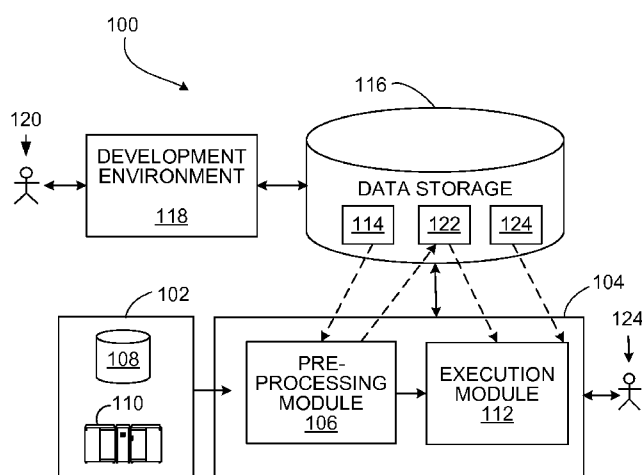


FIG. 1

(57) Abstract: Processing metadata includes storing (202), in a data storage system (116), a specification for each of multiple sources (102), each specification including information identifying one or more data elements of the corresponding source; and processing, in a data processing system (104) coupled to the data storage system, data elements from the sources, including generating (204) a set of rules for each source based on a corresponding one of the stored specifications, and matching (206) data elements of different sources and determining (208) a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source.

MATCHING METADATA SOURCES USING RULES FOR CHARACTERIZING MATCHES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Application Serial No. 61/294,663, filed
5 on January 13, 2010, incorporated herein by reference.

BACKGROUND

This description relates to matching metadata sources using rules for
characterizing matches.

Metadata discovery (also known as metadata scanning) can be used to discover
10 relationships between data elements representing metadata that describes values
appearing within datasets, such as the names of fields or columns of database tables or
spreadsheets. In some cases, the metadata for data appearing within a given dataset is
stored in a variety of different sources. During the metadata discovery process, a match
may be found between a data element in a first source and a data element in a second
15 source. A match can correspond to similar field names and/or descriptions of metadata
for fields in a table, for example. The match may indicate that the matching data
elements represent metadata for the same types of data values in respective datasets. In
some cases, a database of synonyms including user-specified, or dictionary-based
databases, e.g. WordNet, can be used to determine matches between data elements that
20 have similar semantic meanings (e.g., a match between “day” and “date,” or between
“gender” and “sex”). A master collection of metadata (sometimes called a “metadata
registry”) can be generated or updated to store metadata based on the discovered
relationships, or to link to metadata that has been found in the metadata discovery
process.

SUMMARY

25 In one aspect, in general, a method includes: storing, in a data storage system, a
specification for each of multiple sources, each specification including information
identifying one or more data elements of the corresponding source; and processing, in a
data processing system coupled to the data storage system, data elements from the

sources, including generating a set of rules for each source based on a corresponding one of the stored specifications, and matching data elements of different sources and determining a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules
5 generated for the first source and the set of rules generated for the second source.

Aspects can include one or more of the following.

The set of rules for each source can produce one or more grades corresponding to the quality metric characterizing the given match. The method can include providing explanatory information corresponding to the one or more grades. The given match can
10 include a match between names corresponding to the first and second data elements respectively and the one or more grades are based on the quality metric characterizing the given match. The given match can include a match between descriptions corresponding to the first and second data elements respectively and the one or more grades are based on the quality metric characterizing the given match.

15 The method can include classifying terms appearing in the first and second data elements into one or more classes; assigning one or more class words for each of the terms in the first and second data elements; comparing the one or more class words corresponding to terms in the first and second data elements respectively to generate the quality metric for the given match; and assigning the one or more grades based on the
20 quality metric characterizing the given match. The quality metric characterizing the given match can include a distance measure metric. A first grade can be assigned to an output of a first rule of the set of rules and a second grade is assigned to a second, different output of a second rule of the set of rules, wherein the first grade can indicate a better quality metric characterizing the given match relative to the second grade.

25 The set of rules are based on a similarity of names appearing in the first and second data elements respectively. The set of rules can be based on a similarity of descriptions appearing in the first and second data elements respectively. The method can include providing a user an ability to provide input for generating one or more rules of the set of rules for quantifying a quality of a match between the first data element and
30 the second data element. Each of the rules in the set of rules include trigger inputs and outputs based on the trigger inputs. Each of the rules in the set of rules are read

sequentially until all trigger inputs of a given rule in the set of rules evaluate to true. The quality metric characterizing the given match is based on a measure of a number of times a word occurs in a term of the first or second data element and the number of times a word occurs in a set of terms from the first or second source.

5 The method can include computing the quality metric characterizing the given match by: providing a first weight to terms that occur with a first frequency in the first or second source; and providing a second weight to terms that occur with a second frequency in the first or second source, wherein a value of the first weight is less than a value of the second weight. The method can include computing the quality metric characterizing the given match by: computing a first frequency of terms in the first source and a second frequency of terms in the second source; and producing the quality metric based on a product of numerical values of the first and second frequencies. The method can include normalizing the quality metric to range between predetermined limits (e.g., between 0 and 1).

15 The method can include generating a set of terms from the second source corresponding to a term in the first source and having predetermined quality metrics characterizing the match between the term and each of the set of terms. A number of terms in the set of terms is specified by a user. The match between the term and each of the set of terms is based on matching names appearing in the terms. The match between the term and each of the set of terms is based on matching descriptions appearing in the terms.

 In another aspect, in general, a computer-readable medium storing a computer program, the computer program including instructions for causing a computer to: store a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; generate a set of rules for each source based on a corresponding one of the stored specifications; and match data elements of different sources and determine a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source.

In another aspect, in general, a system includes: a data storage system storing a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; and a data processing system coupled to the data storage system configured to generate a set of rules for each source based on a corresponding one of the stored specifications, and match data elements of different sources and determine a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source.

In another aspect, in general, a system includes: means for storing a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; and means for processing data elements from the sources, including generating a set of rules for each source based on a corresponding one of the stored specifications, and matching data elements of different sources and determining a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source.

Aspects can include one or more of the following advantages.

Generally, when searching for a match between one or more key words and some text (e.g., a web page), a search process can display to a user why a given match occurred, for example, by highlighting the appearance of the key word(s) within the text (e.g., by making the key word(s) bold). In some examples, the techniques described herein may be used to identify changes in different versions of documents. Further, source or key terms may be visually linked to target terms by a relationship diagram that can include details such as matching scores and grades. When performing matches between two data elements that may represent the metadata for the same type of data, the reason why the match occurred (or did not occur) may be more complicated than the existence of an exact match between respective key words. For example, terms appearing in the data elements may have been expanded or transformed (e.g., using stemming) and relationships between matching terms may be based on finding synonyms

or classifying terms into categories (called “classes”). The procedure used to perform the matching can use rules to characterize the match quality by assigning each match a grade. The grades can be stored in association with the matches to indicate the match quality.

In cases in which there are a large number of sources of metadata, differences
5 between different sources can be accounted for so that the sources can be processed efficiently any number of times as the matching process is repeated. A pre-processing procedure enables generation of source processing information that enables processing of data elements directly from the sources by providing information needed to interpret and/or transform the data elements and information needed to define rules for
10 characterizing the matches.

Other features and advantages of the invention will become apparent from the following description, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of a system for executing graph-based computations.

15 FIG. 2 is a flowchart of an exemplary metadata processing procedure.

FIG. 3 shows phases of an exemplary automated matching process.

FIG. 4 is an exemplary graph-based implementation of an automated matching
process.

FIG. 5 is an exemplary output from the graph-based implementation of the
20 automated matching process of FIG. 4.

FIGS. 6-8 are screenshots of example rules and interfaces for managing the rules.

FIGS. 9-12 are screenshots of an exemplary metadata interface.

DESCRIPTION

Business analysts may maintain multiple data dictionaries that include listings of
25 business characteristics of data elements across many systems. Data dictionaries (or metadata repositories) are repositories of information about data such as meaning, relationships to other data, origin, usage, and format. As such, data dictionaries facilitate standardization of definitions of terms and consistency of use of these terms. In some scenarios, an enterprise wide data dictionary may be maintained to capture metadata
30 about the data used within the enterprise.

A data element to be matched can have a name portion that identifies the data element using one or more descriptive terms, and may optionally have a description portion that describes the data element or various properties characterizing the element. The names and corresponding descriptions contained in the different dictionaries may be in a variety of formats. For example, data dictionaries may be developed at different times and as parts of different systems that may be independently maintained. For at least this reason, there may not be a commonly adopted naming standard. As such, an advantage of the metadata processing techniques described in this application is harmonization of names and descriptions across diverse data dictionaries. Additionally, by providing a quality metric or score quantifying the matches of data elements, automated metadata processing may direct a business analyst's attention to only a fraction of the matches that require human analysis. For example, the analyst may be interested in only those matches that are scored by the metric to be close matches.

FIG. 1 shows an exemplary data processing system 100 in which the metadata processing techniques can be used. The system 100 includes sources 102 that may include one or more sources of data and/or metadata such as storage devices or connections to online data streams, each of which may store data and/or metadata in any of a variety of storage formats (e.g., database tables, spreadsheet files, flat text files, or a native format used by a mainframe). In some cases, a source stores metadata independently from the data that are described by that metadata. In some cases, the metadata is stored within the same data structures as the data that are described by that metadata, or stored in association with the data using links or pointers, for example. In one example, the sources 102 are associated with multiple data storage systems that are to be integrated to form a single master data storage system. In the process of integrating the systems, it may be necessary to determine matches between metadata describing corresponding data that is to be merged. For example, it may be necessary to determine which fields from a customer list from one source store the data values representing the same attribute as fields from a customer list from another source (e.g., a "social security #" field from one source is the same attribute as a "SSN" field from another source). The matches between the data elements can then be used to integrate data storage systems. An execution environment 104 includes a pre-processing module 106 that reads the sources

102 and generates source processing information 122 for the metadata sources based on a source registry 114, and an execution module 112 that performs the metadata processing to determine matches and record quality information based on the source processing information 122 and reference information 124. A data storage system 116 stores the source registry 114, source processing information 122, and reference information 124, as described in more detail below. The execution environment 104 may be hosted on one or more general-purpose computers under the control of a suitable operating system, such as the UNIX operating system. For example, the execution environment 104 can include a multiple-node parallel computing environment including a configuration of computer systems using multiple central processing units (CPUs), either local (e.g., multiprocessor systems such as SMP computers), or locally distributed (e.g., multiple processors coupled as clusters or MPPs), or remotely, or remotely distributed (e.g., multiple processors coupled via a local area network (LAN) and/or wide-area network (WAN)), or any combination thereof. Storage devices providing the sources 102 may be local to the execution environment 104, for example, being stored on a storage medium connected to a computer running the execution environment 104 (e.g., hard drive 108), or may be remote to the execution environment 104, for example, being hosted on a remote system (e.g., mainframe 110) in communication with a computer running the execution environment 104, over a remote connection.

The data storage system 116 is also accessible to a development environment 118 in which a developer 120 is able to configure the pre-processing module 106 and execution module 112. The development environment 118 is, in some implementations, a system for developing applications as dataflow graphs that include vertices (components or datasets) connected by directed links (representing flows of work elements) between the vertices. For example, such an environment is described in more detail in U.S. Publication No. 2007/0011668, entitled "Managing Parameters for Graph-Based Applications," incorporated herein by reference. Both the pre-processing module 106 and the execution module 112 can be configured with the ability to process multiple sources in parallel with each module implemented as a dataflow graph that receives a flow of input data from the sources 102 and provides a flow of output data, for example, as a stream of potential matches between data elements in the sources 102.

The pre-processing module 106 prepares the source processing information 122 based on information from the sources according to the source registry 114. The source registry 114 includes a specification for each source to be processed that specifies: access information indicating how to access the source, format information indicating the format of the data elements in the source, and an identification of the particular data elements in the source that are to be included in the matching process. Each specification can be stored as a row in a table, for example. The pre-processing module 106 reads the data elements from the sources identified by the source registry 114, and generates the source processing information 122. The source processing information 122 includes any information in addition to the format information from the source registry 114 needed to interpret and/or transform the data elements to extract terms and descriptions that are to be used in the matching process. For example, a different transformation function can be stored for transforming each of a variety of formats into a common format to be used in the matching process.

The source processing information 122 also includes information needed to define rules for characterizing the matches. Some of the rules for determining grades may depend on characteristics of the data elements. So each source can have a corresponding set of rules, and together the different sets of rules can be used to determine a grade for a given match.

The execution module 112 uses the source processing information 122 generated by the pre-processing module 106 and reference information 124 stored in the data storage system 116 accessible to the execution environment 104. The execution module 112 generates words to be matched from the terms and descriptions extracted from the data elements, and performs the matching process to yield matches between data elements. The matching process includes storing data characterizing the quality of the matches, as described in more detail below. In some implementations, matching is performed between each source listed in the source registry 114 and a canonical metadata repository (CMR) stored in the data storage system 116. For example, a CMR can represent an enterprise data dictionary that is to serve as a master reference dictionary in an enterprise environment. Data elements in the source are compared with canonical attributes in the CMR to find matches.

A match between a term from a name or description or other metadata stored in a data element of a source and a term from a name or description of a canonical attribute represented in the CMR indicates that the matched data element potentially has the same meaning as the canonical attribute. In some implementations a match is determined
5 based on a combination of matching a canonical attribute name with a data element name and matching a canonical attribute description with a data element description.

In some implementations, matching is performed such that each source is compared to all other sources in addition to the CMR to find matches between respective data elements or between a data element and a canonical attribute. In some
10 implementations, the matching enables a comparison between sources by adding unmatched terms from a previous comparison with a source to the CMR between iterations. This process can thus obviate a need for ‘all to all’ processing. For example, an exemplary matching process uses the following sequence in which $CMR(n)$ is the n th iteration of updating the CMR with selected terms from a name or description of a data
15 element in the previous source comparison that was unmatched:

- compare Source1 to $CMR(0)$
- add all Source1 terms that are unmatched to $CMR(0)$ creating $CMR(1)$
- compare source2 to $CMR(1)$
- add all Source 2 terms that are unmatched to $CMR(1)$ creating $CMR(2)$
- 20 • compare Source3 to $CMR(2)$
- etc.

In one example of a matching process performed by the execution module 112, the process starts by normalizing, expanding, and cleansing terms extracted from a data element into a standard form and identifying terms that correspond to a name of an
25 attribute defined by the metadata in a data element and terms that correspond to a description of that attribute. The cleansing may include selectively filtering certain punctuation (e.g., underbars, dashes, etc.), converting a case (e.g., to lowercase), and removing extra spaces. Lists of predetermined words (e.g., “a, also, and” etc.) including standard linguistic “nuisance words” or “stop words” can also be removed from the
30 terms. In some implementations, the reference information 124 can include lookup files including lists of stop words, acronyms, and aliases. For example, a stop words lookup

file that includes a list of stop words may be used to assist in the cleansing. Users may add or remove words from this list by modifying the lookup file. The process also includes expanding abbreviations and acronyms into fully worded phrases, and expanding terms in the name or description into common aliases. Again, an acronym
5 lookup file can be used to assist in this process. As such, users may modify the acronym lookup file through an interface. In some examples, the interface may include controls to solicit user feedback in the approval and notification of any changes to the file.

In some implementations, a synonym lookup file may include synonyms for words in the terms and descriptions to support different words that can have similar
10 meanings. For example, “agency”, “authority”, “bureau”, and “organization” have similar meanings to the word “office” in a context of a particular term or description. In some examples, such synonyms can include international equivalents of certain words in the terms and descriptions. For example, “liter” may be a synonym for “litre”. Additionally, the synonym lookup file may also provide support for addressing aliasing
15 words such as “address1” and “address2” in one data source that correspond to simply “address” in a different data source. Further, some words in the terms and descriptions may be transformed to their stem forms in an effort to normalize differences between words. In some implementations, the transformations may also account for conjugations, tenses, and/or pluralities by, for example, adjusting suffixes. As an example,
20 “acquisition” may be transformed to “acquisit” and “parameters” may be transformed to “paramet”. In some implementations, a group of cleansed words is generated for the name and a group of cleansed words is generated for the description.

The process also includes determining a “class word” for each attribute. A class word is a word defines the content and role of a piece of data described by an attribute.
25 An exemplary set of class words is: amount, code, date, time, date-time, class, description, identifier, image, indicator, name, address, number, quantity, percent, rate, sound, and text. To determine a class word for a given attribute, terms in the attribute name can be scanned from right to left to identify the first match to one of the predetermined set of class words. For example, the class word corresponding to the
30 attribute name “start date” is “date.” Some class words are determined based on words appearing in the name and/or description without necessarily requiring a match to the

determined class word (e.g., an attribute Name “title” may correspond to the class word “text”, “indicator”, or “name”, depending on terms appearing in the attribute description).

The computation of the similarity between a term and description in a source and terms and descriptions in the CMR may be performed by a modified TF-IDF process as described below. A “Term Frequency – Inverse Document Frequency” (TF-IDF) weight is a statistical measure used to evaluate how important a word in a term appearing in an attribute name or description is to a given data element and to a source of data elements. The importance of a word increases proportionally to the number of times a word appears in the data element (e.g., including the name and the description). But the importance of the word is offset by the frequency of the word in the attributes represented in the CMR.

The TF-IDF weight diminishes the weight of words that occur very frequently in the CMR and increases the weight of words that occur rarely. By way of example, consider the word “code” a common word in data dictionary terms. If the word “code” occurs in source and target terms, the match between the occurrences would not properly be explained since “code” is a common string. However, if the source and target terms both contained the term “disputed” the match between the terms is better explained, and thus the word “disputed” in both terms contributes better to facilitating a match between the two terms.

An exemplary weight vector for a “document” d (e.g., representing at least a portion of a data element from which representative terms are taken) of a set of documents D (e.g., representing a set of data elements in a source)

is $\mathbf{v}_d = [w_{1,d}, w_{2,d}, \dots, w_{N,d}]^T$, where

$$w_{t,d} = \text{tf}_t \cdot \log \frac{|D|}{|\{t \in d\}|}$$

and

* tf_t is term frequency of term t in document d (a local parameter)

* $\log \frac{|D|}{|\{t \in d\}|}$ is inverse document frequency (a global parameter).

$|D|$ is the total number of documents in the document set, and $|\{t \in d\}|$ is the number of documents containing the term t .

In one example, an attribute name and description contain 8 words and the word “branch” appears 2 times. The term frequency (TF) for “branch” is 0.25 (2 words of 8). In the CMR, there are ~3,300 attributes and “branch” appears in 12 of these. Then, the inverse document frequency (IDF) is calculated as $\ln(3,300 / 12) = 5.61$. The TF-IDF weight is the product of these quantities: $0.25 * 5.61 = 1.4$. In another example, an attribute name and description contain 8 words and the word “code” appears 1 time. The TF for “code” is 0.125 (1 word of 8). In the CMR, there are ~3,300 attributes and “code” appears in 900 of these. Then, the IDF is calculated as $\ln(3,300 / 900) = 1.99$. The TF-IDF weight is the product of these quantities: $0.125 * 1.99 = 0.16$. So, in these examples, the word “branch” with a weight of 1.4 is likely to be more important than the word “code” with a weight of 0.16.

In some implementations, the similarity between the terms and descriptions can be absolute numbers which be normalized so they range from, for example, 0 to 1. As such, for each source term, a result of the matching computation based on the modified TF-IDF approach may be a set of, e.g., N CMR terms that best match the source term in a name of an attribute and N CMR descriptions that best match the corresponding source description of the attribute. The number N may be an input parameter to the matching system. In some examples, a value of N=3 may be used.

The matching process can take into account the TF-IDF weights of words when performing matches to match words of a data element with highest weights. The matching process can use any of a variety of matching techniques for determining when words extracted from terms in the name or description match words extracted from attributes of the CMR. For example, U.S. Publication No. 2009/0182728, entitled “MANAGING AN ARCHIVE FOR APPROXIMATE STRING MATCHING,” incorporated herein by reference, describes techniques for approximate string matching.

The output of the matching process include lists of data elements associated with the respective attributes in the CMR to which those data elements match. In some implementations, matches correspond to matching words in both the name and description. The output can optionally include words in the name and description that matched, and can include lists of data elements with no name or description matches.

In association with each match, the execution module 112 is able to store explanatory information including a grade characterizing the quality of the match, an explanation of how that grade was obtained (e.g., rule firing), and an explanation of why the match occurred. A grade can be determined based on a “name match” between cleansed words for a data element name and cleansed words for a canonical attribute name, a “description match” between cleansed words for a data element description and cleansed words for a canonical attribute description, and “class match” between a class word associated with a data elements and a class word associated with a canonical attribute. A metric for the match can indicate how close (e.g., with respect to a distance measure) the match is.

For example, a grade of “AA” can be assigned based on a rule that determines there was an exact name match (e.g., at least one cleansed word matches exactly), a high description match (e.g., metric >75% for a match between cleansed words), and the same class words. A grade of “AB” can be assigned based on a rule that determines there was a high name match (e.g., metric >95% for a match between cleansed words), and a high description match (e.g., metric >70% for a match between cleansed words), and the same class words. If class words are not the same, the grade is typically set much lower (e.g., “DA” and lower) since there is likely a meaningful difference in what the data described by the data element represents.

The execution environment 104 also includes a user interface to allow a user 124 interact with the output of the matching process to review and accept matches based on the associated grades and other information. The user interface can include a list of data elements that include terms in the name and/or description that matched to a term in a canonical attribute or another data element, and the list can include links to the original data elements in the sources or to copies of the original data elements. In some implementations, matches are accepted based on comparisons of grades to thresholds without requiring user input. Unmatched data elements can be reviewed and used to update the reference information 124 based on user input. For example, a user can review terms of art or unexpanded acronyms for addition to a reference containing synonyms or a reference containing expansions of acronyms, respectively.

FIG. 2 shows a flowchart for an exemplary procedure 200 for pre-processing the metadata from the sources 102 and executing the matching on the metadata. The procedure 200 includes storing (202), in the data storage system 116, a specification for each of multiple sources 102, each specification including information identifying one or more data elements of the corresponding source. The procedure 200 includes processing, in a data processing system providing the execution environment 104 coupled to the data storage system 116, data elements from the sources. The processing includes generating (204) a set of rules for each source based on a corresponding one of the stored specifications, and matching (206) data elements of different sources and determining (208) a quality metric (e.g., grades) characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source. After the sources are processed, results are stored (210) identifying the determined matches. As additional sources are added, the procedure 200 can be repeated, processing the additional sources.

FIG. 3 shows example phases of an automated matching process performed by a data processing system (e.g., system 100 shown in FIG. 1) in which a source (e.g., a source data dictionary) is matched against a CMR (e.g., an enterprise dictionary). For example, the CMR may be compiled from a variety of sources over a period of time.

During a prepare phase 310, the information in the source data dictionary may be converted into a format that is compatible with the metadata processing techniques described above. For example, pre-processing module 106 (FIG. 1) may be used in this phase to map source data structures to a common record format.

5 In an implementation of the prepare phase 310, the system may accept input from a user to generate a data structure for a particular source and register a corresponding source data dictionary in a registration form. Subsequently, the registration form may be read and the source data dictionary may be converted into a format that is compatible with the matching system. For example, in a graph-based system, a “Generate Metadata”
10 dataflow graph may be run to load the source data dictionary. The graph may read the registration form and generate metadata that is used in the matching process. In some implementations, the graph may also generate sets of parameters for configuring additional dataflow graphs and rule files for configuring a rules engine for loading business terms, descriptions, acronyms, and links corresponding to the source data
15 dictionary. Once the metadata and the parameter sets and rule files are generated a matching dataflow graph can be run. In some examples, more than one source dictionary can be run through the “Generate graph” simultaneously.

During a parse phase 320, the terms in the source dictionary can be processed to extract individual words. As described in connection with the cleansing process above,
20 punctuation without semantic meaning may be removed from the terms. In some implementations, certain characters such as “\$” and “%” can be left within the terms because they may have semantic meaning. In some implementations, the terms can further be classified into class words as described above.

During a standardize phase 330, variability in terms and descriptions can be
25 reduced. In this phase, the terms and descriptions can be further cleansed to remove, for example, stop words, expand abbreviations, and map aliases. During an approve phase 340 the source terms may be matched with canonical terms using one or more matching techniques. For example, a TF-IDF weight as described above can be used to evaluate how important a word in a term or description is to a given data element and to a source
30 of data elements. In some examples, a “fuzzy matching” technique may be employed to

perform the matching process (e.g., as described in U.S. Publication No 2009/0182728, incorporated herein by reference).

Once the weights are assigned, one or more user developed rules can be used to grade the matches during the match phase 350. For example, the matches may be graded as described above. In one implementation, grades such as “A”, “B”, “C” or “F” may be assigned to matches depending on the quality of the match, where “A” may be a grade specifying a highest quality match, and “F” may be a grade specifying a poorest quality match. Finally, during a score phase 360, a user may review and accept the matches based on grades assigned to the matches. In some implementations, the user may also propose new canonical terms for the CMR.

FIG. 4 shows an example graph-based approach for implementing the matching technique described above. The matching graph 400 may be called by, for example, the “Generate Metadata” graph described above once the source data is ready for matching. As such, a read target component 402 and a read source component 404 begin the matching process by reading corresponding target and source files 406, 408 respectively. The target file 406 may include CMR terms and descriptions from the CMR. Subsequently, a mapping components 410, 412 may perform target and source specific mapping processes on the terms and descriptions in the target and source files respectively. For example, multiple instances and variations of certain terms and descriptions may be mapped together by the mapping components 410, 412. As such, the process allows for a source term to be matched against multiple targets with match scores for each match so that a user using the workflow can support a process to determine the ‘best’ match.

Subsequently, using the classification process described in detail above, the classification components 414, 416 may determine class words for the terms and descriptions in the target and source files 406, 408. In some situations, matching terms using string text can be computationally slow. As such, text terms can be converted, i.e., tokenized, to numeric keys and thus dramatically speed up the matching process. For example, components 415 may perform this conversion on source and target terms. At the conclusion of the matching the keys may be decoded back to the original text terms. Once the source and target terms and descriptions are mapped and standardized, a

matching subgraph 418 performs the matching of the words in the terms and descriptions in the source with the words in the terms and descriptions in the target. For words that match in the source and target, the matching subgraph 418 can return the term or description with an accompanying quality metric indicating the closeness of the match.

5 The join component 420 can perform a join using a source attribute name derived from an original source of the particular term or description and output the results as matched outputs 422.

An example output 500 of the graph 400 of FIG. 4 is shown in FIG. 5. As shown, a source term “milestone identifier” 502 corresponds to at least three CMR terms (or any
10 user-specified number of ‘best matches’), namely, “milestone name”, “milestone identifier” and “milestone date” 504. Quality metrics 506 quantify the degree of similarity of the source term with each of the CMR terms. For example, the similarity measure for the second item in the output, “milestone identifier” is 1, indicating a perfect match. In implementations, the best three term matches of the term “milestone identifier”
15 502 can be joined with best three description matches, and the nine term/description communications can be sent to a set of business rules that to decide a best match between the source and CMR terms.

In some examples, the business rules can be based on similarity of the term name and description matches as well as computed class words for the source and CMR terms
20 as described in detail above. In some implementations, an output of the business rules is a best match as well as a letter grade for that match. Users may set a predetermined quality of the match to correspond to the letter grades. Further, users may specify a predetermined range of accepted grades. For example, a user may only accept match quality that corresponds with grades A through BC (or B minus).

25 FIG. 6 shows example business rules 600 used in a matching processing. Triggers 602 act as inputs to the business rules 600, which produce corresponding outputs 604 as shown. Similarity measures such as a similarity of names between source and CMR terms may be quantified as a numeric value ranging from 0 to 1. As such, a first business rule 606 can be interpreted as follows: If a similarity of names between source
30 and CMR terms is greater than 0.95, a similarity of descriptions between the source and CMR description is greater than 0.70, the names of the two terms are identical, and the

class words corresponding to the two terms are the same, then the match grade between the two terms is an “AA” (or “A plus”). In some examples, if any of the triggers 602 in the business rule above evaluate to false, then a second business rule 608 is read, which states as follows: If the similarity of names between the source and CMR terms is greater than 0.95, the similarity of descriptions between the source and CMR descriptions is greater than 0.70, the names of the two terms are *not* identical, and the class words corresponding to the two terms are the same, then the match grade between the two terms is an “AB” (or “A minus”). In examples, the business rules 600 are read one after the other until a business rule is read where all of the input triggers evaluate to true. A user may define a minimum grade that is triggered by the business rules 600.

In some implementations, an analyst may be able to perform test runs of the business rules 600 by using test data. FIG. 7 is a screenshot 700 of an example test run. As shown, for each of the test data items 702, a match grade 704 is generated and displayed to the analyst. In addition, similarity scores 706 and class word matches 708 are displayed corresponding to each item 702. The analyst can select a test item 710, which is then highlighted and, thus can view more information about the item 710.

FIG. 8 is an example screenshot 800 showing exactly which rule fired for a particular test data item (e.g., item 702 of FIG. 7). In addition, an analyst may also view information about how many times a business rule (e.g., one of rules 600 of FIG. 6) fired. As shown, in some implementations, one or more graphical buttons 802 may be used to indicate whether a trigger corresponding to a button 802 evaluated to true. The buttons corresponding to rule 5 are all depressed indicating that rule 5 has fired for a particular test data. Further, a number of times each rule has fired may be displayed. For example, rule 1 has fired 77 times while rule 5 has fired 303 times. This approach, termed a rapid ‘iterative test, modify, and rerun’ approach, may be used to optimize the matching rules. An analyst can use such an interface to see if there are terms that were inappropriately matched.

FIGS. 9-12 are screenshots of an example metadata interface 900 for displaying metadata information to a user. In some implementations, the results of the matching process (e.g., matched outputs 422 of FIG. 4) can be incorporated into the interface 900. As shown in FIG. 6, the interface 900 can provide a text-field 902 for searching the

metadata repository. The terms may be further stored as hierarchical groups (e.g., “Business” 904) and child groups (e.g., “Baseline” 906).

FIG. 10 shows the interface 900 displaying details of a term “acceptance date” 1002 belonging to the “Baseline” 906 group. In some implementations, a user may right click on the term “acceptance rate” 1002 and request to see relationships pertaining to the term “acceptance rate” 1002.

FIG. 11 shows a diagrammatic representation 1102 of the matches to the Baseline term “acceptance date” 1002. In some examples, information about the source of the matches can also be displayed to the user.

Referring now to FIG. 12, in some implementations, a tabular view of the matches may be available to the user. The interface 900 can be configured to display only approved matches. A user may use the “approval workflow” tab 1202 to review matches including pending and/or rejected matches for the terms.

The metadata processing approach described above can be implemented using software for execution on a computer. In some implementations, the process may automate the matching process for an unlimited number of dictionaries in a very short run. For instance, the software forms procedures in one or more computer programs that execute on one or more programmed or programmable computer systems (which may be of various architectures such as distributed, client/server, or grid) each including at least one processor, at least one data storage system (including volatile and non-volatile memory and/or storage elements), at least one input device or port, and at least one output device or port. The software may form one or more modules of a larger program, for example, that provides other services related to the design and configuration of dataflow graphs. The nodes and elements of the graph can be implemented as data structures stored in a computer readable medium or other organized data conforming to a data model stored in a data repository.

The software may be provided on a storage medium, such as a CD-ROM, readable by a general or special purpose programmable computer or delivered (encoded in a propagated signal) over a communication medium of a network to the computer where it is executed. All of the functions may be performed on a special purpose computer, or using special-purpose hardware, such as coprocessors. The software may

be implemented in a distributed manner in which different parts of the computation specified by the software are performed by different computers. Each such computer program is preferably stored on or downloaded to a storage media or device (e.g., solid state memory or media, or magnetic or optical media) readable by a general or special
5 purpose programmable computer, for configuring and operating the computer when the storage media or device is read by the computer system to perform the procedures described herein. The inventive system may also be considered to be implemented as a computer-readable storage medium, configured with a computer program, where the storage medium so configured causes a computer system to operate in a specific and
10 predefined manner to perform the functions described herein.

A number of embodiments of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, some of the steps described above may be order independent, and thus can be performed in an order different from that described.

15 It is to be understood that the foregoing description is intended to illustrate and not to limit the scope of the invention, which is defined by the scope of the appended claims. For example, a number of the function steps described above may be performed in a different order without substantially affecting overall processing. Other embodiments are within the scope of the following claims.

What is claimed is:

1. A method, including:

storing, in a data storage system, a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; and

processing, in a data processing system coupled to the data storage system, data elements from the sources, including

generating a set of rules for each source based on a corresponding one of the stored specifications, and

matching data elements of different sources and determining a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source.

2. The method of claim 1 wherein the set of rules for each source produces one or more grades corresponding to the quality metric characterizing the given match.

3. The method of claim 2 further including providing explanatory information corresponding to the one or more grades.

4. The method of claim 2 wherein the given match includes a match between names corresponding to the first and second data elements respectively and the one or more grades are based on the quality metric characterizing the given match.

5. The method of claim 2 wherein the given match includes a match between descriptions corresponding to the first and second data elements respectively and the one or more grades are based on the quality metric characterizing the given match.

6. The method of claim 2 further including:
classifying terms appearing in the first and second data elements into one or more classes;
assigning one or more class words for each of the terms in the first and second data elements;
comparing the one or more class words corresponding to terms in the first and second data elements respectively to generate the quality metric for the given match; and
assigning the one or more grades based on the quality metric characterizing the given match.

7. The method of claim 1 wherein the quality metric characterizing the given match includes a distance measure metric.

8. The method of claim 1 wherein a first grade is assigned to an output of a first rule of the set of rules and a second grade is assigned to a second, different output of a second rule of the set of rules, wherein the first grade indicates a better quality metric characterizing the given match relative to the second grade.

9. The method of claim 1 wherein the set of rules are based on a similarity of names appearing in the first and second data elements respectively.

10. The method of claim 1 wherein the set of rules are based on a similarity of descriptions appearing in the first and second data elements respectively.

11. The method of claim 1 further including providing a user an ability to provide input for generating one or more rules of the set of rules for quantifying a quality of a match between the first data element and the second data element .

12. The method of claim 1 wherein each of the rules in the set of rules include trigger inputs and outputs based on the trigger inputs.

13. The method of claim 1 wherein each of the rules in the set of rules are read sequentially until all trigger inputs of a given rule in the set of rules evaluate to true.

14. The method of claim 1 wherein the quality metric characterizing the given match is based on a measure of a number of times a word occurs in a term of the first or second data element and the number of times a word occurs in a set of terms from the first or second source.

15. The method of claim 1 further including computing the quality metric characterizing the given match by:

providing a first weight to terms that occur with a first frequency in the first or second source; and

providing a second weight to terms that occur with a second frequency in the first or second source, wherein a value of the first weight is less than a value of the second weight.

16. The method of claim 1 further including computing the quality metric characterizing the given match by:

computing a first frequency of terms in the first source and a second frequency of terms in the second source; and

producing the quality metric based on a product of numerical values of the first and second frequencies.

17. The method of claim 1 further including normalizing the quality metric to range between predetermined limits.

18. The method of claim 1 further including generating a set of terms from the second source corresponding to a term in the first source and having predetermined quality metrics characterizing the match between the term and each of the set of terms.

19. The method of claim 18 wherein a number of terms in the set of terms is specified by a user.

20. The method of claim 18 wherein the match between the term and each of the set of terms is based on matching names appearing in the terms.

21. The method of claim 18 wherein the match between the term and each of the set of terms is based on matching descriptions appearing in the terms.

22. A computer-readable medium storing a computer program, the computer program including instructions for causing a computer to:

store a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source;

generate a set of rules for each source based on a corresponding one of the stored specifications; and

match data elements of different sources and determine a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source.

23. A system, the system including:

a data storage system storing a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; and

a data processing system coupled to the data storage system configured to generate a set of rules for each source based on a corresponding one of the stored specifications, and match data elements of different sources and determine a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source.

24. A system, the system including:

means for storing a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; and

means for processing data elements from the sources, including

generating a set of rules for each source based on a corresponding one of the stored specifications, and

matching data elements of different sources and determining a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules generated for the first source and the set of rules generated for the second source.

AMENDED CLAIMS

received by the International Bureau on 26 May 2011 (26.05.2011)

1. A method, including:

storing, in a data storage system, a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; and

processing, in a data processing system coupled to the data storage system, data elements from the sources, including

generating a set of rules for the sources based on the stored specifications, and

matching data elements of different sources and determining a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules.

2. The method of claim 1 wherein the set of rules for each source produces one or more grades corresponding to the quality metric characterizing the given match.

3. The method of claim 2 further including providing explanatory information corresponding to the one or more grades.

4. The method of claim 2 wherein the given match includes a match between names corresponding to the first and second data elements respectively and the one or more grades are based on the quality metric characterizing the given match.

5. The method of claim 2 wherein the given match includes a match between descriptions corresponding to the first and second data elements respectively and the one or more grades are based on the quality metric characterizing the given match.

6. The method of claim 2 further including:
classifying terms appearing in the first and second data elements into one or more classes;

assigning one or more class words for each of the terms in the first and second data elements;

comparing the one or more class words corresponding to terms in the first and second data elements respectively to generate the quality metric for the given match; and

assigning the one or more grades based on the quality metric characterizing the given match.

7. The method of claim 1 wherein the quality metric characterizing the given match includes a distance measure metric.

8. The method of claim 1 wherein a first grade is assigned to an output of a first rule of the set of rules and a second grade is assigned to a second, different output of a second rule of the set of rules, wherein the first grade indicates a better quality metric characterizing the given match relative to the second grade.

9. The method of claim 1 wherein the set of rules are based on a similarity of names appearing in the first and second data elements respectively.

10. The method of claim 1 wherein the set of rules are based on a similarity of descriptions appearing in the first and second data elements respectively.

11. The method of claim 1 further including providing a user an ability to provide input for generating one or more rules of the set of rules for quantifying a quality of a match between the first data element and the second data element.

12. The method of claim 1 wherein each of the rules in the set of rules include trigger inputs and outputs based on the trigger inputs.

13. The method of claim 1 wherein each of the rules in the set of rules are read sequentially until all trigger inputs of a given rule in the set of rules evaluate to true.

14. The method of claim 1 wherein the quality metric characterizing the given match is based on a measure of a number of times a word occurs in a term of the first or second data element and the number of times a word occurs in a set of terms from the first or second source.

15. The method of claim 1 further including computing the quality metric characterizing the given match by:

providing a first weight to terms that occur with a first frequency in the first or second source; and

providing a second weight to terms that occur with a second frequency in the first or second source, wherein a value of the first weight is less than a value of the second weight.

16. The method of claim 1 further including computing the quality metric characterizing the given match by:

computing a first frequency of terms in the first source and a second frequency of terms in the second source; and

producing the quality metric based on a product of numerical values of the first and second frequencies.

17. The method of claim 1 further including normalizing the quality metric to range between predetermined limits.

18. The method of claim 1 further including generating a set of terms from the second source corresponding to a term in the first source and having predetermined quality metrics characterizing the match between the term and each of the set of terms.

19. The method of claim 18 wherein a number of terms in the set of terms is specified by a user.

20. The method of claim 18 wherein the match between the term and each of the set of terms is based on matching names appearing in the terms.

21. The method of claim 18 wherein the match between the term and each of the set of terms is based on matching descriptions appearing in the terms.

22. A computer-readable medium storing a computer program, the computer program including instructions for causing a computer to:

store a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source;

generate a set of rules for the sources based on the stored specifications; and

match data elements of different sources and determine a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules.

23. A system, the system including:

a data storage system storing a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; and

a data processing system coupled to the data storage system configured to generate a set of rules for the sources based on the stored specifications, and match data elements of different sources and determine a quality metric characterizing a given match between a first data element of a first source and a second data element of a second source according to the set of rules.

24. A system, the system including:

means for storing a specification for each of multiple sources, each specification including information identifying one or more data elements of the corresponding source; and

means for processing data elements from the sources, including
generating a set of rules for the sources based on the stored specifications, and
matching data elements of different sources and determining a quality metric
characterizing a given match between a first data element of a first source and a second
data element of a second source according to the set of rules.

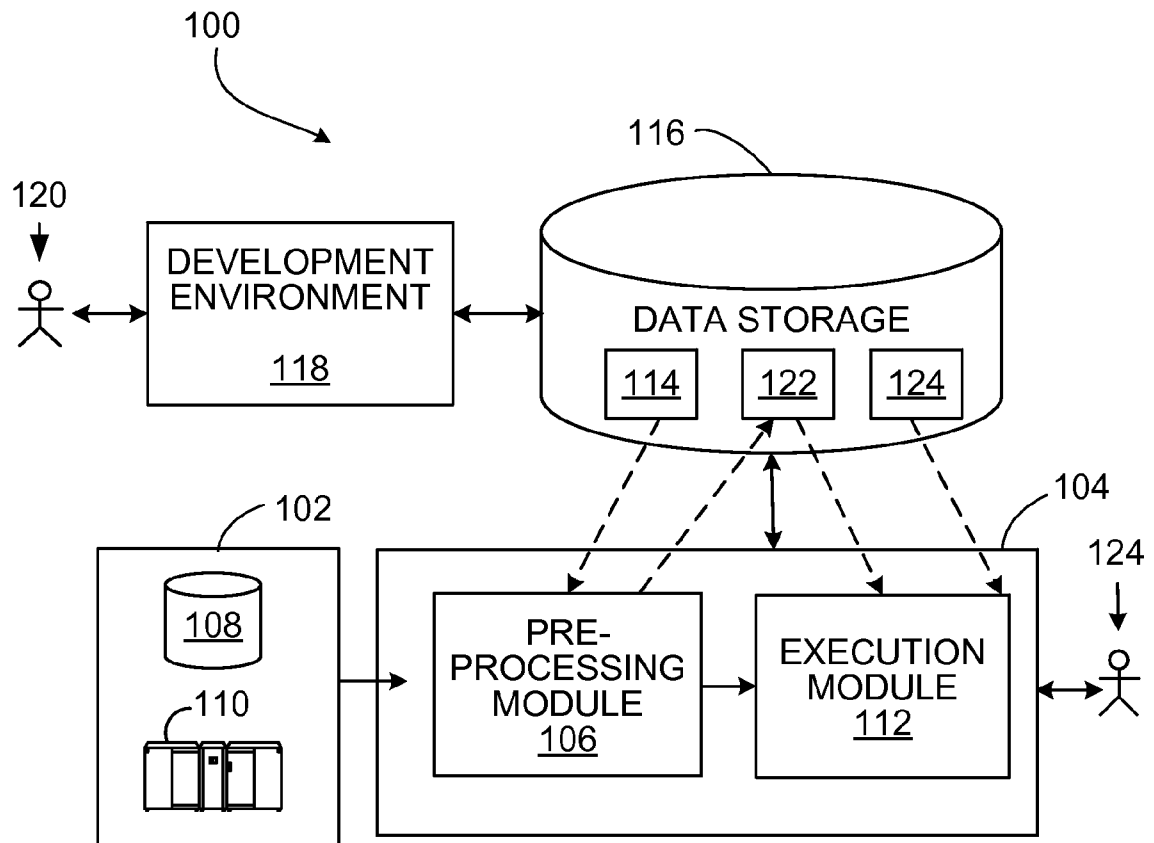


FIG. 1

2/13

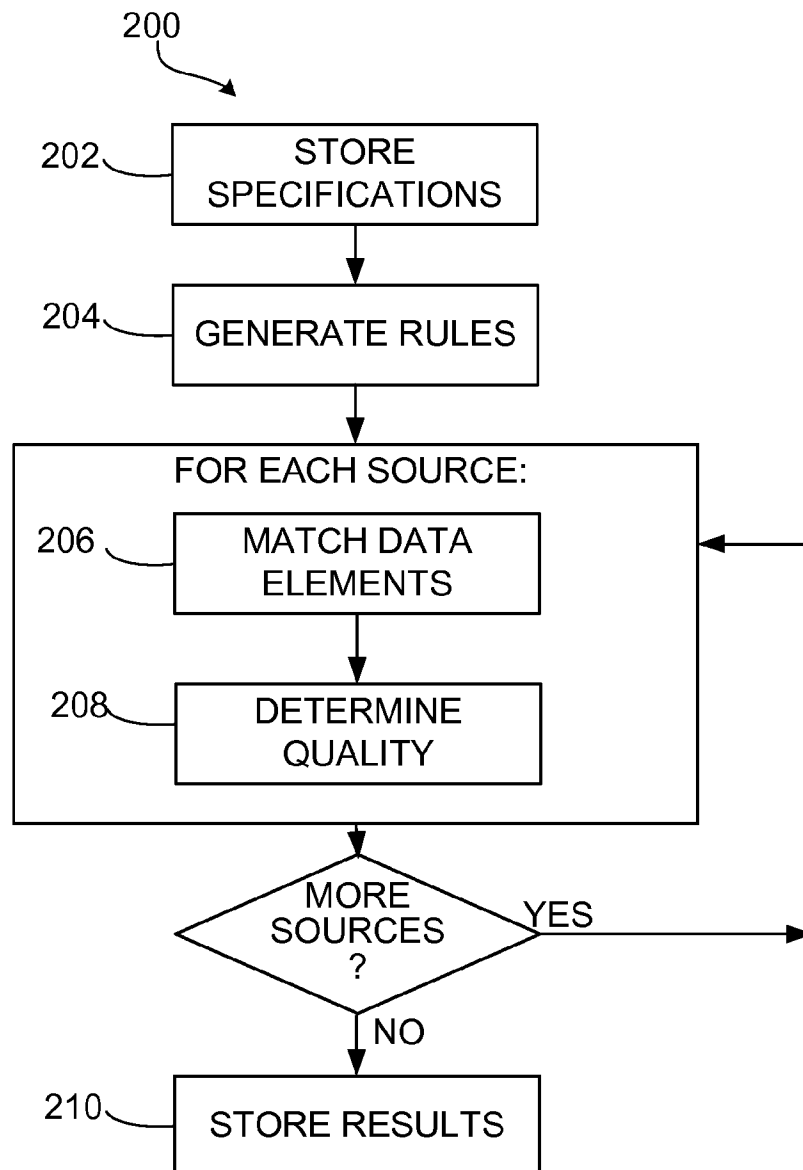


FIG. 2

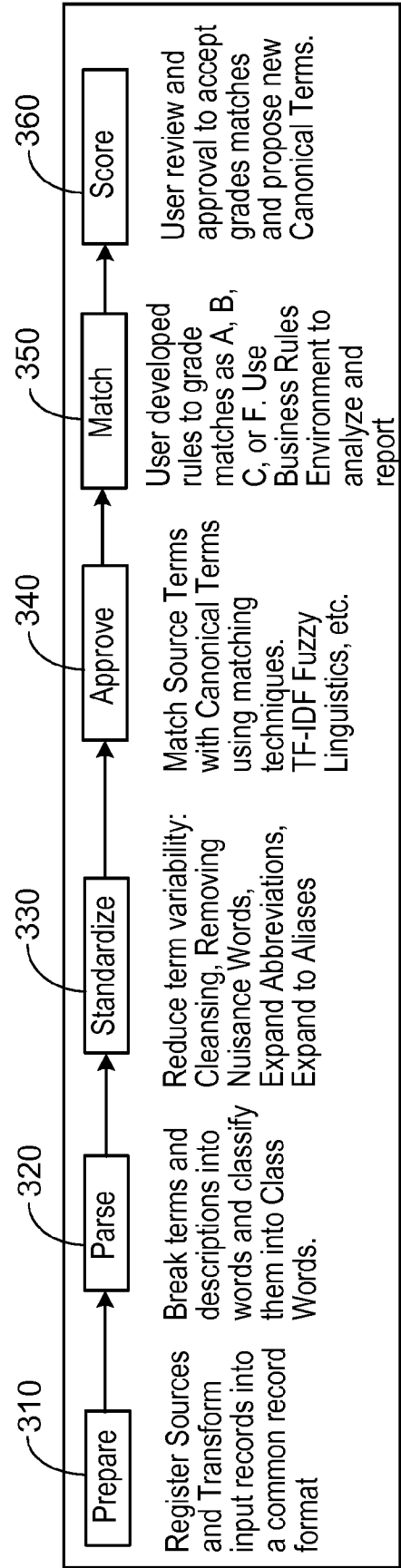


FIG. 3

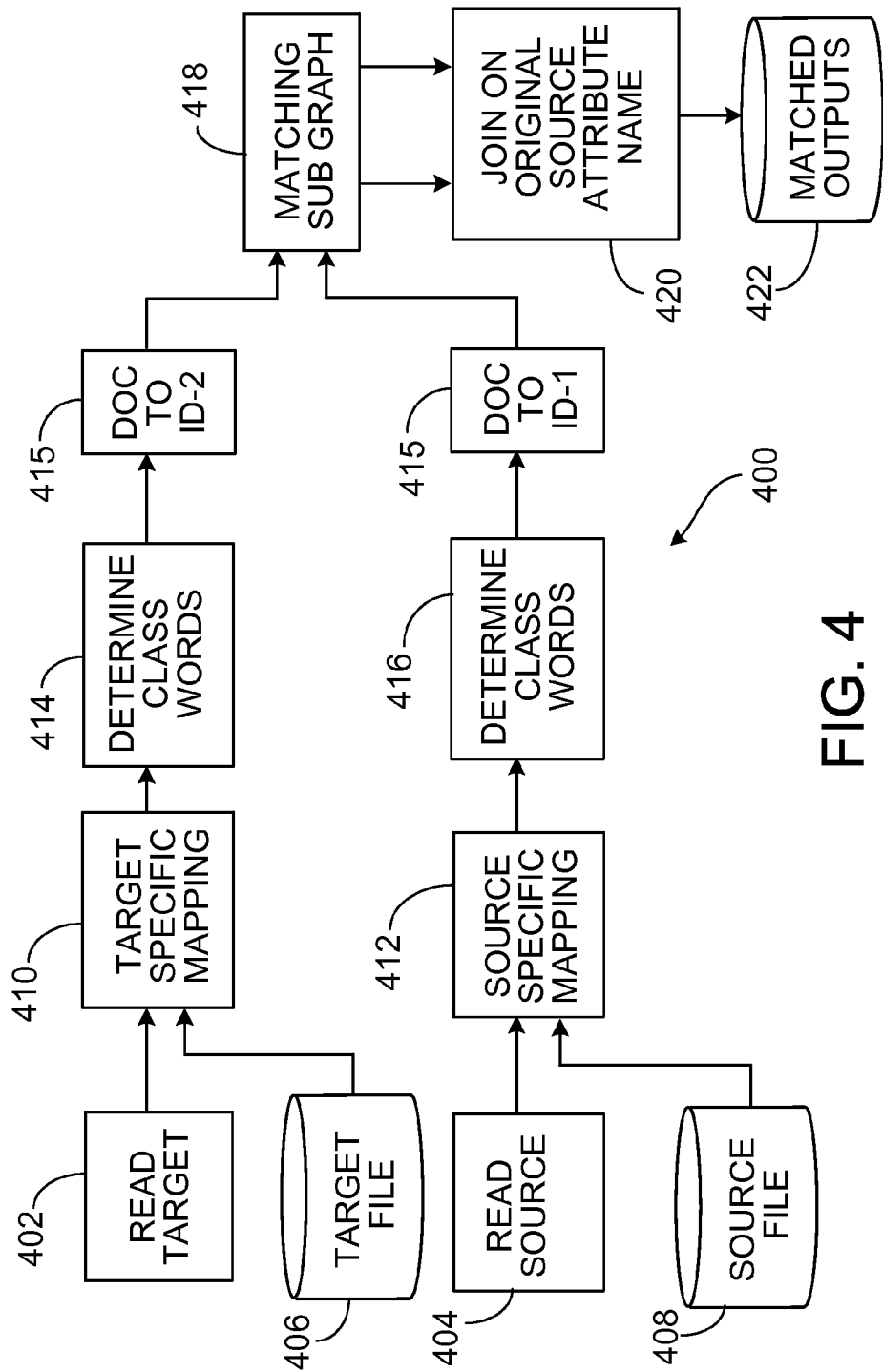


FIG. 4

5/13

500

VIEW DATA: MATCHED OUTPUTS									
FILE EDIT VIEW HELP									
MORE RECORDS: 10000									
TARGET	SOURCE	SOURCE_ATTRIBUTE_NAME	NM CANONICAL_ATTRIBUTE_NAME			MM SIMILARITY	506		
1 BEA 6	ATL	MILESTONE IDENTIFIER	MILESTONE NAME			8335110577...			
2 BEA 6	ATL	MILESTONE IDENTIFIER	MILESTONE IDENTIFIER } 504			1			
3 BEA 6	ATL	MILESTONE IDENTIFIER	MILESTONE DATE			8995557909...			
4 BEA 6	ATL	KEY PERFORMANCE PARAMETER URI	PROGRAM PERFORMANCE PARAMETER DIMENSION IDENTIFIER			4821268436...			
5 BEA 6	ATL	KEY PERFORMANCE PARAMETER URI	ACQUISITION PROGRAM BASELINE PERFORMANCE PARAMETER NAME			4165397139...			
6 BEA 6	ATL	KEY PERFORMANCE PARAMETER URI	ACQUISITION PROGRAM BASELINE PERFORMANCE PARAMETER CAT...			4083771164...			
7 BEA 6	ATL	KEY PERFORMANCE PARAMETER ACTUAL MEASUREMENT DATE	PROGRAM PERFORMANCE PARAMETER DIMENSION IDENTIFIER			4371043409...			
8 BEA 6	ATL	KEY PERFORMANCE PARAMETER ACTUAL MEASUREMENT DATE	ACQUISITION PROGRAM BASELINE PERFORMANCE PARAMETER NAME			4102323311...			
9 BEA 6	ATL	KEY PERFORMANCE PARAMETER ACTUAL MEASUREMENT DATE	ACQUISITION PROGRAM BASELINE PERFORMANCE PARAMETER CAT...			4021939225...			
10 BEA 6	ATL	KEY PERFORMANCE PARAMETER ACTUAL MEASUREMENT	PROGRAM PERFORMANCE PARAMETER DIMENSION IDENTIFIER			4406992243...			
11 BEA 6	ATL	KEY PERFORMANCE PARAMETER ACTUAL MEASUREMENT	ACQUISITION PROGRAM BASELINE PERFORMANCE PARAMETER NAME			4136068156...			
12 BEA 6	ATL	KEY PERFORMANCE PARAMETER ACTUAL MEASUREMENT	ACQUISITION PROGRAM BASELINE PERFORMANCE PARAMETER CAT...			4055016917...			
13 BEA 6	ATL	SYSTEM PHASE	PHASE TASK NAME			5160325894...			
14 BEA 6	ATL	SYSTEM PHASE	PHASE TASK IDENTIFIER			5300263797...			
15 BEA 6	ATL	SYSTEM PHASE	ACQUISITION PROGRAM PHASE NAME			5574949488...			
16 BEA 6	ATL	PROGRAM URI	PROGRAM RESOURCE IDENTIFIER			7979022796...			
17 BEA 6	ATL	PROGRAM URI	PROGRAM PLAN IDENTIFIER			4943804452...			
18 BEA 6	ATL	PROGRAM URI	PROGRAM IDENTIFIER			5642681677...			
19 BEA 6	ATL	CONTRACT EFFORT NAME	CONTRACT TYPE CODE			7167656693...			
20 BEA 6	ATL	CONTRACT EFFORT NAME	CONTRACT IDENTIFIER			7424746381...			
21 BEA 6	ATL	CONTRACT EFFORT NAME	ACQUISITION PROGRAM NAME			7411930088...			
22 BEA 6	ATL	ORIGINAL QUANTITY	SHIPMENT ORIGIN IDENTIFIER			6484262431...			
23 BEA 6	ATL	ORIGINAL QUANTITY	ORIGINAL CONSTRUCTION IDENTIFIER			5760806298...			
24 BEA 6	ATL	ORIGINAL QUANTITY	COLLECTION ORIGINATOR NAME			5012578877...			
25 BEA 6	ATL	CONTRACT EFFORT NAME	CONTRACT TYPE CODE			7167656693...			

FIG. 5

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

FIG. 5 (Cont.)

602											
604											
600											
RULE CASES											
TRIGGER	TRIGGER	TRIGGER	TRIGGER	TRIGGER	TRIGGER	TRIGGER	TRIGGER	TRIGGER	TRIGGER	TRIGGER	TRIGGER
NAME	DESCRIPTION	SAME	SAME	SAME	SAME	SAME	SAME	SAME	SAME	SAME	SAME
SIMILARITY	SIMILARITY	NAMES	NAMES	WORDS	WORDS	WORDS	WORDS	WORDS	WORDS	WORDS	WORDS
1 >.95	>.70	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
2 >.95	>.70			TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
3 >.95	>.55	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
4 >.80	>.95			TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
5 >.73	>.40			TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
6 >.70	>.25			TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
7 >.62		TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
8 >.62				TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
9 >.70											
10	>.70										
11 >.39	>.10			TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
12											

FIG. 6

MATCHING RULES RULESET - AB INITIO BUSINESS RULES ENVIRONMENT									
FILE EDIT VIEW RULESET TOOLS WINDOW HELP									
<div style="display: flex; justify-content: space-between;"> TEST MODE: FILE TEST DEPLOYMENT: DEFAULT DEPLOY 13 OF 43 </div>									
<div style="display: flex; justify-content: space-between;"> R: COMPUTE BEST MATCH FOR SOURCE ATTRIBUTE RESULTS </div>									
FILE TEST RESULTS									
RULESET - "MATCHING RULES"									
CANONICAL ATTRIBUTE NAME	CANONICAL ATTRIBUTE DESC	CANONICAL CLASS WORD	SIMILARITY	MATCH GRADE	SAME NAMES	SAME CLASS WORDS			
252 CONTRACT-IDENTIFIER	THE-PROCUREMENT-INSTRUM...	IDENTIFIER	1.000000...	AC	0	1	CON		
525 ACQUISITION-PROGRAM-NAME	THE-OFFICIAL-DESIGNATION-A...	NAME	1	AC	0	1	ACQ		
526 ACQUISITION-PROGRAM-NAME	THE-OFFICIAL-DESIGNATION-A...	NAME	1	AC	0	1	ACQ		
3977 COMPENSATION-POLICY-INDICATOR	A-CHARACTER-STRING-THAT-IN...	CODE	99999999...	AB	1	1	COM		
4473 DOCUMENT-IDENTIFIER	THE-DESIGNATOR-THAT-DISTIN...	IDENTIFIER	1	AB	1	1	DOC		
4474 DOCUMENT-IDENTIFIER	THE-DESIGNATOR-THAT-DISTIN...	IDENTIFIER	1	AB	1	1	DOC		
4475 DOCUMENT-IDENTIFIER	THE-DESIGNATOR-THAT-DISTIN...	IDENTIFIER	1	AB	1	1	DOC		
4500 DOCUMENT-STATUS-CODE	THE-CODE-THAT-REPRESENTS...	CODE	99999999...	AB	1	1	DOC		
9735 AUTHORITY-TYPE-CODE	THE-AUTHORITY-TYPE-CODE-ID...	CODE	1	AB	1	1	AUT		
9736 AUTHORITY-TYPE-CODE	THE-AUTHORITY-TYPE-CODE-ID...	CODE	1	AB	1	1	AUT		
9751 AVAILABILITY-TYPE-CODE	THE-AVAILABILITY-TYPE-VALUE...	CODE	1	AC	0	1	AVA		
9771 BEGIN-END-INDICATOR	THE-BEGIN-END-INDICATOR-ID...	CODE	99999999...	AB	1	1	BEG		
9787 BUDGET-ACTIVITY-IDENTIFIER	THE-BUDGET-ACTIVITY-IDENTIF...	IDENTIFIER	9985698...	AC	0	1	BUD		
9826 BUSINESS-EVENT-TYPE-CODE	THE-CODE-THAT-DESIGNATES...	CODE	1	AB	0	1	BUS		
9845 CONTINGENCY-CODE	THIS-DATA-ELEMENT-IS-INTEN...	CODE	1	AC	0	1	CON		
9856 COST-CENTER-IDENTIFIER	A-COST-CENTER-IDENTIFIER-CL...	IDENTIFIER	1.000000...	AB	1	1	COS		
9864 COST-ELEMENT-CODE	COST-ELEMENT-CODE-IS-A-CL...	CODE	1	AB	1	1	COS		
III									
ACTIVE RECORD INPUTS:									
NAME	VALUE								
INPUT									
SOURCE	SFIS								
TARGET	BEA 6								
SOURCE ATTRIBUTE NAME	BUDGET-ACTIVITY-IDENTIFIER								
SOURCE ATTRIBUTE DESC	BUDGET-ACTIVITY-REPRESENT-THAT-FIRST-LEVEL-OF-SUB-D...								
SOURCE CLASS WORD	IDENTIFIER								
ACTIVE RECORD OUTPUTS(VERT):									
NAME	VALUE								
SIMILARITY	99856982744099								
MATCH GRADE	AC								
SAME NAMES	0								
SAME CLASS WORDS	1								
CANONICAL ATTR WITH HIGHEST...	BUDGET-ACTIVITY-IDENTIFIER								
CANONICAL ATTR DESC WITH HIG...	THE-BUDGET-ACTIVITY-IDENTIFIER-REPRESENTS-THE-FIRST...								
✓ EME: BEA MATCHING EME : MAIN									

FIG. 7

9/13

800

MATCHING RULES RULESET - AB INITIO BUSINESS RULES ENVIRONMENT

FILE EDIT VIEW RULESET TOOLS WINDOW HELP

TEST MODE: FILE TEST | DEPLOYMENT: DEFAULT DEPLOY | 5 OF 500

HOME X RS: MATCHING RULES X R: COMPUTE BEST MATCH FOR SOURCE ATTRIBUTE X RESULTS X

RULESET - "MATCHING RULES"

SINGLE FIRE RULE - "COMPUTE BEST MATCH FOR SOURCE ATTRIBUTE"

ABOUT

CASES GRID (ONLY THE FIRST TRUE CASE WILL FIRE)

TRIGGERS				OUTPUTS				TIMES FIRED	
NAME SIMILARITY	DESCRIPTION SIMILARITY	SAME CLASS WORDS	CANONICAL ATTRIBUTE NAME	CANONICAL ATTRIBUTE DESC	MATCH GRADE	CANONICAL CLASS WORD	SIMILARITY		
1 >.95	>.70	TRUE	NM CANONICAL...	NM CANONICAL A...	"AA"	NM CANONIC...	NAME SIMI...	"NAMES AN..."	77
2 >.95	>.70	TRUE	NM CANONICAL...	NM CANONICAL A...	"AB"	NM CANONIC...	NAME SIMI...	"NAMES AN..."	4
3 >.95	>.55	TRUE	NM CANONICAL...	NM CANONICAL A...	"AB"	NM CANONIC...	NAME SIMI...	"NAMES AN..."	25
4 >.80	>.95	TRUE	DS CANONICAL...	DS CANONICAL AT...	"AC"	DS CANONICA...	DESSCRIPTIO...	"NAMES AN..."	14
5 >.73	>.40	TRUE	CANONICAL ATTR...	CANONICAL ATTR...	"BA"	CANONICAL CL...	HIGHEST SI...	"LESS GOOD"	303
6 >.70	>.25	TRUE	CANONICAL ATTR...	CANONICAL ATTR...	"CA"	CANONICAL CL...	HIGHEST SI...	"LESS GOOD"	259
7 >.62	ANY	TRUE	NM CANONICAL...	NM CANONICAL A...	"BA"	CANONICAL CL...	HIGHEST SI...	"LESS GOOD"	9
8 >.62	ANY	TRUE	CANONICAL ATTR...	CANONICAL ATTR...	"OB"	CANONICAL CL...	HIGHEST SI...	"LESS GOOD"	322
9 >.70	ANY	ANY	CANONICAL ATTR...	CANONICAL ATTR...	"CC"	CANONICAL CL...	HIGHEST SI...	"LESS GOOD"	517
10 ANY	>.70	ANY	CANONICAL ATTR...	CANONICAL ATTR...	"CC"	CANONICAL CL...	HIGHEST SI...	"LESS GOOD"	211
11 >.39	>.10	TRUE	CANONICAL ATTR...	CANONICAL ATTR...	"CD"	CANONICAL CL...	HIGHEST SI...	"LESS GOOD"	1355
12 ANY	ANY	ANY	CANONICAL ATTR...	CANONICAL ATTR...	"F"	CANONICAL CL...	HIGHEST SI...	"NO MATCH"	7220
*	802								

ACTIVE RECORD INPUTS: X ACTIVE RECORD OUTPUTS(VERT):

✓ EME: BEA MATCHING EME: MAIN

FIG. 8

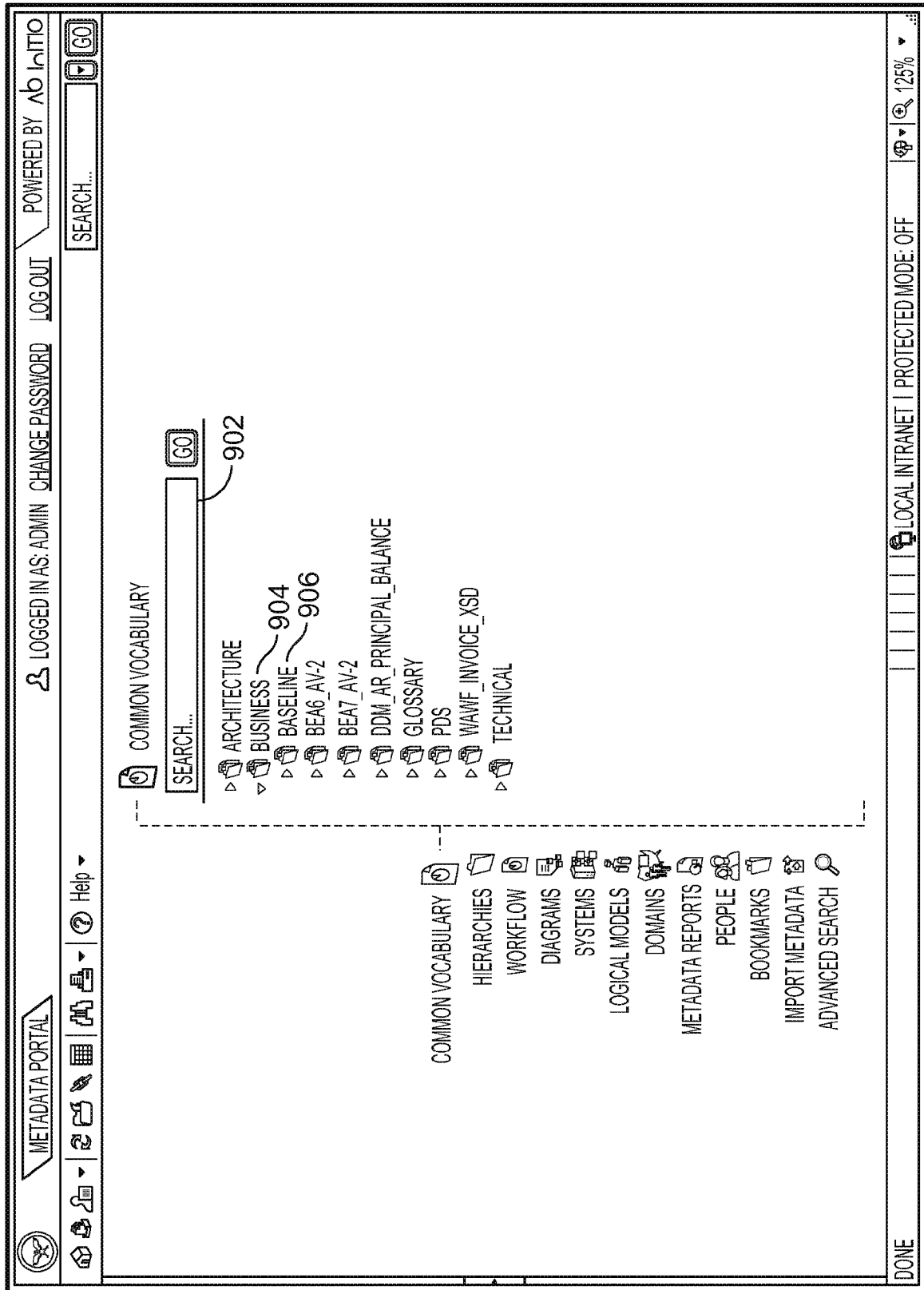


FIG. 9

900

LOGGED IN AS: ADMIN

CHANGE PASSWORD

LOG OUT

POWERED BY

DT

METADATA PORTAL

DETAILS FOR BUSINESS TERM ACCEPTANCE DATE 1002

INFO

ACRONYMS

CLASSIFICATIONS

MATCHES

APPROVAL WORKFLOW

NOTES

VERSIONS

EDIT

DELETE

GROUP

DESCRIPTION

APPROVAL STATUS

STEWARD

NAME

DOMAIN

TYPE

LENGTH

SEMANTIC RELATIONSHIPS

OWL RDF

OBJECT OWNER

BASELINE

THE CALENDAR DATE ON WHICH AN ACCEPTANCE OCCURS AFTER RECEIPT INSPECTION OF A QUANTITY OF UNITS OF MEASURE OF A CONTRACTED LINE ITEM

MATCHED TO BASELINE

ACCEPTANCE DATE

STRING

12

METADATA PORTAL

DONE

FIG. 10

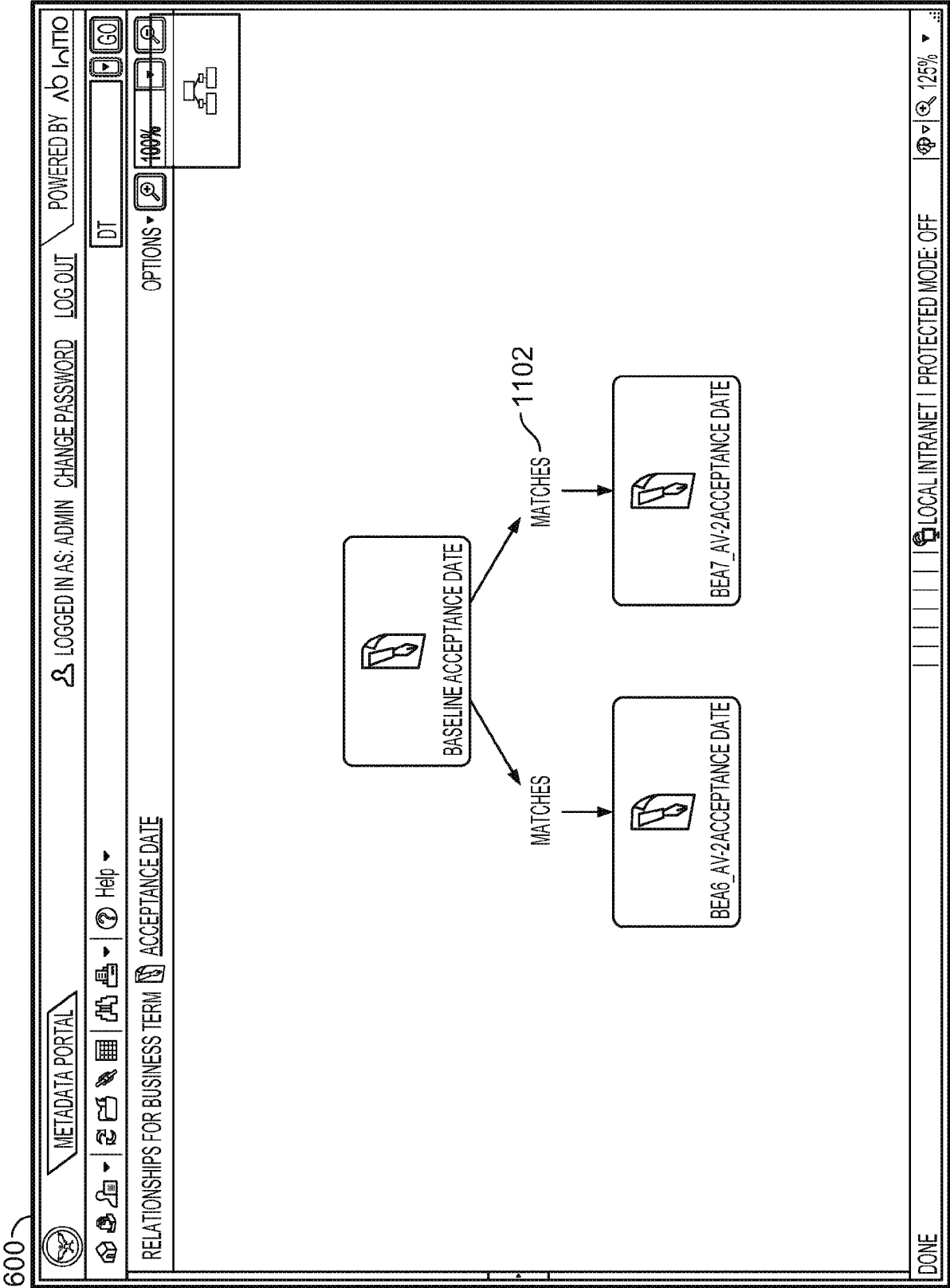


FIG. 11

600

DETAILS FOR BIZTERM: ACCEPTANCE DATE

RELATIONSHIPS

GO

INFO

ACRONYMS

CLASSIFICATIONS

MATCHES

APPROVAL WORKFLOW

NOTES

VERSIONS

1202

FROM GROUP

FROM TERM

RELATIONSHIPS

TO GROUP

TO TERM

1	<div>BASELINE</div>	<div>ACCEPTANCE DATE</div>	<div>MATCHES</div>	<div>BEA6 AV-2</div>	<div>ACCEPTANCE DATE</div>
2	<div>BASELINE</div>	<div>ACCEPTANCE DATE</div>	<div>MATCHES</div>	<div>BEA7 AV-2</div>	<div>ACCEPTANCE DATE</div>

FIG. 12

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/021108

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30
ADD.

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)

EP0-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2006/116286 A2 (FALIC LEON [US]) 2 November 2006 (2006-11-02) paragraphs [0032] - [0053]; figures 1-17 -----	1-24
A	US 2005/065921 A1 (HRLE NAMIK [DE] ET AL) 24 March 2005 (2005-03-24) the whole document -----	1-24
A	US 2004/249682 A1 (DEMARCKEN CARL G [US] ET AL) 9 December 2004 (2004-12-09) the whole document -----	1-24

☐

Further documents are listed in the continuation of Box C.

☒

See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier 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

24 March 2011

Date of mailing of the international search report

01/04/2011

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Moon, Timothy

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2011/021108

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2006116286 A2	02-11-2006	CA 2609911 A1 EP 1877973 A2	02-11-2006 16-01-2008
US 2005065921 A1	24-03-2005	NONE	
US 2004249682 A1	09-12-2004	EP 1634200 A1 WO 2005001717 A1	15-03-2006 06-01-2005