



- (51) International Patent Classification:
G06F 17/30 (2006.01) *G06F 17/21* (2006.01)
- (21) International Application Number:
PCT/US2014/012028
- (22) International Filing Date:
17 January 2014 (17.01.2014)
- (25) Filing Language: English
- (26) Publication Language: English
- (71) Applicant: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).
- (72) Inventors: **ZANGARO, Pablo Sebastian**; 165 Dascomb Road, Andover, Massachusetts 01810-5897 (US). **DE SOUZA, Joaquim Gomes Da Costa Eulalio**; Avenida Ipiranga 6681, Building 99A, Rio Grande do Sul, 90619-900 Porto Alegre (BR). **SOMBRIO, Evandro**; Avenida Ipiranga 6681, Building 99A, Rio Grande do Sul, 90619-900 Porto Alegre (BR).
- (74) Agents: **ORTEGA, Arthur S.** et al.; Hewlett-Packard Company, Intellectual Property Administration, Mail Stop 35, 3404 E. Harmony Road, Fort Collins, Colorado 80528 (US).

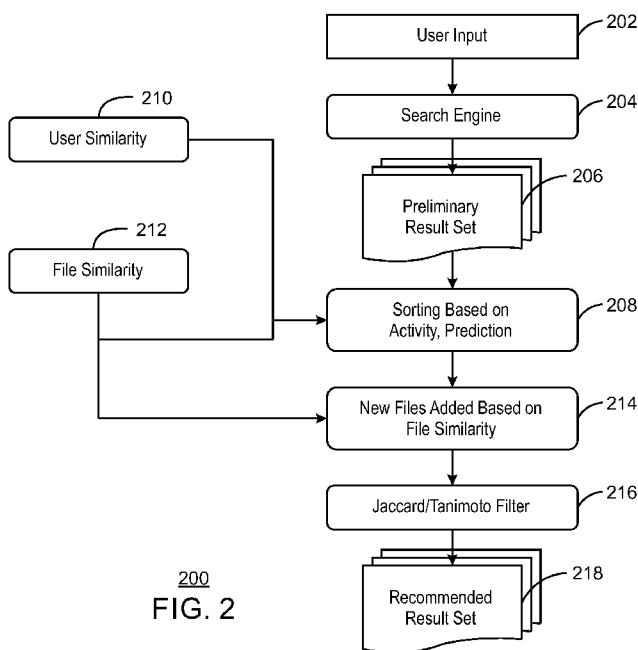
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

[Continued on next page]

(54) Title: FILE LOCATOR



200
FIG. 2

(57) Abstract: Locating files in a storage system includes calculating a similarity between a number of users, and calculating a similarity between a number of files. A keyword is obtained, and a list of files is recommended to a user based, at least in part, on the similarity between the user and other users, the similarity between the files and other files, and the keyword.



Published:

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

FILE LOCATOR

BACKGROUND

[0001] The size of file based storage system continues to grow exponentially, driven, in part, by the need for content sharing in organizations, compliance requirements from government regulations, and entertainment content all expand. Some commercial systems can store 16 petabytes of data, or more. When considering single namespaces at this petabyte scale, the traditional methods for searching specific objects by name, path, or extension may not be adequate. A search for a specific file name in a large file system could take several hours or even days. Further, the user may need to know precise information about the file being searched, since file name searches return the results of the query sorted by date order, and not by relevance or importance.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Certain exemplary embodiments are described in the following detailed description and in reference to the drawings, in which:

[0003] Fig. 1 is an example block diagram of a large scale network that includes multiple storage area networks (SANs) and multiple users;

[0004] Fig. 2 is an example process flow diagram of a method for a combined keyword and similarity search;

[0005] Fig. 3 is an example process flow diagram of another method for a combined similarity and keyword search; and

[0006] Fig. 4 is an example block diagram of a tangible, computer readable medium that includes code configured to direct a processor to execute combined searches.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

[0007] The techniques described herein disclose the use of combined search systems that use both recommender engines and search engines. As used herein, a recommender system compares user data and user file access to determine what files similarly situated users are opening. The analysis of similarities for both users and files may be implemented using Pearson Correlation Scores, among others. The numerical scores for the correlation can be used to predict the relevancy of

content, and make recommendations accordingly. Once recommendations are made, a keyword search can be run for the relevant group of files to identify the most likely files of use.

[0008] Fig. 1 is an example block diagram of a large scale network 100 that includes multiple storage area networks (SANs) 102 and multiple users 104. The users 104 may be members of different groups, for example, a first group 106 may be users in a corporate HR organization, a second group 108 may be users in a corporate sales organization, a third group 110 may be users in an executive functions for the corporation, other groups 112 may be users in engineering, R&D, distribution functions, or customers, among others. In another example, the network can be part of server farm.

[0009] The users 104 may be coupled to a corporate storage, such as the SANs 102, through a network 114. The network 114 may be a private corporate network, a public network, such as the Internet, or a combined public and private network, such as a corporate network in which multiple sites are connected through virtual private networks over a public network. Different users 104 may have different needs for files, as well as different levels of access. The SANs 102 may hold millions of files, or more, for example, including pay records, compensation guidelines, new products, and plant production records, among many others.

[0010] As an example, a user 116 in a particular group 110 may wish to locate a file 118 in a corporate storage, such as the SANs 102. However, as the SANs may contain very large numbers of files, any normal search function may be very time consuming. In addition to the long search times, users are generally unaware of other documents that could be relevant to their work, based on the activity of other similar users. For example, other users 120 from a single group, such as group 110, accessing the same files 122 may indicate that the content of those files 122 and files 124 that are similar in type and content have a high probability of being relevant to the user 116 searching for the file 118.

[0011] In this example, a storage manager 126 may be used to implement a method to make it easier to find the desired file 118. The storage manager 126 may include a similarity engine 128 that calculates a similarity between users, such as members of a single group 110, and files 122 and 124. The results of the similarity calculation can be stored for future use in a similarity database 130. The storage manager 126 may also include a metadata database 132 which stores keywords and

metadata concerning files in the SANs 102. The metadata database 132 may include the similarity database 130. As used herein, the metadata may include file types, such as image files, documents, and the like, as well as sizes, storage formats, document owners, document groups, authors, and the like. The metadata database 132 may also store custom metadata, such as a series of key-value properties arbitrarily assigned by the users to each document.

[0012] For files that have text, or text descriptions, key words can also be generated for files, for example, using text processing techniques to identify the most common words, concepts, and the like. A search engine 134 can be used to search for metadata or a keyword in the metadata database 132, or subsets thereof, for example, to identify files that have the highest correlation with that word or concept. A recommender engine 136 can be used to identify files that have been accessed by users that are similarly situated, such as files 122 or files 124 having similar type and content. In this simplified example seven files 122 and 124 would be reported. However, in actual situations, the number of files will likely be much higher, for example, several hundred or several thousand.

[0013] In this example, the recommendations provided by the recommender engine 136 can be further narrow by making use of the search engine 134 to search the metadata database 132 for file system metadata, file contents, or both. In this example, the recommender engine 136 combines regular search algorithms with recommender systems to provide better and more accurate results. Thus, searching for specific content in a file system with billions of files can include a double search or a search within a search. For example, a first search with keywords or other metadata, and a second search, based on similarity calculations for meaningful results inside the search results. The application of recommender systems to these search results will help users reduce the time to filter out noise in the keyword search results.

[0014] Fig. 2 is an example process flow diagram of a method 200 for a combined keyword and similarity search. The method begins at block 202 with a user entering search conditions. These conditions may include system metadata, custom metadata, and keywords related to file contents. For example, in a medical PACS (Picture Archiving and Communication System) objects may be tagged with metadata like Patient Name, Patient Sex, Study Date, and Type of Study. At block 204, the search is run, for example, on a metadata database, using these metadata

and others like file size, last accessed time, keywords, and others. Although the search could be run on the file system, as discussed above, this may lead to long search times. The metadata database indexes the metadata of all objects in the file system, and may perform searches 100,000 times faster than a regular find command on the file system.

[0015] Once a preliminary result set 206 is returned by the search engine, at block 208, a recommender engine may apply a file-based nearest neighbor recommendation to sort the preliminary results set. This is based on the activity of similar users on the file systems, which may be determined at block 210, prior to the search.

[0016] The similarity calculation represented in block 210 may use the mapping detailed in Table 1. When the recommender system starts, it “learns” about the behavior of users in the file system assigning a value for each type of activity a user performs on a file. This information is maintained in a table, for example, in the metadata database or a similarity database, and can be used for the file based recommendations.

[0017] Table 1: Mapping between file system activities and scores

File based storage systems	
Open file	3
Display file properties	2
No activity on the file	1

[0018] The recommender system first processes the results returned by the search engine by sorting the results by an interest defined, for example, by previous activity. The system may assume that the user will be interested in files he has already worked with, so they will be the first in the results list. The files browsed by the user, e.g., when a user selects a file and display the file properties, may be appended to the results list. The recommender system takes the rest of the result set and tries to predict how the user will rank those results. The prediction can be based on file-based nearest neighbor recommendation, e.g., if there are files similar to the ones the user already interacted with, those will be ranked high in the result set. The similarity of files is calculated at block 212, and may be performed prior to the search.

[0019] The similarity of files in the recommender system may be calculated using the adjusted cosine similarity algorithm, among others. For example, the similarity (sim) between two objects x and y is defined as: $\text{sim}(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| * |\vec{y}|}$, with “.” being the dot product of vectors, and $|\vec{x}|$ and $|\vec{y}|$ representing the Euclidian length of the vectors. The values are extracted from the usage matrix stored in the metadata database with the file system activity, as shown in Table 2. Each vector is defined as the set of activities for each object. For example the vector x for Object 1 is defined in this case as (3, 2, 2, 1, 1).

[0020] Table 2: User-Object activity matrix

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6
User A	3	2	1	1	2	?
User B	2	3	3	3	3	3
User C	2	1	2	1	1	1
User D	1	1	1	1	1	1
User E	1	1	1	2	2	3

[0021] This table is adjusted by subtracting the average value for each user in the Matrix. For example, the average value for User A for Objects 1, 2, 3, 4, 5 is 1.8, so the line for user A will be as shown in Table 3.

[0022] Table 3: Average value for objects for User A

	Object 1	Object 2	Object 3	Object 4	Object 5	Object 6
User A	1.2	0.2	-0.8	-0.8	0.2	?

[0023] In this example, the sim between Object 6 and Object 4 is 0.89. The sim values range between -1 and 1, with 1 denoting perfect match or strong similarity. After the similarities are calculated, the recommender system predicts the action User A will take on Object 6. A prediction close to 3 means the User will probably open the file, 2 means the user will inspect the file metadata, and 1 means the User will probably ignore the file. This prediction will be used to enhance the search results by categorizing the results based on this probability. To reduce the amount of resources used to calculate similarities between huge numbers of objects, the

matrix will take only into account activity from a reduced set of users and find the users with similar activity in the system. The calculation of this reduced set of users may be used for collaborative recommendations.

[0024] At block 214, the sorted list of results may be enhanced by the addition of a list of similar files collected during this step, for example, using the file similarity results from block 212 and user similarity calculated at block 210. The list will contain files based on similarity by activity, and present the users with files that other users in the group are using, which the user might be interested as well. This will produce a list of files recently used by similar user.

[0025] For the collaborative recommendations made in block 214, e.g., recommendations based on activity from other users, user similarity is calculated at block 210 only between users belonging to the same user group, for example, as defined in groups in POSIX file systems or Windows Active Directory groups. To find the correlation score between two users, the system calculates the Pearson's coefficient (r), which shows the degree of relationship between two variables, as shown in Table 4. The Pearson's coefficient will have a value between -1 and 1, wherein each extreme represents a perfect positive or negative correlation, respectively. In the present example, the two variables may represent the activity scores for the two users. The more similar activity the users have on the same files, e.g., they open the same files, display properties on the same files, and ignore the same files, the higher the Pearson's coefficient. A value for the Pearson's coefficient of +1 implies that the users performed identical activities in the file system. A value of 0 denotes the absence of linear relationship. Negative values for the Pearson's coefficient scores means that most files opened by one user are ignored by the second user.

[0026] At block 210, the recommender system may calculate the coefficient r , e.g., the Pearson's coefficient, between all members of a group. After these coefficients are calculated, it selects the top N users with higher coefficient r . After constructing the list of similar users, the system then selects the files that those users have recently opened and at block 214 presents them to the user as files added to the list of recommendations. For example, if a number of users from the Human Resources department are reading a document about an important change in the employees' benefits, this document will be displayed as added to the

preliminary results set for a similar user in the department that has not read the file yet.

[0027] Table 4: Scores and Pearson's coefficient for Users A and B

	User A scores	User B scores
File A	3	3
File B	3	3
File C	3	2
File D	2	1
File E	3	3
File F	3	3
File G	2	1
File H	2	2
File I	1	1
File J	1	1
File K	1	1
File L	1	1
File M	2	3
File N	3	3
File O	2	1
File P	1	3
File Q	3	3
File R	3	3
Correlation coefficient (r)	0.688333	

[0028] Many organizations implement security policies where files can be only viewed by specific people or groups. For these cases the recommender system may implement security options to reflect the organization policies. Even though recommendations are anonymous, e.g., the system recommends files used by similar users but does not display any user information, users can opt out of the system. In this example, the activity of the user is not tracked and other users participating in the system will not receive any recommendation from them. Individual files or group of files can also be excluded from the system.

[0029] The list of recommended files generated at block 214 may include files recently opened by some users in the group that are not related to the one selected by the user, e.g., some users from the group may be listening to an mp3 file stored in the storage system. To correct this, at block 216, the recommender system

calculates the similarity of the documents, for example, using a Jaccard/Tanimoto Index, among others.

[0030] The Jaccard/Tanimoto Index is used in statistics to compare similarity and diversity of sample sets. The sample sets are built for each document using the custom metadata assigned to the files. Given two files A and B, and their sets of custom metadata S_a and S_b respectively, the Jaccard/Tanimoto index is calculated with the following equation: $T = \frac{|S_a \cap S_b|}{|S_a| + |S_b| - |S_a \cap S_b|}$, with $|S_a|$ = number of elements in custom metadata set S_a , $|S_b|$ the number of elements in custom metadata set S_b and $|S_a \cap S_b|$ the number of elements in the intersecting set (the custom metadata they have in common). The closer the index T to 1, the more similar both files are in terms of custom metadata. A T index of 0 denotes no proximity between the sets.

[0031] For filtering the list, identifying a first file that matches the search using custom metadata is the first step in the algorithm. Since this file will contain all the metadata that the user is looking for in the search, the algorithm will use it for filtering the collaborative recommendations. T is calculated twice for each pair of files, wherein a first file is the top file in the preliminary result search and the second is each file from the list produced by the Pearson Coefficient. One calculation of T is for the Key part of the key-value pair of the custom metadata and the second calculation is for the double tuple key-value.

[0032] The T index calculation for the key-value pair part will give the provenance index (T_p). The provenance index determines whether the files have similar keys in their key-value set, which indicates that the files were created by similar applications. For example a medical application for storing medical images will create a set of metadata keys for each file with keys like Patient Name, Patient Sex, Study Date, and Performing Physicians Name.

[0033] The T index calculation for the T key-value tuple will give the similarity index, (T_s). For example, a file A with custom metadata "PatientName=JohnDoe, PatientSex=M, StudyDate=1/1/2013" and a file B with custom metadata "PatientName=JohnDoe, PatientSex=M, StudyDate=01/02/2013" will have a $T_p = 1$ and a $T_s = 0.5$. These indexes indicate that the file have the same custom metadata keys but differ in the values. As the example also shows, more custom metadata for a file will result in a more accurate T index. In the semantics of the present example, this can be interpreted as "users who viewed file A also viewed...." Since file-based

systems can store billions of files, the recommender system limits the files used for correlation with the original file using only the files with recent activity among the users in the same group. Accordingly, the user, or an administrator, may select a threshold value for the correlation coefficient, below which files are omitted from the list of recommendations.

[0034] At block 218, the list of recommended files is presented to the user. The list may include hypertext links that allow the user to directly access the metadata for each file, or the file contents, by selecting options in the list.

[0035] The method 200 is not limited to the order or techniques described above. Any number of other correlation and filtering functions may be substituted for the Pearson's coefficient, the Jaccard/Tanimoto filter, or both. Further, any number of search engines can be used to build the preliminary result set.

[0036] Fig. 3 is an example process flow diagram of another method 300 for a combined similarity and keyword search. Like numbered items are as discussed with respect to Fig. 2. This is similar to the method 200 discussed with respect to Fig. 2, although the recommender search is run first in the example discussed with respect to Fig. 3. The method 300 starts at block 302 with the user entering a keyword and starting the search. At block 304, the recommender engine identifies files based on similarities between users and files, for example, as calculated in blocks 210 and 212. The identification may use the same techniques as described with respect to block 208 of Fig. 2. However, in block 304, all files identified as similar are entered into a list. At block 306, further files may be added to the list based, at least in part, on file similarity. This may be performed as described with respect to block 214 of Fig. 2. As noted, the files that are added may not be directly related to the search. Thus, at block 308, a filtering function, such as the Jaccard/Tanimoto filter discussed with respect to block 216, may be used to remove files that have a similarity below a selected limit.

[0037] At block 310, the user may provide keywords, if they were not provided at block 302. For example, at block 302 the user may open a search screen and be presented with a list of recommendations based on similar users and the similarity with other files. At that time, the user may be presented with an option to enter metadata or keywords to narrow the list of files further. At block 314, a search engine may then run a metadata or keywords search on the preliminary results set in

a metadata database, as described with respect to block 204. The final results set 316 may then be reported to the user.

[0038] The method 300 is not limited to the blocks shown in Fig. 3, but may include any number of other techniques. For example, the search may be iterative. In this example, after keywords are entered process flow may return from block 314 to block 304 to run through the similarity steps prior to presenting the recommended results set.

[0039] Fig. 4 is an example block diagram of a tangible, computer readable medium 400 that includes code configured to direct a processor 402 to execute combined searches. The tangible computer readable medium 400 may be a hard drive, an optical drive, a solid state drive, a thumb drive, a RAM drive, or any number of other tangible storage devices. The tangible computer readable medium 400 may be accessed by the processor 402 over a bus 404.

[0040] The tangible computer readable medium 400 may include a metadata database 406 that stores data about files in a data store. The data may include metadata, keywords, and results of similarity calculations as described above. A recommender engine 408 may be included to identify files for recommendations to users based, at least in part, on similarities between users and between files. A user similarity calculator 410 may be included to calculate the similarity between users, e.g., the likelihood that a user in a similar group or position would select similar files. A file similarity calculator 412 may be included to perform the same calculation for files. A filtering function, such as a Jaccard/Tanimoto filter, may be included in either or both of the similarity calculators 410 and 412. The tangible computer readable medium 400 may also include a search engine 414 that can search the metadata database 406 for metadata, keywords, or both. The search engine 414 may include the capability of directly searching the file structure, although it may be rarely used. The search engine 414 may also include language processing capabilities to allow the search engine 414 to locate documents that have similar words to the user entered metadata or keywords, allowing the search to progress even when words are synonyms, misspelled, and the like.

[0041] The code blocks are not limited to those shown, but may be organized in any number of ways while retaining the same functionality. For example, the filtering function may be set up as a separate module. Similarly, a separate coordination

module may be included to operate the functions. The coordination function, however, may be a part of the recommender engine 408.

[0042] While the present techniques may be susceptible to various modifications and alternative forms, the exemplary examples discussed above have been shown only by way of example. It is to be understood that the technique is not intended to be limited to the particular examples disclosed herein. Indeed, the present techniques include all alternatives, modifications, and equivalents falling within the true spirit and scope of the appended claims.

CLAIMS

What is claimed is:

1. A method for locating files in a storage system, comprising:
calculating a similarity between a plurality of users;
calculating a similarity between a plurality of files;
obtaining a keyword; and
recommending a list of files to the user based, at least in part, on the similarity between the user and other users, the similarity between the files and other files, and the keyword.
2. The method of claim 1, comprising creating a preliminary results set by:
identifying a subset of the plurality of files based, at least in part, on activity of similar users;
adding files to the subset based, at least in part, on similarity to other files; and
performing a filtering function of the files in the subset to remove at least a portion of the files, creating the preliminary results set.
3. The method of claim 2, comprising the preliminary results set before a user conducts a search.
4. The method of claim 2, comprising performing a search in the preliminary results set to eliminate a portion of the results base, at least in part, on the keyword.
5. The method of claim 1, comprising identifying a subset of the plurality of files as a preliminary results set based, at least in part, on a keyword entered by a user.
6. The method of claim 5, comprising
changing the number of files in the subset based, at least in part, on activity of similar users;
adding files to the subset based, at least in part, on similarity to other files; and

performing a filtering function of the files in the subset to remove at least a portion of the files, creating the preliminary results set.

7. The method of claim 1, comprising searching for the keyword in a database, wherein the database comprises keywords generated from a plurality of files.

8. The method of claim 1, comprising obtaining the keyword from metadata for a file.

9. The method of claim 1, comprising obtaining the keyword from a uniform resource locator (URL) string.

10. A file locator system, comprising:
a processor; and
a storage system, wherein the storage system comprises code configured to direct the processor to:
determine a similarity between a plurality of users;
determine a similarity between a plurality of files;
obtain a keyword;
identify a subset of plurality of files based, at least in part, on a similarity between the user and other users;
identify a subset of the plurality of files based, at least in part, on the keyword; and
present a listing of the subset of the plurality of files.

11. The file locator system of claim 10, comprising a storage attached network (SAN) device.

12. The file locator system of claim 10, comprising a storage manager configured to recommend files in a large scale storage system.

13. The file locator system of claim 10, comprising a server farm.

14. A tangible, computer readable medium comprising instructions that, when executed by a processor, direct the processor to:

- calculate a similarity between a plurality of users;
- calculate a similarity between a plurality of files;
- obtain a keyword; and
- recommend a list of files to the user based, at least in part, on the similarity between a user and other users, the similarity between a file and other files, and the keyword.

15. The tangible, computer readable medium of claim 14, comprising instructions to direct a processor to add files to a keyword search based, at least in part, on a similarity between a user and other users, a similarity between files and other files, or both.

1/4

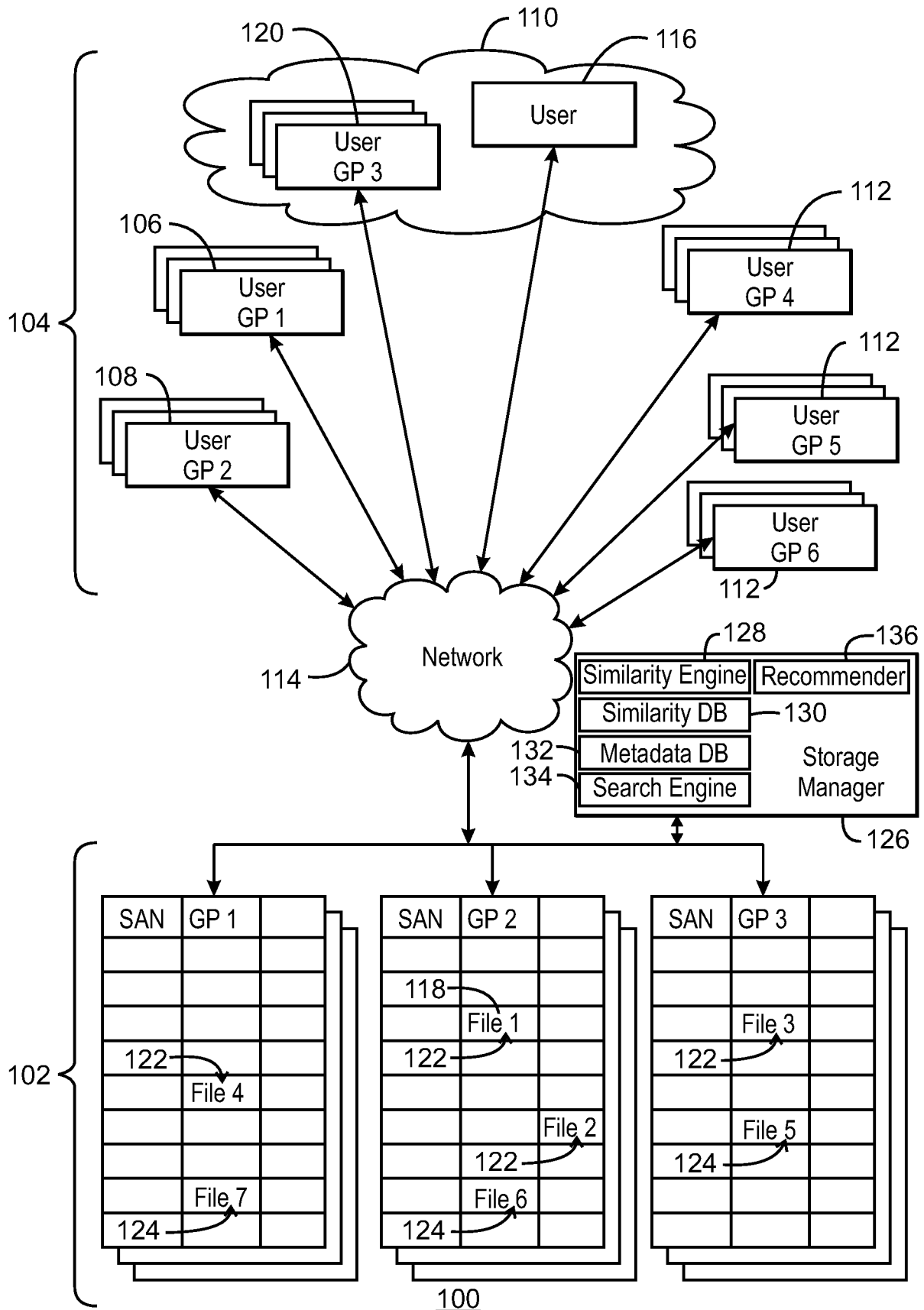
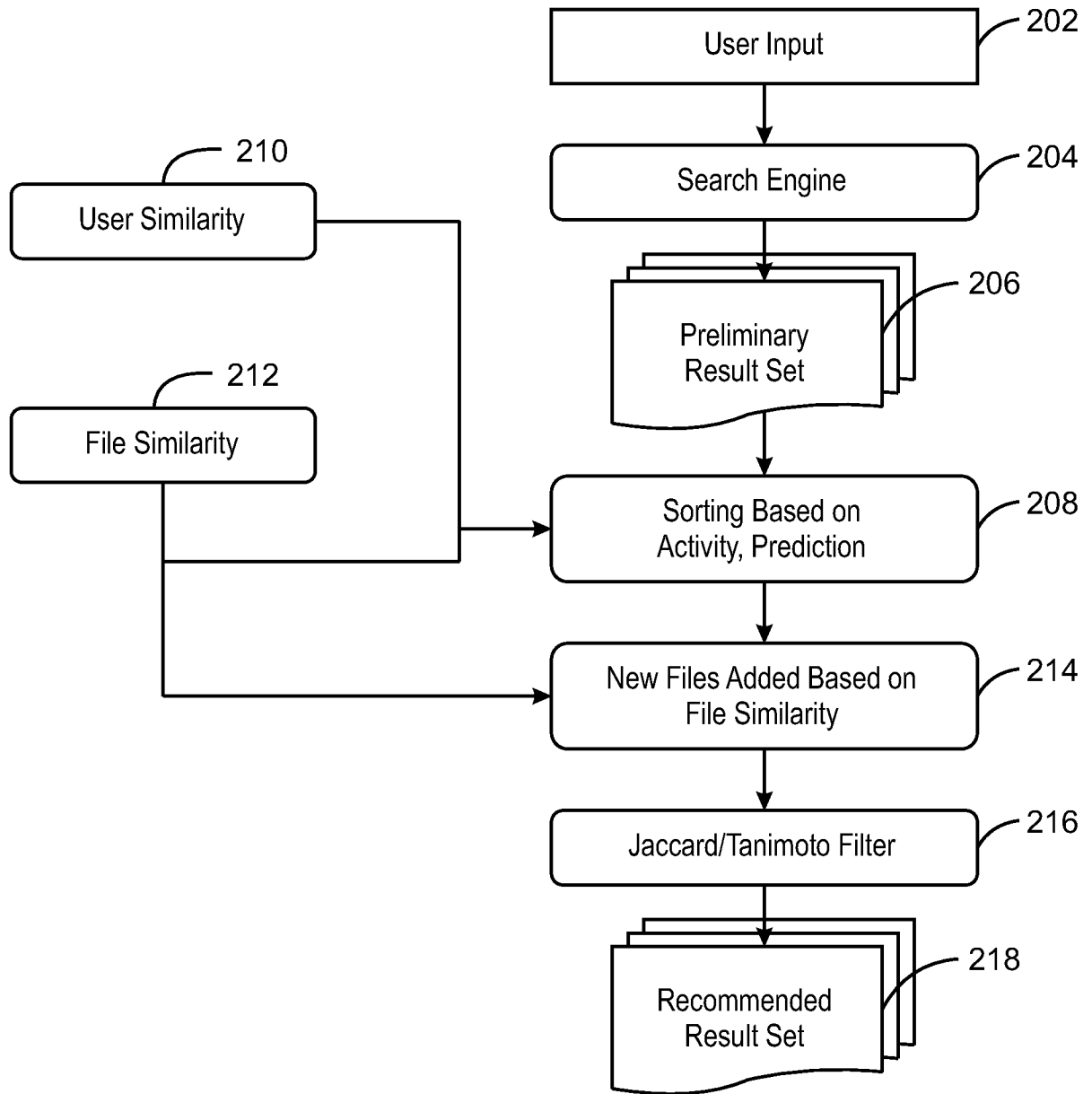
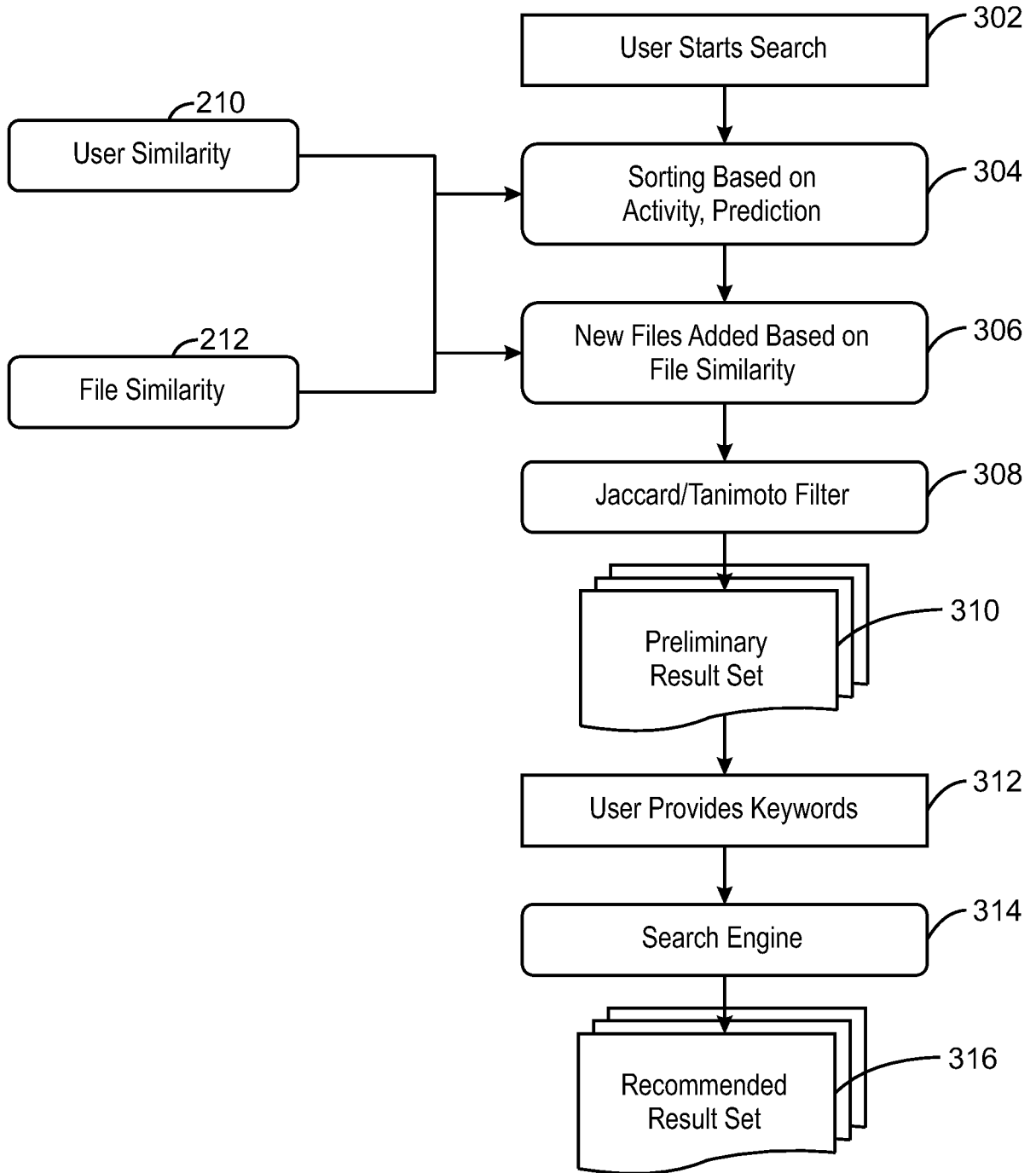


FIG. 1



200
FIG. 2



300
FIG. 3

4/4

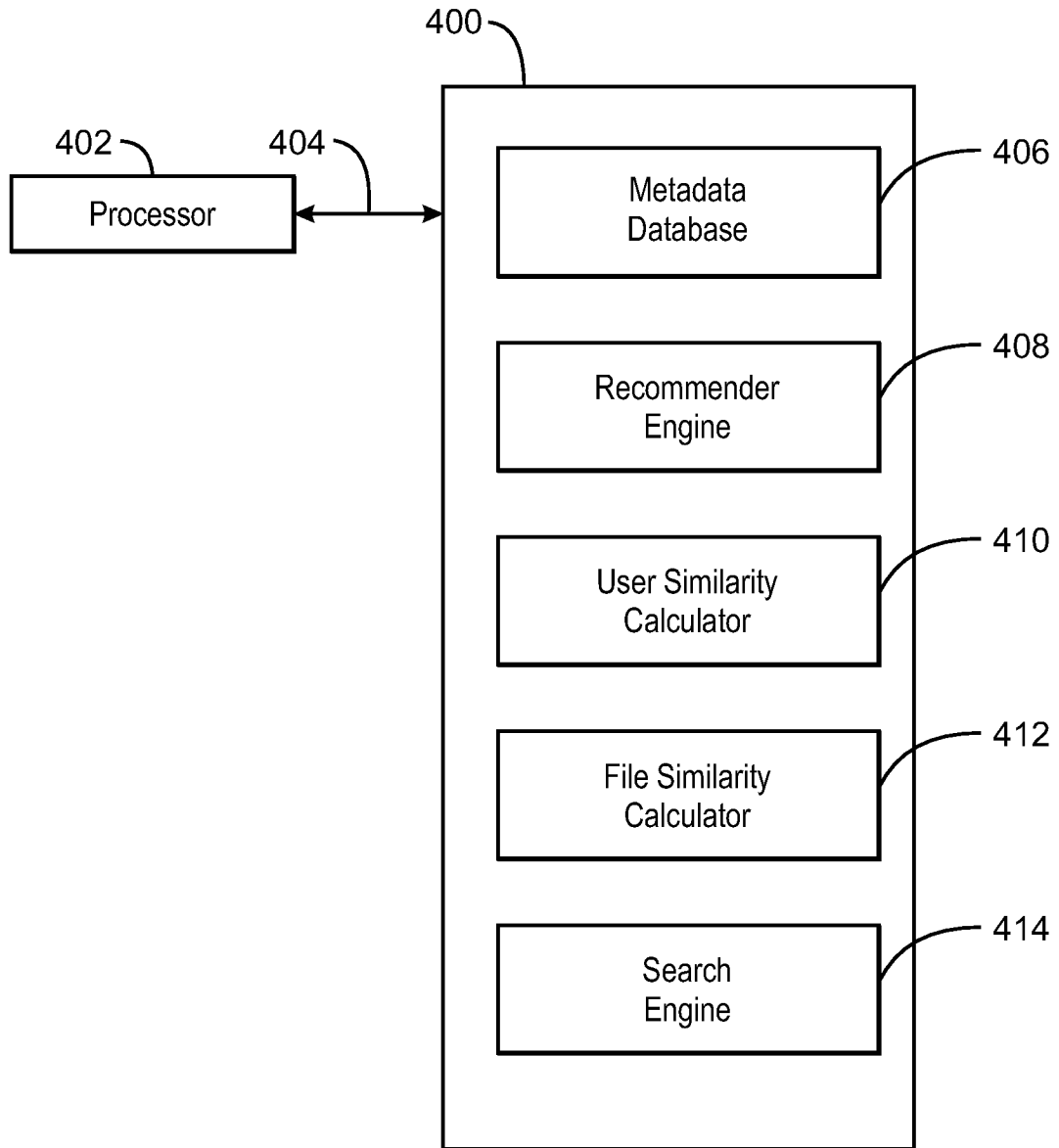


FIG. 4

A. CLASSIFICATION OF SUBJECT MATTER**G06F 17/30(2006.01)i, G06F 17/21(2006.01)i**

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 17/30; G06F 7/00; G06F 17/21

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: locate, similarity, user, file, keyword, recommend

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7930301 B2 (CEZARY MARCJAN et al.) 19 April 2011 See column 1, lines 28-32; column 2, lines 55-59; column 4, lines 50-64; column 5, lines 21-45; column 6, lines 17-60; claim 6; and figure 5.	1-6, 10, 12, 14-15
Y		7-9, 11, 13
Y	US 2005-0154723 A1 (PING LIANG) 14 July 2005 See paragraphs [0015], [0063], [0150]; and figures 2-4.	7-9, 11, 13
A	EP 1826692 A2 (COPERNIC TECHNOLOGIES, INC.) 29 August 2007 See paragraphs [0005]-[0007]; and figure 1.	1-15
A	US 2009-0287650 A1 (SUN HWA CHA) 19 November 2009 See paragraphs [0005]-[0006]; and figure 2.	1-15
A	US 2001-0020238 A1 (HIROSHI TSUDA) 06 September 2001 See paragraphs [0030]-[0032], [0036]-[0039]; and figures 2, 7.	1-15

 Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:

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

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

30 October 2014 (30.10.2014)

Date of mailing of the international search report

30 October 2014 (30.10.2014)

Name and mailing address of the ISA/KR

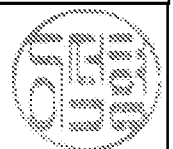
International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

LEE, Seok Hyung

Telephone No. +82-42-481-5983



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/012028

Patent document cited in search report	Publication date	Patent family member(s)	Publication date		
US 7930301 B2	19/04/2011	AU 2004-201344 A1	21/10/2004		
		AU 2004-201344 B2	17/12/2009		
		BR PI0401092 A	11/01/2005		
		CA 2462415 A1	30/09/2004		
		CA 2462415 C	04/02/2014		
		CN 100468394 C	11/03/2009		
		CN 1540552 A	27/10/2004		
		EP 1465084 A2	06/10/2004		
		EP 1465084 A3	28/03/2007		
		JP 04455120 B2	21/04/2010		
		JP 2004-303250 A	28/10/2004		
		KR 10-1002451 B1	17/12/2010		
		KR 10-2004-0088366 A	16/10/2004		
		MX PA04003000 A	01/07/2005		
		RU 2004-109570 A	20/10/2005		
		RU 2343537 C2	10/01/2009		
		US 2004-0254938 A1	16/12/2004		
		US 2005-0154723 A1	14/07/2005	CN 100495392 C	03/06/2009
				CN 1716244 A	04/01/2006
				US 2005-0144162 A1	30/06/2005
US 2005-0160107 A1	21/07/2005				
US 2006-0047649 A1	02/03/2006				
US 2006-0106793 A1	18/05/2006				
EP 1826692 A2	29/08/2007	EP 1826692 A3	25/03/2009		
		US 2007-208733 A1	06/09/2007		
US 2009-0287650 A1	19/11/2009	KR 10-2008-0000203 A	02/01/2008		
		WO 2008-002074 A1	03/01/2008		
US 2001-0020238 A1	06/09/2001	JP 2001-290843 A	19/10/2001		
		US 7225181 B2	29/05/2007		