

SYSTEM AND METHOD FOR PROVIDING AN INCENTIVIZED TAGGING SYSTEM FOR ELECTRONIC RESOURCES

DESCRIPTION

18 Drawings, 21 Claims

FIELD OF INVENTION

The present invention relates to system and method for providing user interface for tagging an electronic resource, exposing tag data for search and a reward system to credit users for tag relevance and usage.

BACKGROUND OF INVENTION

[001] Present enterprise search implementations utilize the same technology that is used in the web for searching documents and resources. A vast amount of structured and unstructured information in the form of documents and the like is available in an enterprise similar to web. This information may include business information such as, for e.g., proposal to specific industry customer, quotation for the industrial customer in a certain geographical area under a particular business unit, and may be in the form of spreadsheets, power point presentations, visio drawings, one note documents, pdf documents, HTML documents or a bevy of other formats and applications including custom ones. Essentially vast swaths of documents are churned by an enterprise in today's environment.

[002] In the enterprise environment, it is important to understand that information is changing both in terms of volume and accessibility; essentially, information provided in this environment is dynamic and agile. To realize the amount of useful information available, enterprises have recognized that one of the most valuable assets is, indeed, the intellectual capital gained through knowledge discovery and sharing through internet or through its own intranet. Leveraging this gained knowledge has become critical to gaining a strategic advantage in the competitive worldwide marketplace.

[003] Although burgeoning amounts of information is available within an enterprise, finding the most relevant information is a challenge to even the most sophisticated user. For example, it is necessary, prior to retrieving information, to

[004] understand from the business or enterprise point what information is really needed,

[005] how can that information be accessed and retrieved most efficiently & effectively and

[006] what specific knowledge would this information provide to the requestor and how the requester (e.g., an enterprise) can benefit economically or otherwise from such information.

[007] Without doubt, it has thus become increasingly important to devise a sound search strategy prior to conducting a search on the enterprise. This enables an enterprise to more effectively and efficiently utilize its resources. Hence by devising a coherent search strategy, it may be possible to gather information so that an informed and educated decision can be made. Without such proper and timely gathered information, it may be impossible or extremely difficult to accomplish a task or make a decision.

[008] Current enterprise search engines use the typical search engine technology available in the web to crawl the configured content and employ technologies like keyword search, boolean, concept, phrase, proximity, thesaurus, fuzzy, query-by-example, stop words and host of other search result optimizations and rankings. Certain search engines, if metadata is inherently available in the document repository, can search on them. These metadata attributes are defined and follow specific formats and are not open enough that can be applied to any resource in a simplified manner. Essentially if this rich metadata is not available, these search engines apply typical methods described above to search and hence has the following drawbacks

1. irrelevant and incomplete search results,
2. information overload experienced by users due to the inability of being able to narrow searches logically and quickly,
3. display of search results as lengthy lists of documents that are laborious to review

[009] In essence, these search engines never have the ability to understand business relevancy and effectively find metadata information associated with each of the document. They typically try to mine physical file properties as provided by operating system or their host handling program while searching the documents. Basically user involvement in adding additional business relevant tag information is missing for each of the resources that enterprises generate in today's environment in a simplified and affordable manner. If an enterprise user wants, for example, to search an architecture visio diagram file and associated architecture word & pdf document created in the year 2000 belonging to creative division in the Americas region with an associated proposal that was sold for 2 million isn't available. This kind of search is remote with current brew of enterprise search engines.

[010] Rich metadata for each of the resource shared, can be added automatically by system or by an enterprise user. If user adds business and business process related tags, then the metadata available for searching provide clues to answers for above search example. Hence a simplified, intuitive interface to collect and enter tag attribute and data is essential. To persuade and to achieve good business tagging, user participation in enterprise needs to be incentivized.

SUMMARY

[011] Embodiments of present invention provide a system and method to provide tag and search capability for electronic resources using a intuitive user interface system for tagging, creating multi-dimensional fact tables to expose tag data for searching and a reward system to credit users for tagging relevance and usage

[012] In accordance to an aspect of the invention, a user interface program to tag an electronic resource is provided. In one embodiment, access to user interface system is provided by right clicking the target file resource from the operating system file explorer and selecting the user interface program listed in menu. In another embodiment, access to user interface system is provided by a means for user to click a button on the tool bar or other configured areas within the target resource's handling program to launch the user interface within the host program's display or process application space. In another embodiment, access to user interface is provided via a browser that launches URL to fetch the user interface program available as a web application with plug-ins. In another embodiment, resources can be tagged by dragging and dropping file icons from operation system file explorer or URLs from browser into the user interface.

[013] In accordance to another aspect of the invention, user interface program comprises of a specific area to display information wherein a plurality of predefined business process or business classification tag attributes are displayed in the bottom of the interface visually as icons. Display of above said icons are in a carousel display format allowing users to scroll. Tag header and values pertaining to selected attribute are shown on the left as known in the art and is searchable. Tag values can be loaded on demand and narrowed down by searching.

[014] In accordance to another aspect of the invention, user interface program comprises of a specific area to display information wherein a plurality of tag attributes currently associated under different categories that include but not limited to dimensional and custom categories are shown. For designated categories, users can dynamically add as many tag attributes and values as required.

[015] In accordance to another aspect of the invention, user interface program comprises of a specific area to display information wherein a plurality most recently created tags associated to individual resources are shown in reverse chronological order. User can search by entering filter criteria to narrow down the requisite tags for view and to apply it to current resource being tagged.

[016] In accordance to another aspect of the invention, user interface program comprises of a specific area to display information wherein tag statistics is displayed that comprises of count of resources that has been tagged, searched, viewed and downloaded and reward points awarded. Reward algorithm decides points to be awarded based on the tag relevance and usage factors configured.

[017] In accordance to another aspect of the invention, tag attributes are based and linked to columns of dimensional tables, master detail tables, hierarchical tables of a relational data store. Tag attributes are also based on headers of any structured data list.

[018] In accordance to another aspect of the invention, tag data of any tagged resource is saved locally in the computer system in XML format prior to submitting tag data and resource content to designated resource repository through a web service wherein web service or resource repository can be online or offline.

[019] In accordance to another aspect of the invention, tag data received by the web service is stored as XML document and database records are created in multi-dimensional fact table where in the tag data items and their derived data items forms part of the dimension or fact. These multi-dimensional fact tables, derived views and query interfaces are exposed to search engines and search algorithms.

[020] In accordance to another aspect of the invention, counts are maintained for each user whose tagged resources are accessed, viewed and downloaded through interfaces exposed. Also count of number of resources tagged by each and all users are maintained.

[021] In accordance to another aspect of the invention, a reward algorithm will assign points for each user determining tag relevance and usage based on the above said count information.

[022] In accordance to another aspect of the invention, a user interface program to search tag data XML files wherein a plurality distinct tag attributes along with their values from tag data XML files and recently searched text is shown on the left. Free form search text can be entered into search text box or a tag attribute with value can be selected or dragged and dropped into search text box to refine the search.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

[023] FIG. 1 depicts a representation of a network of data processing systems in which the present invention may be implemented;

[024] FIG. 2 is a block diagram illustrating a data processing system in which the present invention may be implemented;

[025] FIG. 3 is a block diagram illustrating components used in collecting, managing, storing, preparing and exposing tag metadata and resources in accordance with an illustrative embodiment of the present invention;

[026] FIG. 4 is a block diagram illustrating sub components used in client and server side in accordance with an illustrative embodiment of the present invention;

[027] FIG. 5 is a block diagram illustrating client user interface outline in accordance with an illustrative embodiment of the present invention;

[028] FIG. 6 is a block diagram illustrating invoking of client user interface from operating system file explorer in accordance with an illustrative embodiment of the present invention;

[029] FIG. 7 is a block diagram illustrating invoking of client user interface from within word editing program (Microsoft Word®) in accordance with an illustrative embodiment of the present invention;

[030] FIG. 8 is a block diagram illustrating invoking of client user interface from within spreadsheet program (Microsoft Excel®) in accordance with an illustrative embodiment of the present invention;

[031] FIG. 9 is a block diagram illustrating display of client user interface as web application served from a web URI in accordance with an illustrative embodiment of the present invention;

[032] FIG. 10 is a block diagram illustrating detailed client user interface elements and various display areas in accordance with an illustrative embodiment of the present invention;

[033] FIG. 11 is a flow chart diagram illustrating invocation of client user interface and data fetch process with an illustrative embodiment of the present invention;

[034] FIG. 12 is a flow chart diagram illustrating functionalities in client user interface and their business process with an illustrative embodiment of the present invention;

[035] FIG. 13 is a block diagram illustrating various configurations in which the present invention may be implemented;

[036] FIG. 14 is a block diagram illustrating server data reception process with an illustrative embodiment of the present invention;

[037] FIG. 15 is a block diagram illustrating server data preparation process with an illustrative embodiment of the present invention;

[038] FIG. 16 is a block diagram illustrating server plug-in and search process with an illustrative embodiment of the present invention;

[039] FIG. 17 is a block diagram illustrating client user interface elements and various display areas in accordance with an illustrative embodiment of the present invention used to search tag data xml documents saved with tagxml file extension.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[040] With reference now to the figures and in particular with reference to FIGS. 1-2, exemplary diagrams of data processing environments are provided in which embodiments of the present invention may be implemented. It should be appreciated that FIGS. 1-2 are only exemplary and are not intended to assert or imply any limitation with regard to the environments in which aspects or embodiments of the present invention may be implemented. Many modifications to the depicted environments may be made without departing from the spirit and scope of the present invention.

[041] With reference now to the figures, FIG. 1 depicts a pictorial representation of a network of data processing systems in which aspects of the present invention may be implemented. Network data processing system 100 is a network of computers in which embodiments of the present invention may be implemented. Network data processing system 100 contains network 102, which is the medium used to provide communications links between various devices and computers connected together within network data processing system 100. Network 102 may include connections, such as wire, wireless communication links, or fiber optic cables.

[042] In the depicted example, server 104 and server 106 connect to network 102 along with storage unit 108. In addition, clients 110, 112, and 114 connect to network 102. These clients 110, 112, and 114 may be, for

example, personal computers or network computers. In the depicted example, server 104 provides data, such as boot files, operating system images, and applications to clients 110, 112, and 114. Clients 110, 112, and 114 are clients to server 104 in this example. Network data processing system 100 may include additional servers, clients, and other devices not shown.

[043] In the depicted example, network data processing system 100 is the Internet with network 102 representing a worldwide collection of networks and gateways that use the Transmission Control Protocol/Internet Protocol (TCP/IP) suite of protocols to communicate with one another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers, consisting of thousands of commercial, governmental, educational and other computer systems that route data and messages. Of course, network data processing system 100 also may be implemented as a number of different types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). FIG. 1 is intended as an example, and not as an architectural limitation for different embodiments of the present invention.

[044] With reference now to FIG. 2, a block diagram of a data processing system is shown in which aspects of the present invention may be implemented. Data processing system 200 is an example of a computer, such as server 104 or client 110 in FIG. 1, in which computer usable code or instructions implementing the processes for embodiments of the present invention may be located.

[045] In the depicted example, data processing system 200 employs a hub architecture including north bridge and memory controller hub (NB/MCH) 202 and south bridge and input/output (I/O) controller hub (SB/ICH) 204. Processing unit 206, main memory 208, and graphics processor 210 are connected to NB/MCH 202. Graphics processor 210 may be connected to NB/MCH 202 through an accelerated graphics port (AGP).

[046] In the depicted example, local area network (LAN) adapter 212 connects to SB/ICH 204. Audio adapter 216, keyboard and mouse adapter 220, modem 222, read only memory (ROM) 224, hard disk drive (HDD) 226, CD-ROM drive 230, universal serial bus (USB) ports and other communication ports 232, and PCI/PCIe devices 234 connect to SB/ICH 204 through bus 238 and bus 240. PCI/PCIe devices may include, for example, Ethernet adapters, add-in cards, and PC cards for notebook computers. PCI uses a card bus controller, while PCIe does not. ROM 224 may be, for example, a flash binary input/output system (BIOS).

[047] HDD 226 and CD-ROM drive 230 connect to SB/ICH 204 through bus 240. HDD 226 and CD-ROM drive 230 may use, for example, an integrated drive electronics (IDE) or serial advanced technology attachment (SATA) interface. Super I/O (SIO) device 236 may be connected to SB/ICH 204.

[048] An operating system runs on processing unit 206 and coordinates and provides control of various components within data processing system 200 in FIG. 2. As a client, the operating system may be a commercially available operating system such as Microsoft® Windows® XP, Windows® Vista (Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both). An object-oriented programming system, such as the C# programming system, may run in conjunction with the operating system and provides calls to the operating system from .NET programs or applications executing on data processing system 200 (.NET is a trademark of Microsoft in the United States, other countries, or both).

[049] As a server, data processing system 200 maybe, for example, an Wintel server computer system, running the Microsoft® Windows Server 2003 / 2008® operating system. Data processing system 200 may be a symmetric multiprocessor (SMP) system including a plurality of processors in processing unit 206. Alternatively, a single processor system may be employed.

[050] Instructions for the operating system, the object-oriented programming system, and applications or programs are located on storage devices, such as HDD 226, and may be loaded into main memory 208 for execution by processing unit 206. The processes for embodiments of the present invention are performed by processing unit 206 using computer usable program code, which may be located in a memory such as, for example, main memory 208, ROM 224, or in one or more peripheral devices 226 and 230.

[051] Those of ordinary skill in the art will appreciate that the hardware in FIGS. 1-2 may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash memory, equivalent non-volatile memory, or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in FIGS. 1-2.

[052] In some illustrative examples, data processing system 200 may be a personal digital assistant (PDA), which is configured with flash memory to provide non-volatile memory for storing operating system files and/or user generated data. Also, the processes of the present invention may be applied to a multiprocessor data processing system.

[053] A bus system may be comprised of one or more buses, such as bus 238 or bus 240 as shown in FIG. 2. Of course, the bus system may be implemented using any type of communication fabric or architecture that provides for a transfer of data between different components or devices attached to the fabric or architecture. A communication unit may include one or more devices used to transmit and receive data, such as modem 222 or network adapter 212 of FIG. 2. A memory may be, for example, main memory 208, ROM 224, or a cache such as found in NB/MCH 202 in FIG. 2. The depicted examples in FIGS. 1-2 and above-described examples are not meant to imply architectural limitations. For example, data processing system 200 also may be a tablet computer, laptop computer, or compatible processing device in addition to taking the form of a PDA.

[054] Embodiments of the present invention provide tagging functionality to electronic resources including electronic documents, files and URL resources. User interface part of the invention can be hosted from within the display area of any resource handling program subjected to such facility available. The user's resource handling programs may include a calendaring program, document editing program, spreadsheet program, drawing editing program and presentation document editing program.

[055] FIG. 3 represents an overview of an exemplary tag, submit, search and retrieval application which may be used to implement the method and system of the present invention. It should be recognized by those of ordinary skill in the art that the system and method of the present invention may equally be implemented over a host of other application platforms, and may equally be a standalone module. Accordingly, the present invention should not be limited to the application shown in FIG. 1, but is equally adaptable as a standalone module or implemented through other applications, search engines and the like.

[056] The overall system architecture shown in FIG. 3 includes five modules: (i) User Interface module 300, (ii) Middleware module 301, (iii) Resource Repository Store 302, (iv) Search Engine Server 303 and (v) RDBMS Data Store 304 will be briefly described in order to provide an understanding of the overall exemplary system; however, the present invention is directed more specifically to innovations associated with the Client module 300 and Middleware module 302.

[057] User Interface module 300 is client application running on the operating system allows users to collect, manage and edit tag metadata associated to a document. Even though the present invention is described herein with reference to a generic electronic resource, may also include other documents such as word processing documents, ADOBE documents, workbooks, web pages, emails, notes or any other type of document or file or resource where tagging is desired. It should be appreciated that, according to one embodiment of the invention, a spreadsheet document comprises the Excel® spreadsheet application program, a document edit program comprises the Word® edit program, a presentation document comprises the PowerPoint® program, a note document comprises OneNote® program, email document comprises of Outlook® messaging program, drawing document comprises of Visio® editing program from Microsoft Corporation headquartered in Redmond, Wash and pdf document comprises of Adobe® document editing application from Adobe Corporation. It should be appreciated, however, that other document or file or resource application programs from other manufactures may be utilized to embody various aspects of the present invention. Further, the User Interface module allows users to submit tag data and the concerned resource content to middleware 301 which can store the tag data and document inside a RDBMS data store 304 and a resource repository store 302 respectively.

[058] FIG. 4 represents an exemplary aspect of the present invention to be compared with FIG. 3 above. FIG. 4 represents in detail the various embodiments of the client user interface and the server middleware.

[059] The client user interface 409 may include a number of program modules and data files stored on the client comprising Operating System Service 401, User Interface program 400 and local file storage 402. The user interface 400, in one embodiment, may have a Web browser application that is operative to request, receive, render, and provide interactivity with electronic documents, such as a Web page that has been formatted using HTML. According to one embodiment of the invention, the Web browser application comprises the Internet Explorer® Web browser application program from Microsoft Corporation. It should be appreciated, however, that other Web browser application programs from other manufactures may be utilized to embody various aspects of the present invention. The user interface 401, in another embodiment, may have a WINDOWS application that comprises of Windows Forms®, Windows Presentation Foundation® and Silverlight® technologies from Microsoft Corporation. User interface 400 can connect to Middleware 410 in a server system directly or alternatively through the Operating System Service 401. 401 also provide asynchronous services to upload or download files or resources.

[060] The middleware 410 may include a number of data providers comprising statistics data provider 404, tag data provider 405 and configuration data provider 406 and program modules comprising tag data and resource receiver module 403, tag data ETL (Extract, Transform and Load) engine module 407 and tag data search engine provider module 408. Statistics data provider 404 provides statistics data related to tag relevancy and usage on a per user basis. Tag data provider 405 provides tag item data in XML (eXtensible Markup Language) format that can be retrieved from tables comprising dimensional tables, hierarchical tables, master-detail tables, master

tables or any designated tables from the RDBMS data store or from any designated files or resources as anyone in the art understands. XML tag item data can be cached in the client and can be accessed by the proxy providers in the client in offline mode when connectivity to middleware is not available. ETL module 407, in general has database objects and routines to implement extraction, transformation of tag data received by module 403 and loading of prepared data into multi-dimensional tables. Module 403 also facilitates saving of resource content in a designated resource repository. In one embodiment, the resource repository may be a system running Windows XP or Windows Vista or Windows 2008 Server software with files services or any other operating system with file services. In another embodiment, resource repository can be a RDBMS data store or a Microsoft SharePoint Server or any server system capable of holding content. Tag data search engine provider module 408 comprises of search engine plug-in and search API implementation supporting any Search Server or as a standalone. In one embodiment, Search Server may be a Microsoft Search Server or a Microsoft SharePoint Server or any other search server or appliance implementations.

[061] Every tag item data provided by tag data provider module 405 comprises of a unique number and source name associated to tables in a RDBMS store or local file or any resource. Tag attribute metadata provided by module 405 comprises of properties associated to the tag like name, source table, number of occurrence, and other associated source table names. Tags from previously tagged documents are also provided by this module.

[062] FIG. 5 represents an exemplary aspect of the present invention wherein it depicts the general screen display outline of the client user interface comprising of various display areas dedicated to provide specific information for intuitive collection and submission of tag data and resource content. Those of ordinary skill in the art will appreciate that display area size, color and appearance may vary but comprises of a status bar panel 509 at the bottom, Tag item data comprising tag attributes and their values is presented in panel 501 and a tag data item search panel 502 below it. There's a sliding error message panel below 502. On the right is previous tags listing panel 506, following below is Statistics data panel 507 and toolbar panel 508. Straddling between left and right panel at the middle is the Tag dashboard 503 displaying the resource properties and tag attribute names and values, below this is action bar 504 that allows various operations on the current resource being tagged. Immediately above status bar 509, in the middle is the tag attribute name selector panel 505.

[063] As contemplated in the present invention, the client user interface can be invoked in different ways. FIG. 6 represents an exemplary aspect of the present invention wherein the user interface may be invoked from the operating system file explorer 600. Users may right click on the concerned document or file or resource as displayed by the file explorer and look for text description entry 601 to click on it or any of its context menu items. In one example, the text description may be "Tag with Searchprise". In another example may be "Tag with ..." followed by a listing of context menu items. In another example the description text may vary and may be configured.

[064] FIG. 7 and FIG. 8 represents an exemplary aspect of the present invention wherein the user interface 701 and 801 may be invoked within the word edit program and spreadsheet edit program respectively using the toolbar panel or ribbon panel 700 and 800 provided by the host resource handling program like Microsoft Word® or Microsoft Excel® programs. In one embodiment, this mechanism can be implemented across Microsoft Office® 2003 and 2007 programs comprising Word®, Excel®, PowerPoint®, Outlook®, OneNote® and InfoPath®. In another embodiment, user interface can be invoked from any host resource handling program

subject to API (Application Programming Interface) availability and a visual graphic icon display which can be invoked to trigger the user interface. In one example, text in 700 and 800 may be "Tag & Search" with graphical icons. In another, description and graphical icons may vary and can be configured.

[065] FIG. 9 represents an exemplary aspect of the present invention wherein the user interface 901 is displayed in the browser 900. User interface 901 may be rendered from XBAP® (XAML Browser Application) or Silverlight® or ActiveX® application, all technologies from Microsoft Corporation or rendered by a HTML (HyperText Markup Language) application.

[066] FIG. 10 represents an exemplary aspect of the present invention wherein it depicts details of the tag user interface elements as outlined in FIG. 3. Tag Items panel 1000 allows the tag item data to be displayed and selected using a tree view controller or a list box. Further the data can be searched using a tag item search text box and button in panel 1020. A sliding error message panel 1019 is displayed whenever there's an error in operation. Tag dashboard 1001 comprises of 2 panels at the minimum wherein Resource Properties panel 1002 displays resource (selected file or document or resource) properties and their values. Each category tag attribute names and values as in panels 1004 and 1005 appear in succession. With as many category of tags, as many panels are shown. Action bar 1007 displays context menu enabled buttons or simple buttons that can act upon the tags and resource being tagged. Tag attribute name selector for each category as selected from the dropdown 1013, is shown as a carousel of graphical icons in carousel graphical control 1014 with scrollable arrow displays 1015. Selectable attribute name is displayed in a label 1016, as the user moves along each of the graphical icons. Tag collections from previously tagged documents are listed in tag listing panel 1008 as shown in the list view 1009. This list view allows drag and drop feature to copy the previous tag collection to current resource being tagged. Statistics panel 1010 displays important tag statistics information like counts related to total tagged documents, searches, downloads and previews made on the tagged documents and assigned reward points to current user logged in from the middleware data provider or from locally cached client data. Status bar 1018 displays the tag data item count and connectivity status.

[067] The process of tagging a resource as illustrated in FIG. 11 begins with invoking the user interface, essentially choosing the client interface 1100 is the next step to perform wherein selecting a thick client 1102, user has the option 1103 to invoke it from the operating system file explorer by right clicking on the concerned file and selecting the user interface or by clicking the installed icon from desktop or program group or by invoking from within the resource or document handling program or by dragging and dropping the file icon from operating system file explorer into the interface. Process 1100 moves to 1104 in the event thin client delivery of user interface is desired and browser 1105 need to be launched with requisite URL to load the client interface. Once the client user interface 1105 is launched, the program checks middleware web server connectivity 1107, wherein with connectivity established 1108, upon successful authentication and authorization, proceeds to block 1113 to download configuration data. With failure to connect to middleware server, establishes a connection to local file storage and proceeds to block 1113. Next the process proceeds to retrieve tag categories 1114, followed by tag business dimensional or taxonomy tag item data 1115, tag statistics data 1116 and previous tags 1117 from tagged documents. If the local connectivity or storage is not available, shows default blank screen 1112 with an error message 1111. In the event, blocks 1116 and 1117 fails, the process proceeds to block 1119 with all acquired data so far and finally acquires resource data as in 1120. If blocks 1116 and 1117 succeeds. proceeds to 1118 to show all acquired data and then to 1120.

[068] FIG. 12 illustrates an exemplary aspect of the present invention wherein functionalities of the user interface is depicted as process flows wherein once a user interface with the resource to be tagged is launched successfully as in block 1201, different functions may be selected. To add new tag attribute name 1204, obtain the new name and call tag manager 1212 to include new tag attribute which then updates the internal storage 1213 and the display as in 1214. Tag attribute can be selected as in block 1215 by clicking on the graphical icon in the carousel panel and corresponding attribute value can be selected from the tag item selector panel as in 1216. Tag collection values as displayed in previously tagged item can be dragged and dropped as in block 1202 upon which this data is updated to current resource being tagged as per block 1203. For any operation under tag attribute removal 1206, tag attribute content clearance 1207 and entering value for tag attributes 1208, a tag attribute has to be selected 1209, followed by a call to tag manager in 1212. If attribute values need to be filled as in 1210, block 1211 fetches tag item data from various sources and displays in tag item selector panel. Whenever tags are saved 1217 and/or tag resource file submitted 1218, validation routine 1219 happens. If successful, proceeds to save the tag content locally 1220 and updates display and checks for middleware server connectivity 1221. If connectivity is successful, proceeds to save tag data to server and if resource is submitted 1223, file is uploaded. Finally statistics is retrieved 1225 and display 1214 is updated. Statistics can be refreshed 1226 separately.

[069] FIG. 13 illustrates an exemplary aspect of the present invention wherein different configurations of the present invention can be implemented. In one embodiment, all components of the solution may follow a highly scalable and available setup. In this scenario, multiple clients 1300 connect to load balanced middleware web server 1302 and load balanced search server 1303 through a network 1301. Those of ordinary skill in the art will appreciate that the network may be represented by internet or intranet over LAN or WAN or wireless in multitude of combinations and protocols. Web server is connected to load balanced middleware application server 1304 which in turn connects to clustered file server 1305 and clustered database server 1307 which is hooked to a common redundant SAN storage 1306. Search server can access both file and database server. In one example, but not limited to, the web, file and application server above may run Windows Server software. Database server may run Microsoft SQL Server. Search Server may run Microsoft Search Server or Microsoft SharePoint Server.

[070] In another embodiment of present invention configuration as in FIG. 13, multiple clients 1308 connects to a single and combined web and application server 1310 and search server 1311 through a network 1309. They in turn connect to database server 1312 and file server repository 1313. In one example, 1310, 1312 and 1313 may run Windows XP or Windows VISTA or Windows Server software. 1311 may run Microsoft Search Server Express or Server. 1312 may run Microsoft SQL Express or SQL Server.

[071] In another embodiment of present invention configuration as in FIG. 13, multiple clients 1314 connects to search server 1316 and a single and combined web, application, database and file server 1317. In one example, 1316 may run Windows XP or Windows VISTA or Windows Server software. 1316 may run Microsoft Search Server Express or Server and Microsoft SQL Express or Server. It should be appreciated that other web, file, application, database and search server from other manufactures may be utilized to embody the present invention as illustrated in FIG. 13.

[072] FIG. 14 illustrates an exemplary aspect of the present invention wherein the middleware process of receiving tag data and file and its processing is depicted. This process starts at block 1400 followed by 1401 receiving the tag data. Received data is ascertained for content type. If content type is tag data as in 1402, follows to block wherein submission entry records are created and unique ID is provided as in 1403 and moves to block 1412 to mark it for processing. If the content type is resource content as in 1404, proceeds to check for the tag data entry for this content as in 1405. If successfully tracks down a submission entry, moves to save the content on the repository server 1406 and updates tag entry record 1407 and marks for processing 1412. If unsuccessful at 1405, moves to 1408 to save the file in repository server and marks as error in submission for resolution. If the content type received at 1410 is of permission related data, moves to block 1411 to set up permission on the content authorization accordingly.

[073] FIG. 15 illustrates an exemplary aspect of the present invention wherein explains the middleware tag ETL engine process. A scheduler as in 1500 invokes the ETL process on preset time interval or on demand. ETL process as in 1501 scans the submitted tag data that are unprocessed and collects them for processing. ETL process 1502 retrieves unique identifier associated to each tag item data, its source table in the data store and category. Further moves to 1503 to determine the validity of source table data and then prepare additional dimensional data from other tables and configurations 1504. Subsequently prepares data for insertion into multi-dimensional fact table as in 1505 and inserts into the table and updates the processed status as in 1506.

[074] FIG. 16 illustrates an exemplary aspect of the present invention wherein depicts search server plug-in and search API interaction with components of the solution. Client 1600 connects to search server 1602 to perform searching wherein the search plug-in 1603 installed inside or outside of the search server acts as an intermediary to talk to search API 1604 in the middleware server which queries the database server 1605 that contains the multi-dimensional views. Search API also fetches the resource content for preview and download by the client either through search server or directly. Activities like previews, downloads and searches delivered by API through the search multi-dimensional database and file server are tracked and counts per user per tagged resource is updated.

[075] FIG. 17 represents an exemplary aspect of the present invention wherein it depicts details of the search user interface elements. This embodiment allows tags to be searched locally by loading all tagxml documents cached or saved locally without involving fact tables and search engines. Tag Items panel 1700 loads distinct tag attributes and their distinct values from all tagxml documents and displays using a tree view representation. Tag attributes can be selectively chosen for inclusion in the search criteria. Tag attributes and values data can be narrowed down or refined by using a tag item search text box and button in panel 1701. 1702 comprises of 2 areas namely search text entry area and search results display area wherein a search text box 1703 allows to enter search criteria in free form text and a panel 1705, below, allows tag attributes and their values to be entered by either selecting or dragging and dropping them in this area. Panel 1705 is displayed only when tag attributes are selected. Search results 1706 are displayed upon invoking button 1704. Search statistics and status info are shown in status bar 1707.

[075] FIG. 18 represents an exemplary aspect of the present invention wherein it depicts details of the tag data XML document saved with a file extension tagxml. This embodiment shown is a XML document conforming to W3C standards body and houses all tag data between TagXMLDocument elements 1800. ResourceTags element

1801 contains group of tag categories. In this example, categories include, but not limited to DimensionTags 1802, CustomTags 1804, UserTags and ResourceProperties 1805. In another example, it may be of different description and can be varied and configured. Each tag category comprises of corresponding collection of tags generally represented as [TagCategory]Tag and in this example as in DimensionTag element 1802. Each tag comprises of Name representing the tag attribute name, Source representing source table name or source list, Type representing tag category type, IsMultiTagContent representing whether the tag value comprises of multiple tag values. Each tag contains one or multiple tag content elements 1803 which comprises of value and a unique ID. TagMetaData element 1806 holds all tagging operation details and also the saved tagxml document name which has a unique name by prefixing a GUID with resource name and adding tagxml file extension at the end.

CLAIMS

Invention claimed is:

1. A computer implemented method to perform associative tagging for a selected document or file or resource, the computer implemented method comprising:
 - receiving an input to display a computer implemented user interface;
 - providing a plurality of graphical objects on the display that each represents an associative tag attribute belonging to each defined tag category;
 - creating database fact table records using each tag items as derived dimensions and facts;
 - exposing database fact tables and views for search, and providing query interfaces to search algorithms and search engines and tracking search activities.
2. The computer implemented method of claim 1, wherein receiving an input to launch and display user interface further comprises:
 - receiving an input by right clicking the resource displayed in operating system file explorer and selecting the user interface program listed;
 - receiving an input to display the user interface from within the resource's host program display area or outside the host program display area;
 - receiving an input to display user interface based on the operating system feature to associate specific file type extension to user interface program.
3. The computer implemented method of claim 2, wherein an electronic resource comprises of any files or documents stored in the computer's physical folders, network folders, web folders and URL addressable resources and objects in memory.
4. The computer implemented method of claim 1, wherein tag data collected and entered in the user interface comprises of plurality of text characters representing tag attribute names and their values including resource properties, can be represented and stored in database table or XML document with a unique key to identify each tag data value.
5. The computer implemented method of claim 4, wherein, XML document is named with a file extension tagxml conforming to XML document standards and comprising TagXMLDocument element as the starting element that contains tag category elements. Each tag category element houses a collection of tags. Each tag comprises tag meta data that includes name, source, type, multi-tag content indicator and a collection of tag content elements that contains a tag data value and a unique identifier.
6. The computer implemented method of claim 1, wherein, user interface may be invoked as a plug-in from within a calendaring, document editing, spreadsheet, diagram editing and presentation editing programs. Tag data is stored in the custom property storage placeholder offered by these programs as XML document and also separately as a XML document in a file.

7. The computer implemented user interface of claim 1, wherein user interface provides specific display areas comprising:
 - providing a display area for associative attributes as plurality of graphical objects at the bottom of the interface where in graphical objects is visually displayed using a carousel representation wherein graphical object size expands upon input device traversal over them or while selecting a graphical icon;
 - providing a display area for tag attributes and their data values including the resource properties and their values at left and all selected and assigned tag attributes and values for the current tagged resource at top center;
 - providing a display area for statistics at bottom right;
 - providing a display area for tag data from previously tagged items at the top right;
 - providing a display area for acting on tag data as to add, remove, fill, clear, save and submit tag attributes and values and resource content at the center;
 - providing a display area for error message at left immediately above status message panel;
 - a display area for the status message panel at the bottom.
8. The computer implemented user interface of claim 6, wherein tag data values comprising attribute names and values, their retrieval and display, comprising:
 - retrieving and displaying of selected column's row values as tag attribute values from tables including dimensional tables or hierarchical tables or master-detail tables or any XML document representing table data;
 - retrieving and displaying of tag data either online or offline from the above data sources represented in any format on a computer readable storage and access medium;
 - retrieving and displaying of tag data values from URL feeds and resources;
 - retrieving and displaying of tag data values from APIs provided by programs that handle or open a tagged document or file or resource.
9. The computer implemented user interface of claim 6, wherein display area for resource properties, comprising:
 - displaying of selective meta data properties of the resource as provided by APIs of the programs that handle these resources or by the operating system.
10. The computer implemented user interface of claim 7, wherein tag data can be extracted from APIs comprising:
 - displaying of table of contents (TOC), header , footer , title properties for any document using API provided by the document editing program;
 - displaying of slide title, slide content, header and footer, title properties for any document using the API provided by presentation editing program;
 - displaying of worksheet titles, formulae for any spreadsheet document using the API provided by spreadsheet editing program;
 - displaying of email, meeting, task attribute values using the API provided by the messaging and calendaring program.

11. The computer implemented user interface of claim 6, wherein display area for listing tag data from previously or most recently tagged resources on the top right, comprising:
 - displaying tag data collection from previously tagged resources in reverse chronological order
 - and a means to perform drag and drop operation to apply selected tag data to current resource being tagged.
12. The computer implemented method of claim 1, wherein user interface implementation further comprising:
 - a means to select tag data items from left display area and assign to tag attributes;
 - a means to create custom tag attributes;
 - a means for validating associated tag attributes and their assigned collection of tag values based on validation rules;
 - a means to submit this tagged data along with resource content to a repository store through a web URL that handles such submission;
 - a means to save tag data locally as a XML document with a file extension called tagxml;
 - a means to uniquely name the resource submitted;
 - a means to attach the unique identifier and source table name that identifies a tag value during tagging operation in the user interface;
 - a means to submit to 3rd party repositories utilizing their respective APIs.
13. The computer implemented method of claim 11, wherein submission step further comprising:
 - a means to submit this content containing the tagged data or resource content or both, Securely over any transport protocols;
 - a means to allow offline submission with no connectivity available.
14. A system for storing the fact table records that includes tag attributes as participating dimensions, comprising:
 - a relational data store that contains fact tables wherein fact tables can be data marts or data cubes in a RDBMS data store;
 - a schema definition defining the schema of the fact tables where tag attributes form dimensions;
 - a query interface for searching the fact table.
15. The computer implemented method of claim 1, wherein creating fact table records, comprising:
 - a computer usable program code to receive the submitted content from the user interface and store content and tags separately and to analyze the submission;
 - a means to perform authorization operation to expose the resource for access by other users or computer programs.
16. A computer program code in claim 14, wherein receiving submitted tag data and content further comprising:

a web service exposed over an URL that can securely receive the submission

17. A computer program code in claim 14, wherein creating the database fact table record further comprising:
- a means to extract and prepare unique keys from tag data values;
 - a means to derive dimension keys;
 - a means to create a fact table record using the above said keys;
 - a means to update related data marts and data cubes to expose them to search engines and search algorithms;
 - a means to expose prepared tag data to search engines through search engine plug-ins.
18. The computer implemented method of claim 1, wherein tracking search activities on the tag data of tagged resources, comprising:
- a means to count the number of searches and previews done by users from search engine or search algorithm;
 - a means to count the number of downloads of the uniquely named resource from the resource repository;
 - a means to expose API for count information for 3rd party computer programs to consume.
19. The computer implemented method of claim 1, wherein tag rewarding system, comprising:
- a means to award points to users whose tagged resources are accessed by other users using search engines and search algorithms;
 - a means to compute reward points based on above said counters in claim 17 and from custom formulae specified;
 - a means to display the reward points real time and offline;
 - a means to expose API for reward information for 3rd party computer programs to consume.
20. The computer implemented method of claim 1, wherein a user interface implementation to search tag data XML documents, comprising:
- providing a display area to list all recently searched terms and search tag attribute and value combinations on the left;
 - providing a display area to list all tag attributes and their values extracted from the tagxml files on the left;
 - a means to select the tag attribute;
 - a means to display all selected tag attributes and values used to refine the search along with search text box to enter free form search criteria;
 - a means to display a panel containing the search criteria attributes and values below the search text box only when tag attributes are selected;
 - a means to store and keep track of the successful search text and tag attribute and value combinations.

21. The computer implemented method of claim 1, wherein a user interface implementation to configure and create tag categories, attributes and values, comprising:

- a wizard based interface to create tag category and associated tag attributes and values;

- a means to collect tag values as list or linked list or hierarchical linked list which can be extracted from XML document, comma separated text document or database tables;

- a means to include tag attribute images to be displayed as tag dimension graphical icons in the user interface;

- a means to save above data as XML document used by the user interface program to initialize and facilitate tagging operation.