(12) STANDARD PATENT (11) Application No. AU 2006304109 C1
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Managing relationships between resources stored within a repository

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

(21) Application No: **2006304109** (22) Date of Filing: **2006.10.10**

(87) WIPO No: **WO07/047300**

(30) Priority Data

(31) Number	(32) Date	(33) Country
11/256,527	**2005.10.20**	**US**

(43) Publication Date: **2007.04.26**
(44) Accepted Journal Date: **2012.09.13**
(44) Amended Journal Date: **2013.05.16**

(71) Applicant(s)
Oracle International Corporation

(72) Inventor(s)
Murthy, Ravi;Sedlar, Eric

(74) Agent / Attorney
Pizzeys, PO Box 291, WODEN, ACT, 2606

(56) Related Art
US 2002169788 (Lee et al.)
US 7072896 (Lee et al.)
US 6044378 (Gladney)
WILDE E ET AL: "From content-centered publishing to a link-based view of information resources" SYSTEM SCIENCES, 2000. PROCEEDINGS OF THE 33RD ANNUAL HAWAII INTERNATIONAL CONFERENCE ON JPISCATAWAY, NJ, USA,IEEE, 4 Jan 2000, pp 824-833 XPO

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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
26 April 2007 (26.04.2007)

PCT

(10) **International Publication Number**
WO 2007/047300 A1

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

(21) **International Application Number:**
PCT/US2006/039706

(22) **International Filing Date:** 10 October 2006 (10.10.2006)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
11/256,527 20 October 2005 (20.10.2005) US

(71) **Applicant** *(for all designated States except US)*: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, Redwood City, CA 94065 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **MURTHY, Ravi** [IN/US]; 33227 Jamie Circle, Fremont, California 94555 (US). **SEDLAR, Eric** [US/US]; 956 Woodside Road, #3, Redwood City, California 94061 (US).

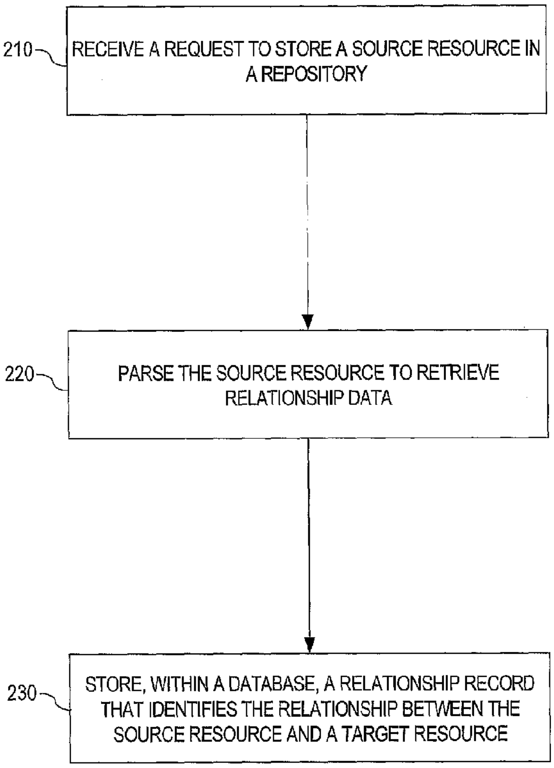
(74) **Agent: BROKAW, Christopher**; HICKMAN PALERMO TRUONG & BECKER LLP, 2055 Gateway Place, Suite 550, San Jose, CA 95110, (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) **Title:** MANAGING RELATIONSHIPS BETWEEN RESOURCES STORED WITHIN A REPOSITORY


210
RECEIVE A REQUEST TO STORE A SOURCE RESOURCE IN A REPOSITORY
220
PARSE THE SOURCE RESOURCE TO RETRIEVE RELATIONSHIP DATA
230
STORE, WITHIN A DATABASE, A RELATIONSHIP RECORD THAT IDENTIFIES THE RELATIONSHIP BETWEEN THE SOURCE RESOURCE AND A TARGET RESOURCE

(57) **Abstract:** A method and apparatus for managing relationships between resources stored in a repository is provided. A client sends, to a server, a request to store a first resource within a repository. In response to receiving the request, the server parses the first resource to retrieve relationship data that identifies a relationship between two or more resources to be stored, or currently stored, within the repository. The server stores, within a database accessible to the server, one or more relationship records that identify the relationship between the two or more resources. The one or more relationship records are stored separate from the two or more resources. Subsequently, the client may issue queries, to the server, about the one or more relationships records stored in the database. In this way, a user may access the one or more relationship records to analyze the relationship between resources stored in repository.

Published:

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

MANAGING RELATIONSHIPS BETWEEN RESOURCES STORED WITHIN A REPOSITORY

FIELD OF THE INVENTION

[0001] Embodiments of the present invention relate to managing the relationships between resources stored in a repository.

BACKGROUND

[0002] A digital resource (or simply a "resource"), as broadly used herein, refers to any unit of digital data stored as an individual entity. Non-limiting, illustrative examples of a resource include a document, an image, a folder, and a file.

[0003] Resources may have different types of relationships with each other. In some cases, the relationship between two resources may be identified by one of the resources. For example, a first web page may have a relationship with a second web page because the first web page includes a web link to the second web page. Thus, when a user selects the web link on the first web page, the second web page is displayed to the user.

[0004] In other cases, the relationship between two resources may not be identified by either of the two resources. For example, a thumbnail image has a relationship with the original image to which the thumbnail corresponds because both the thumbnail image and the original image depict the same image. However, while both the thumbnail image and the original image depict the same image, there is no indication in either of the thumbnail image or the original image of the existence of the other. As another example, a first text document may have a relationship with a second text document because the content of the first text document includes the content of a second text document. However, the first text document may not identify the existence of the second text document.

[0005] Resource management applications are increasingly being used to store resources that are expressed in a self-describing metalanguage, such as the extensible markup language (XML). XML is a language that allows a resource to be defined as a tree of elements. XML is a markup language that allows tagging of document elements and provides for the definition, transmission, validation, and interpretation of data between applications and between organizations. The XML specification was developed by the W3C consortium and is located on the Internet at "http://www.w3.org/XML".

[0006] While XML has been used to describe the contents of a resource, current techniques in the art do not adequately provide for managing the relationships between resources stored in a repository. Consequently, an approach for managing the relationships between resources stored in a repository is desirable. The approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Embodiments of the present invention are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0008] FIG. 1 is a graphical depiction of a system according to an embodiment of the invention;

[0009] FIG. 2A is a flowchart illustrating the functional steps of storing a source resource within the repository according to an embodiment of the invention;

[0010] FIG. 2B is a flowchart illustrating the functional steps of storing a relationship-identifying resource within the repository according to an embodiment of the invention;

[0011] FIG. 3 is a graphical depiction of a schema for an element of a resource configuration according to an embodiment of the invention; and

[0012] FIG 4 is a block diagram that illustrates a computer system upon which an embodiment of the invention may be implemented.

DETAILED DESCRIPTION

[0013] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the embodiments of the present invention described herein. It will be apparent, however, that the embodiments of the present invention described herein may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the embodiments of the present invention described herein.

FUNCTIONAL OVERVIEW

[0014] Approaches are presented herein for managing the relationship between resources stored in a repository. According to a first embodiment, a client sends, to a server, a request to store a first resource. In response to receiving the request, the server parses the first resource to retrieve relationship data that identifies a relationship between the first resource and a second resource in a repository accessible to the server. The relationship data may be expressed in the first resource by a variety of ways, e.g., the relationship data may be expressed using an XML linking language (XLink). Version 1.0 of the XLink specification (the "XLink specification") is available from the W3C consortium. The XLink specification may be obtained on the Internet at http://www.w3.org/XML, and is incorporated by reference as if fully set forth herein. The server stores, within a database accessible to the server, one or more relationship records that identify the relationship between the first resource and the second resource. The one or more relationship records are stored separate from the first resource. Subsequently, the client may issue queries, to the server, about the one or more relationships records stored in the database. In this way, a user may access the one or more relationship records to analyze the relationship between resources stored within the repository.

[0015] In another embodiment, a first resource and a second resource are stored within a repository. Neither the first resource nor the second resource contains any link to the other. In response to a server receiving a request, from a client to store a third resource ("a relationship-identifying resource") in a repository accessible to the server, the server parses the relationship-identifying resource to retrieve relationship data that identifies a relationship between the first resource and the second resource. The relationship data may be, but need not be, expressed within the relationship-identifying resource using XLink. The server stores, within a database accessible to the server and separately from the first resource, one or more relationship records that indicate the existence of a link from the first resource to the second resource. In this way the client may subsequently issue queries to the server about the one or more relationships records stored in the database, even though neither the first resource nor the second resource contains any link to the other.

ARCHITECTURE OVERVIEW

[0016] FIG. 1 is a graphical depiction of a system 100 according to an embodiment of the invention. The system 100 of FIG. 1 may be used to manage the relationships between resources stored in a repository. According to the embodiment depicted in FIG. 1, system 100 comprises a client 110, communications links 120 and 122, server 130, persistent storage 140, repository 150, one or more relationship records 160, and one or more resources 170. System 100 of FIG. 1 is merely illustrative; other embodiments of the invention may employ different components than those depicted in FIG. 1.

[0017] A client 110 may be implemented by any medium or mechanism that provides for issuing a request to store a resource in repository 150. For example, a user may use client 110 for storing and retrieving resources to and from repository 150. Non-limiting, illustrative examples of client 110 include a web browser, a wireless device, a cell phone, a personal computer, and a software application.

[0018] Communications link 120 may be implemented by any medium or mechanism that provides for the exchange of data between client 110 and server 130. Communications link 122 may be implemented by any medium or mechanism that provides for the exchange of data between server 130 and persistent storage 140. Examples of communications links 120 and 122 include, without limitation, a network such as a Local Area Network (LAN), Wide Area Network (WAN), Ethernet or the Internet, or one or more terrestrial, satellite or wireless links.

[0019] Server 130 may be implemented by any medium or mechanism that provides for receiving and processing requests from client 110. For example, server 130 may process a request, received from client 110, to store a resource in repository 150. Non-limiting, illustrative examples of server 130 include a database server or any server capable of issuing commands to persistent storage 140.

[0020] Persistent storage 140 may be implemented by any medium or mechanism that provides for persistently storing relationship records. A non-limiting, illustrative example of persistent storage 140 includes a database. In the embodiment depicted in FIG. 1, persistent storage 140 implements repository 150. In other embodiments (not shown in FIG. 1), repository 150 is not implemented by persistent storage 140, and the repository 150 is separate and distinct from the persistent storage 140.

[0021] Persistent storage 140 may be used to store one or more relationship records 160. A relationship record is a record that describes a relationship between two or more resources stored in repository 150. For example, in an embodiment, a relationship record may be embodied as one or more rows of one or more tables of persistent storage 140. A particular relationship record may describe the type of relationship between two or more resources, information that identifies the resources involved in the relationship, and any other information about how the relationship, such as how to identify, to client 110, the resources involved in the relationship.

[0022] Repository 150 may be implemented by any medium or mechanism that provides for implementing a file system. Repository 150 may provide a hierarchy of folders in which resources may be stored. Non-limiting, illustrative examples of repository 150 include a NFS file repository.

[0023] Repository 150 may be used to store one or more resources 170. A resource, as used herein, refers to any unit of digital data stored as an individual entity. Non-limiting, illustrative examples of a resource include a document, an image, a folder, and a file.

[0024] Having described an illustrative system 100 according to an embodiment, approaches for identifying the relationship between resources 170 stored in repository 150 shall be discussed.

IDENTIFYING A RELATIONSHIP BETWEEN A SOURCE RESOURCE AND A TARGET RESOURCE USING RELATIONSHIP DATA

[0025] For ease of explanation, several embodiments of the invention shall be discussed herein with reference to a first resource (denoted a source resource) that has a relationship to a second resource (denoted a target resource). Embodiments of the invention may manage relationships involving any number of resources; however, for purposes of providing a clear example, several embodiments will be presented with reference to a source resource that has a link to a target resource. A relationship between a source resource and a target resource may be identified to system 100 by information (denoted relationship data) stored in the source resource. Alternately, the relationship data may be stored in a different resource (denoted a relationship-identifying resource) other than source resource or the target resource.

[0026] Relationship-identifying resources are particularly useful when the source resource involved in the relationship cannot carry the relationship data. For example, a thumbnail has a particular relationship with the original image upon which the thumbnail image is based. However, the thumbnail image file cannot easily be modified to carry the relationship data without affecting the way in which the thumbnail image is rendered. Consequently, it may be advantageous to store the relationship data in a relationship-identifying resource, instead of the thumbnail image file itself.

[0027] Relationship data may be identified within the source resource or the relationship-identifying resource using a defined syntax for expressing the relationship data. For example, according to one embodiment, an XML linking language, such as XLink, may be employed to express the relationship data. XLink is a language that allows elements to be inserted into XML documents in order to create and describe relationships between resources.

[0028] Several embodiments shall be described herein with reference to XLink identfiying relationship data within a source resource or a relationship-identfiying resource; however, the use of XLink to identify the relationship data within a resource is merely exemplary of one embodiment, as other embodiments of the invention may use other syntaxes for identfiying relationship data within a source resource or a relationship-identfiying resource.

THE NATURE OF RELATIONSHIPS BETWEEN RESOURCES

[0029] Relationship data may identify a variety of different relationships between resources 170. For example, a first resource may have either an intra-resource or an inter-resource relationship with a second resource. An intra-resource relationship is a relationship wherein a source resource is composed of one or more target resources. On the other hand, an inter-resource relationship is a relationship wherein a first resource has an implicit or explicit relationship with one or more target resources, where the one or more target resources does not form part of the source resource.

[0030] To provide an example of an intra-resource relationship, a large document may be composed of two or more smaller documents. Separate parties may manage each of the smaller documents. Also, certain resources may be designed to be incorporated in many other resources, e.g., a boilerplate document (such as a warranty disclaimer or copyright notice) may be incorporated into numerous other documents. As the boilerplate document is likely to be stored separately from the documents that incorporate the boilerplate document, it would be advantageous to use system 100 to manage the relationships between the boilerplate document and those documents that incorporate the boilerplate document. In this way, system 100 may be used, as described in greater detail below, to prevent the boilerplate document from being accidentally deleted. Also, system 100 may be used to determine all the documents within the repository 150 that incorporate the boilerplate document.

[0031] To provide an example of an implicit inter-resource relation, a source resource may be a thumbnail image. While the thumbnail image is a smaller version of an original image, there is no indication in either of the thumbnail image or the original image about the existence of the other. Thus, the thumbnail image has a relationship to the original image, even though there is no express indication in the thumbnail image that the original image depicts the same image as the thumbnail image.

[0032] To provide an example of an explicit inter-resource relationship, a resource may contain data that identifies another resource. For example, a web page may include a link, specified using a Universal Resource Locator (URL), to one or more other target resources stored in the repository 150. As a further example of an explicit inter-resource relationship, a source resource may contain a link to a particular folder of repository 150.

[0033] The above description of exemplary relationships that may exist between resources 170 are not intended to limit the types of relationships that may be described by relationship data or relationship records 160, as relationship data and relationship records 160 may be used to describe a variety of relationships too numerous to fully enumerate herein.

[0034] Having described the types of relationships that may exist between resources 170, additional details about how those relationships are described by the relationship records 160 are presented below.

HARD, WEAK, AND SYMBOLIC LINKS

[0035] Information about the relationships between resources 170 are recorded in the relationship records 160 that are stored in the persistent storage 140. A particular relationship record may model a relationship between two or more resources 170 using a type of link between the two or more resources 170. Several examples of the types of links that may be identified by relationship records include a hard link, a weak link, and a symbolic link. The description of hard links, weak links, and symbolic links which follows is not meant to fully enumerate the type of links which may be identified by the relationship records 160, as relationship records 160 may be used to identify any type of link between a source resource and a target resource.

[0036] When a source resource has a hard link to a target resource, the target resource is uniquely identified, to repository 150, by the hard link. For example, the particular relationship record establishing the hard link may include a target resource identifier assigned to the target resource by the repository 150. Thus, if the target resource were to move to another location within the repository 150, the hard link between the source resource and the target resource would be preserved because the target resource is uniquely identified, within the repository 150, by the target resource identifier. In an embodiment where the repository 150 is implemented in persistent storage 140, the persistent storage 140 may assign the target resource identifier. For example, if the persistent storage 140 is a database, then the database may assign an object identifier to uniquely identify the target resource. In this way, regardless of where the target resource is moved to within the repository 150, the database may identity the target resource using the object identifier.

[0037] In addition, a hard link guarantees the integrity of the link. In other words, a target resource cannot be deleted from the repository 150 if any other resource, within the repository 150, has a hard link to the target resource. In this way, a hard link may be used to prevent the accidental deletion of a target resource that has a hard link to the target resource.

[0038] Similar to a hard link, when a source resource has a weak link to a target resource, the target resource is uniquely identified by the weak link. For example, the particular relationship record establishing a particular weak link may also include a target resource identifier assigned to the target resource by the repository 150. However, unlike a hard link, a weak link does not guarantee the integrity of the link. Thus, a weak link does not prevent a target resource from being deleted from the repository 150 if another resource, stored within the repository 150, has a weak link to the target resource.

[0039] Because a target resource that has a hard link to it cannot be deleted from the repository 150, the system 100 may be configured to require a certain permission level to add a hard link to a target resource. However, even if a user does not have a sufficient permission level to add a hard link to a target resource, the user may still wish to uniquely identify the target resource with the repository 150 so that the link is preserved if the target resource is moved to a different location within the repository 150. In such a case, using a weak link may be advantageous.

[0040] When a source resource has a symbolic link to a target resource, the symbolic link does not uniquely identify the target resource within the repository 150, but rather, the symbolic link identifies a location, within the repository 150, at which the target resource resides. Thus, if the target resource is moved to a different location within the repository 150, the symbolic link will no longer point to the target resource. However, if a new target resource is moved to the location identified by the symbolic link, then the symbolic link will point to the new target resource. Symbolic links may identify a location within the repository 150 by identifying a particular path, within the repository 150, to the location at which the target resource resides.

[0041] The paths to target resources, identified by symbolic links, are resolved when the symbolic link is accessed. Thus, a symbolic link may be useful when a resource wishes to maintain a link to a specific location within the repository 150, rather than linking to a specific resource within the repository. For example, there may exist seven folders in the repository 150 which correspond to the seven days of the week. Six of the seven folders are stored at a first location, and the folder that corresponds to the current day is stored at a second location. At the end of each day, the folder corresponding to the current day may be moved back to the first location, and the folder corresponding to the new day may be moved from the first location to the second location. Updates are made to the folder corresponding to the current day. In this way, a rolling archive of the week's activity may be performed. If a resource wishes to link to the folder corresponding to the current day, the intent is to link the resource to whichever folder happens to correspond to the current day, rather than linking to a specific folder. Thus, a symbolic link may be used in this situation to link a resource to the folder occupying the second location (the location where the folder for the current day is stored).

[0042] Additionally, symbolic links may be employed when it is desirable to link a resource to another resource that is stored outside of repository 150. Since a target resource identifier cannot be assigned by the repository 150 to a resource that is not maintained by the repository 150, a symbolic link may be used to describe this relationship. For example, the repository 150 may register a callback function to access a resource stored outside of repository 150. In this way, the symbolic link may reference and make use of the callback function to identify a location, where the target resource may be stored, outside of repository 150. Symbolic links also do not guarantee the integrity of the link. Thus, a symbolic link does not prevent a target resource from being deleted from the repository 150 if other resources stored within the repository 150 have a symbolic link to the target resource.

[0043] While embodiments of the invention shall chiefly be described with reference to a single source resource that has a relationship to a single target resource, a single resource may have a relationship with two or more resources. Accordingly, relationship data stored within a source resource or a relationship-identifying resource may identify that a source resource has a relationship with two or more target resources.

[0044] Having described several examples of the different types of relationships that may be identified by relationship data within a source resource or relationship-identifying resource, an approach for storing resources within the repository, according to embodiments of the invention, shall now be described.

STORING A SOURCE RESOURCE WITHIN THE REPOSITORY

[0045] FIG. 2A is a flowchart illustrating the functional steps of storing a source resource within the repository 150 according to an embodiment of the invention. The functional steps of FIG. 2A shall be explained below with reference to FIG. 1.

[0046] In step 210, a request to store a source resource in repository 150 is received by server 130. The request of step 210 may be sent by client 110 over communications link 120 to server 130. As described above, a source resource is a resource that which has a relationship to a target resource. A source resource may contain relationship data identifying the relationship the source resource has to the target resource. For example, a source resource may be a document that contains the relationship data expressed using XLink. After the request to store a source resource in repository 150 is received, processing proceeds to step 220.

[0047] In step 220, server 130 parses the source resource to retrieve relationship data from the source resource. For example, the server 130 may parse the source data to retrieve the relationship data from an XLink embedded within the source resource. Relationship data may be expressed in a variety of different formats within a source resource; as a result, XLink is merely one example of how relationship data may be identified within a source resource. After server 130 parses the source resource to retrieve the relationship data from the source resource, processing proceeds to step 230.

[0048] In step 230, server 130 stores, within persistent storage 140, one or more relationship records 160 that identify the relationship described by the relationship data retrieved in step 220. A relationship record 160 may be embodied as one or more rows of one or more tables of persistent storage 140.

[0049] The information stored in the relationship record 160 in step 230 contains all the information identified in the relationship data. The information stored in a relationship record 160 may vary based upon the relationship data retrieved in step 220. For example, if the relationship data retrieved in step 220 indicated that the relationship should be modeled as a hard link or a weak link, then an object identifier that uniquely identifies the target resource is stored in the relationship record. On the other hand, if the relationship data retrieved in step 220 indicated that the relationship should be modeled as a symbolic link, then a path, within the repository 150, at which the target resource resides, is stored in the relationship record.

[0050] Additionally, since a hard link guarantees the integrity of the link, whenever a new hard link is created within the repository 150, a hard link counter value, associated with the target resource, is incremented. Similarly, whenever a resource that has a hard link to a target resource is removed, then the hard link counter value associated with the target resource is decremented. In this way, the repository 150 may monitor, for each resource in the repository, how many other resources within the repository 150, have a hard link to a particular resource. The server 130 may prevent a particular resource, stored in the repository 150, from being deleted if the hard link counter value, associated with that particular resource, has a value greater than zero.

[0051] Having described the steps of storing a source resource, the steps of storing a relationship-identifying resource shall now be described.

STORING A RELATIONSHIP-IDENTIFYING RESOURCE WITHIN THE REPOSITORY

[0052] Relationship-identifying resources are particularly useful when a resource involved in the relationship cannot easily carry the relationship data. For example, a source resource may be expressed in a format that does not allow for the insertion of relationship data, such as, for example, a digital image or a resource expressed in a propriety format. In such a case, the relationship between the source resource and a target resource may still be managed by the system 100 by storing a relationship-identifying resource within repository 150 according to the steps of FIG. 2B. The steps of FIG. 2B shall be explained below with reference to FIG. 1.

[0053] FIG. 2B is a flowchart illustrating the functional steps of storing a relationship-identifying resource within the repository 150 according to an embodiment of the invention.

[0054] In step 250, a request to store a relationship-identifying resource in repository 150 is received by server 130. The request of step 250 may be sent by client 110 over communications link 120 to server 130. As described above, a relationship-identifying resource is a resource that contains relationship data within the resource, but is not the source resource. For example, a relationship-identifying resource may be a document that contains the relationship data expressed using XLink. After the request to store a source resource in repository 150 is received, processing proceeds to step 260.

[0055] In step 260, server 130 parses the relationship-identifying resource to retrieve relationship data. The performance of step 260 is substantially similar to that of step 220, except that the relationship data is retrieved from a relationship-identifying resource, rather than a source resource. After the relationship data is retrieved, processing proceeds to step 270.

[0056] In step 270, server 130 stores, within persistent storage 140, one or more relationship records 160 that identify the relationship between a source resource and a target resource. The performance of step 270 is substantially similar to that of step 230.

RETRIEVING RELATIONSHIP INFORMATION FROM THE REPOSITORY

[0057] The steps of FIG. 2A and FIG. 2B may be used to store relationship records 160 within the persistent storage 140. Once relationship records 160 are stored within persistent storage 140, a user may issue requests, to server 130, to view certain relationship records 160 that satisfy specified search criteria. A user may issue requests for information about the source resource, the target resource, link type, or any other property of the relationship that is stored in the relationship records 170. In this way, the user may view any information about the relationships between resources 170.

[0058] In an embodiment, the relationship records may be accessed using a database view. Users may issue a request, to server 130, to the database view to view certain relationship records 160 that satisfy criteria identified in the request. The database view may expose several columns, and each row of the database view may identify a particular relationship record 160. The columns of the database view may correspond to information about the various relationships between resources 170. For example, the columns of an illustrative, non-limiting database view, along with a corresponding description, are depicted below in Table 1.

Table 1

Column Name	**Description**
source resource identifier	an unique identifier for the source resource
source path	a path to the source resource within repository 150
target resource identifier	a unique identifier for the target resource
target path	a path to the target resource within repository 150
relationship type	the type of relationship, e.g., hard link, weak link, or symbolic link
relationship properties	information about the properties of the relationship, e.g., if the source resource is a thumbnail image of the target resource, that information may be identified here
relationship-identifying resource identifier	an unique identifier for the relationship-identifying resource
relationship-identifying resource path	a path to the relationship-identifying resource within repository 150
relationship data position data	a position of the relationship data within the resource containing it, e.g., the relationship data position may indicate a particular position within a hierarchy of XML tags within a XML document (such as /3/2/5).

[0059] Note that many relationship records 160 may not store data for each column depicted in Table 1. For example, a relationship record 160 corresponding to hard link and a weak link may store data in the target resource identifier column, but not for the target path column; on the other hand, a relationship record 160 corresponding to a symbolic link may store data in the target path column, but not for the target resource identifier column. As a result, the type of information stored in relationship records 160 depicted in Table 1 is merely illustrative.

CONFIGURING THE BEHAVIOR OF THE SYSTEM

[0060] The framework patent application discloses techniques for integrating logic through the use of "resource configurations." A resource configuration is a unit of logic that is associated with one or more resources within repository 150. Each resource configuration contains one or more configuration items that each defines and/or expresses one or more rules for managing a resource associated with a resource configuration.

[0061] Repository 150 interprets, evaluates, and/or analyzes a resource configuration to carry out the rules expressed therein. A resource configuration can be associated with one or more resources in various ways. For example, a resource configuration may be associated with resources that reside within a particular directory or that belong to a particular resource type. A resource that has been associated with a resource configuration is referred to herein as an associated resource.

[0062] Each time repository 150 performs an operation upon a resource, repository 150 carries out a rule specified in a resource configuration associated with the resource.

[0063] A resource configuration may be used to configure how relationship data is processed within repository 150. An exemplary schema for an element of a resource configuration is graphically depicted in FIG. 3. For example, portion 302 of the schema may be used to specify the type of link in which a relationship identified by relationship data should be modeled, e.g., a hard link, a weak link, a symbolic link, or not at all. Portion 304 of the schema may be used to specify how the location-display attribute is to be resolved when requested by client 110, e.g., if the PathFormat variable associated with a resource has a value of "identifier," then the location-display attribute is resolved upon request to the resource's identifier; on the other hand, if the PathFormat variable has a value of "path," then a resource is resolved upon request to the path, within repository 150, at which the resource resides.

[0064] The location-display attribute may be used in this way to avoid computing the path to a resource to avoid the overhead to do so when preparing information for transmittal to client 110. Thus, if so configured by the resource configuration, when a resource is retrieved from the repository 150, if the resource contains information that identifies another resource by resource identifier, then that information may be replaced with information that identifies the other resource by a path within the repository 150 at which the other resource resides. Alternately, if so configured by the resource configuration, when a resource is retrieved from the repository 150, if the resource contains information that identifies another resource by a path within the repository 150 at which the other resource resides, then that information may be replaced with information that identifies the other resource by a resource identifier.

[0065] As shown in portion 302 of the exemplary schema for a resource configuration, the resource configuration may be configured to cause system 100 to ignore relationship data contained within an associated resource. In this case, relationship records 160 are not created in persistent storage 140 when an associated source resource or a relationship-identifying resource, so configured, is stored within repository 150.

[0066] A resource configuration may specify that all associated resources exhibit a certain characteristic, e.g., all associated resources, which are source resources, shall be processed as a certain type of link (such as a symbolic link) regardless of the type of link identified in the relationship data.

[0067] In an embodiment, the configuration of the resource configuration may cause relationship data to be processed in a certain way depending upon where the relationship data is located. For example, certain types of resources may be assumed to have a certain type of link to the target resource identified within the relationship data. Further, depending on where the relationship data is located within a resource, the system 100 may make certain inferences about what type of link to make to the target resource identified within the relationship data. For example, if the relationship data is contained within the body of a web page, then the type of link identified in the corresponding relationship record 160 may be a weak link, regardless of what type of link is identified in the relationship data contained in the web page.

IMPLEMENTING MECHANISMS

[0068] In some embodiments, a client 110, a server 130, and persistent storage 140 may each be implemented on one or more computer systems. FIG 4 is a block diagram that illustrates a computer system 400 upon which an embodiment of the invention may be implemented. Computer system 400 includes a bus 402 or other communication mechanism for communicating information, and a processor 404 coupled with bus 402 for processing information. Computer system 400 also includes a main memory 406, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 402 for storing information and instructions to be executed by processor 404. Main memory 406 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 404. Computer system 400 further includes a read only memory (ROM) 408 or other static storage device coupled to bus 402 for storing static information and instructions for processor 404. A storage device 410, such as a magnetic disk or optical disk, is provided and coupled to bus 402 for storing information and instructions.

[0069] Computer system 400 may be coupled via bus 402 to a display 412, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 414, including alphanumeric and other keys, is coupled to bus 402 for communicating information and command selections to processor 404. Another type of user input device is cursor control 416, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 404 and for controlling cursor movement on display 412. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0070] The invention is related to the use of computer system 400 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 400 in response to processor 404 executing one or more sequences of one or more instructions contained in main memory 406. Such instructions may be read into main memory 406 from another machine-readable medium, such as storage device 410. Execution of the sequences of instructions contained in main memory 406 causes processor 404 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0071] The term "machine-readable medium" as used herein refers to any medium that participates in providing data that causes a machine to operation in a specific fashion. In an embodiment implemented using computer system 400, various machine-readable media are involved, for example, in providing instructions to processor 404 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 410. Volatile media includes dynamic memory, such as main memory 406. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 402. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications. All such media must be tangible to enable the instructions carried by the media to be detected by a physical mechanism that reads the instructions into a machine.

[0072] Common forms of machine-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0073] Various forms of machine-readable media may be involved in carrying one or more sequences of one or more instructions to processor 404 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 400 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 402. Bus 402 carries the data to main memory 406, from which processor 404 retrieves and executes the instructions. The instructions received by main memory 406 may optionally be stored on storage device 410 either before or after execution by processor 404.

[0074] Computer system 400 also includes a communication interface 418 coupled to bus 402. Communication interface 418 provides a two-way data communication coupling to a network link 420 that is connected to a local network 422. For example, communication interface 418 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 418 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 418 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0075] Network link 420 typically provides data communication through one or more networks to other data devices. For example, network link 420 may provide a connection through local network 422 to a host computer 424 or to data equipment operated by an Internet Service Provider (ISP) 426. ISP 426 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 428. Local network 422 and Internet 428 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 420 and through communication interface 418, which carry the digital data to and from computer system 400, are exemplary forms of carrier waves transporting the information.

[0076] Computer system 400 can send messages and receive data, including program code, through the network(s), network link 420 and communication interface 418. In the Internet example, a server 430 might transmit a requested code for an application program through Internet 428, ISP 426, local network 422 and communication interface 418.

[0077] The received code may be executed by processor 404 as it is received, and/or stored in storage device 410, or other non-volatile storage for later execution. In this manner, computer system 400 may obtain application code in the form of a carrier wave.

[0078] In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

The claims defining the invention are as follows:

1. A method comprising:
for each particular resource of a plurality of resources:
a server receiving a request to store each particular resource in a repository;
in response to said server receiving said request to store said particular resource in a repository:
said server parsing the particular resource to retrieve relationship data from within the particular resource, wherein said relationship data identifies a relationship between the particular resource and a second resource that is in the repository prior to the server parsing the particular resource;
based on the relationship data retrieved, said server determining a type of link to represent the relationship;
said server storing, within a database, one or more relationship records that identify said relationship using the type of link determined between said particular resource and said second resource, and
wherein said one or more relationship records are stored separate from said particular resource.

2. The method of Claim 1, wherein said type of link is one of: a hard link, a weak link, and a symbolic link.

3. The method of Claim 2, wherein said type of link is a hard link, and wherein said one or more relationship records identify said second resource by an object identifier assigned to said second resource by said repository.

4. The method of Claim 2, wherein said type of link is a weak link, and wherein said one or more relationship records identify said second resource by an object identifier assigned to said second resource by said repository.

5. The method of Claim 2, wherein said type of link is a symbolic link, and wherein said one or more relationship records identify said second resource by specifying a path, within said repository, at which said second resource resides.

6. The method of Claim 1, further including:

in response to receiving a request to delete said second resource from said repository, determining whether at least one resource, stored in said repository, has a first type of link to said second resource;

upon determining that said at least one resource, in said repository, has a first type of link to said second resource, denying said request to delete said second resource from said repository; and

upon determining that said at least one resource, in said repository, does not have a first type of link to said second resource, granting said request to delete said second resource from said repository.

7. The method of Claim 1, wherein the one or more records identify how said relationship is to be displayed.

8. The method of Claim 1, wherein said relationship is of a first type, and wherein the method further includes:

in response to storing, within said database, said one or more relationship records, incrementing a counter associated with said second resource, wherein said counter indicates how many resources, in said repository, have a relationship of the first type within said second resource.

9. The method of claim 1, wherein said type of link is hard link; and wherein the steps further include preventing deletion of said second resource while said type of link is hard link.

10. A method comprising:

a server storing, within a repository, a first resource and a second resource, wherein said first resource does not contain any link to said second resource;

after said server storing, within a repository, a first resource and a second resource, receiving a request to store a third resource in said repository;

in response to said server receiving said request to store a third resource in said repository:

said server parsing the third resource to retrieve relationship data, wherein said relationship data identifies a relationship between the first resource and said second resource;

wherein the first resource and the second resource exist in the repository before parsing the third resource;

said server determining a type of link to represent the relationship;

said server storing, within a database, one or more relationship records that indicate the existence of said type of link from said first resource to said second resource, and

wherein said one or more relationship records are stored separate from either of said first resource and said second resource.

11. The method of Claim 10, wherein said type of link is one of: a hard link, a weak link, and a symbolic link.

12. The method of Claim 11, wherein said type of link is a hard link, and wherein said one or more relationship records identify said second resource by an object identifier assigned to said second resource by said database.

13. The method of Claim 11, wherein said type of link is a weak link, and wherein said one or more relationship records identify said second resource by an object identifier assigned to said second resource by said database.

14. The method of Claim 11, wherein said type of link is a symbolic link, and wherein said one or more relationship records identify said second resource by specifying a path, within said repository, at which said second resource resides.

15. The method of Claim 10, the method further including:

in response to receiving a request to delete said second resource from said repository, determining whether at least one resource, stored in said repository, has a first type of link to said second resource;

upon determining that said at least one resource, in said repository, has a first type of link to said second resource, denying said request to delete said second resource from said repository; and

upon determining that said at least one resource, in said repository, does not have a first type of link to said second resource, granting said request to delete said second resource from said repository.

16. The method of Claim 10, wherein the one or more records identify how said relationship is to be displayed.

17. The method of Claim 10, wherein said relationship is of a first type, and the method further includes:

incrementing a counter associated with said second resource, wherein said counter indicates how many resources, in said repository, have a relationship of a first type with said second resource.

18. A non-transitory computer readable medium having stored thereon sequences of instructions which, when executed by one or more processors, undertake the steps of the method as defined in any one of claims 1 to 17.

19. A method for managing relationships between resources stored in a repository, substantially as hereinbefore described and illustrated in the accompanying drawings.

FIG. 1

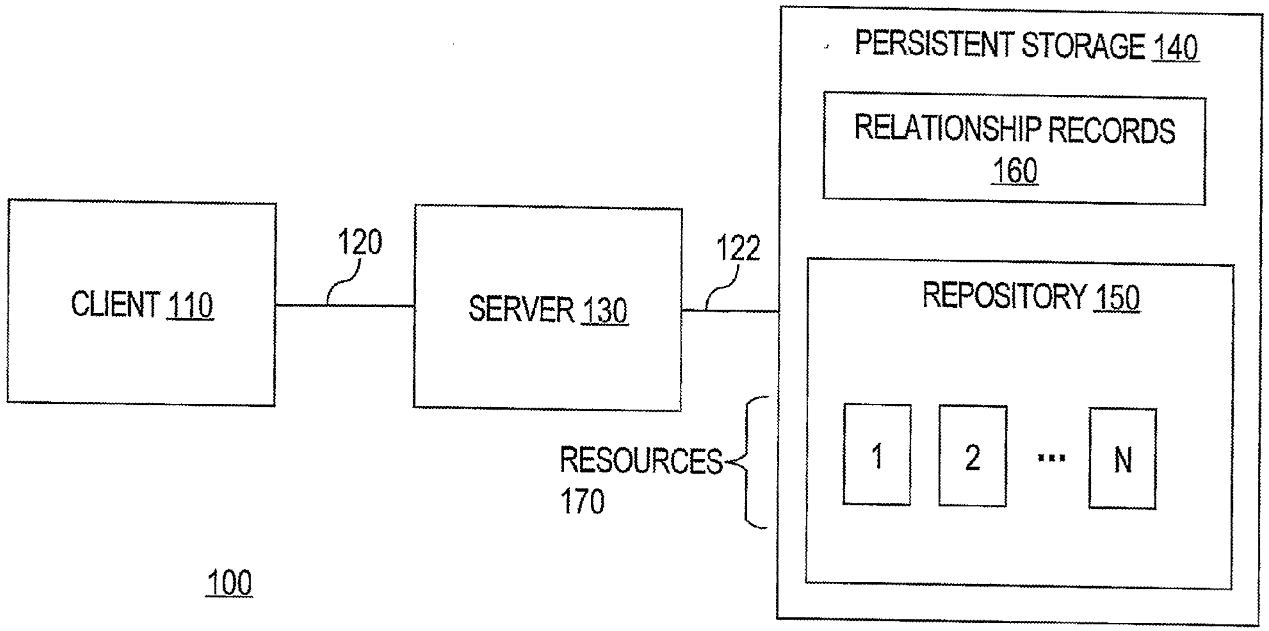

PERSISTENT STORAGE 140
RELATIONSHIP RECORDS 160
REPOSITORY 150
CLIENT 110
120
SERVER 130
122
1
2
...
N
RESOURCES 170
100

FIG. 2A

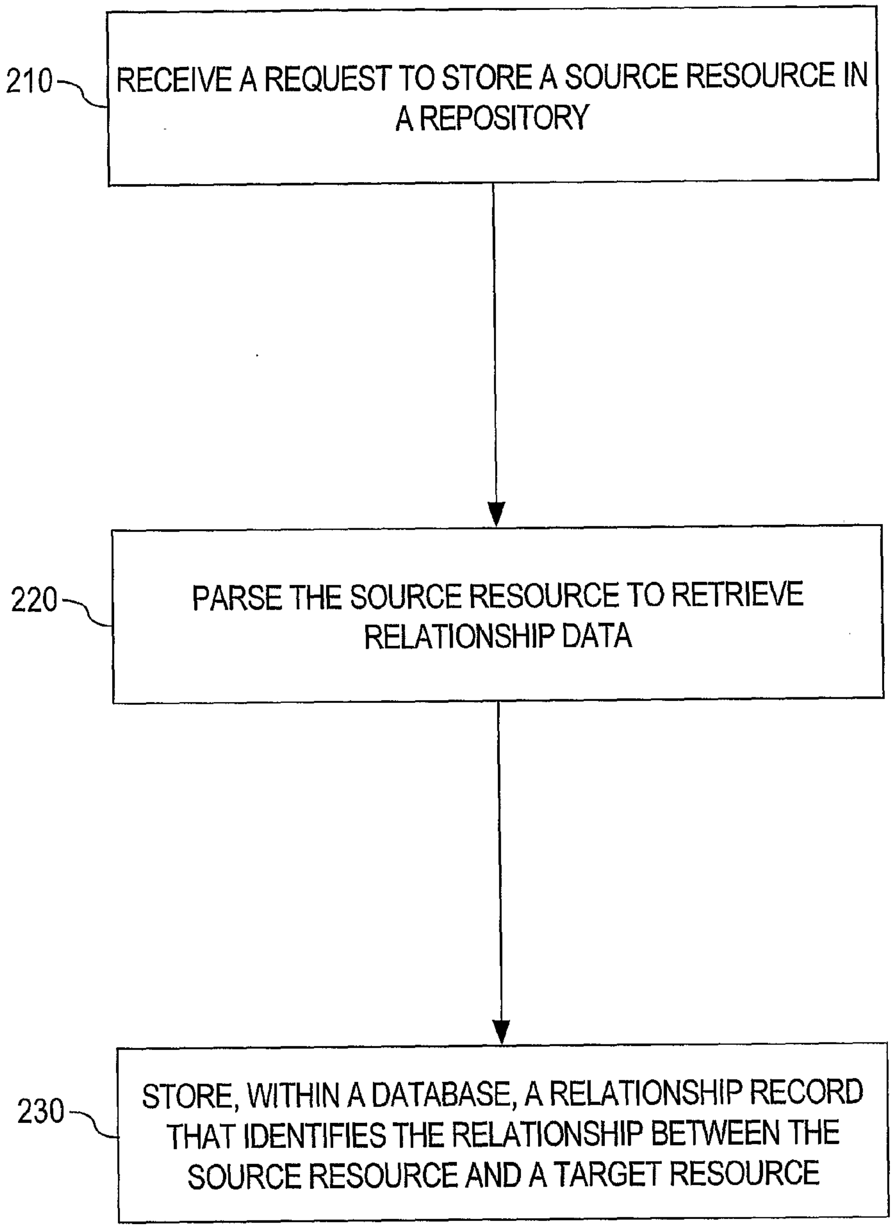

210
RECEIVE A REQUEST TO STORE A SOURCE RESOURCE IN A REPOSITORY
220
PARSE THE SOURCE RESOURCE TO RETRIEVE RELATIONSHIP DATA
230
STORE, WITHIN A DATABASE, A RELATIONSHIP RECORD THAT IDENTIFIES THE RELATIONSHIP BETWEEN THE SOURCE RESOURCE AND A TARGET RESOURCE

FIG. 2B

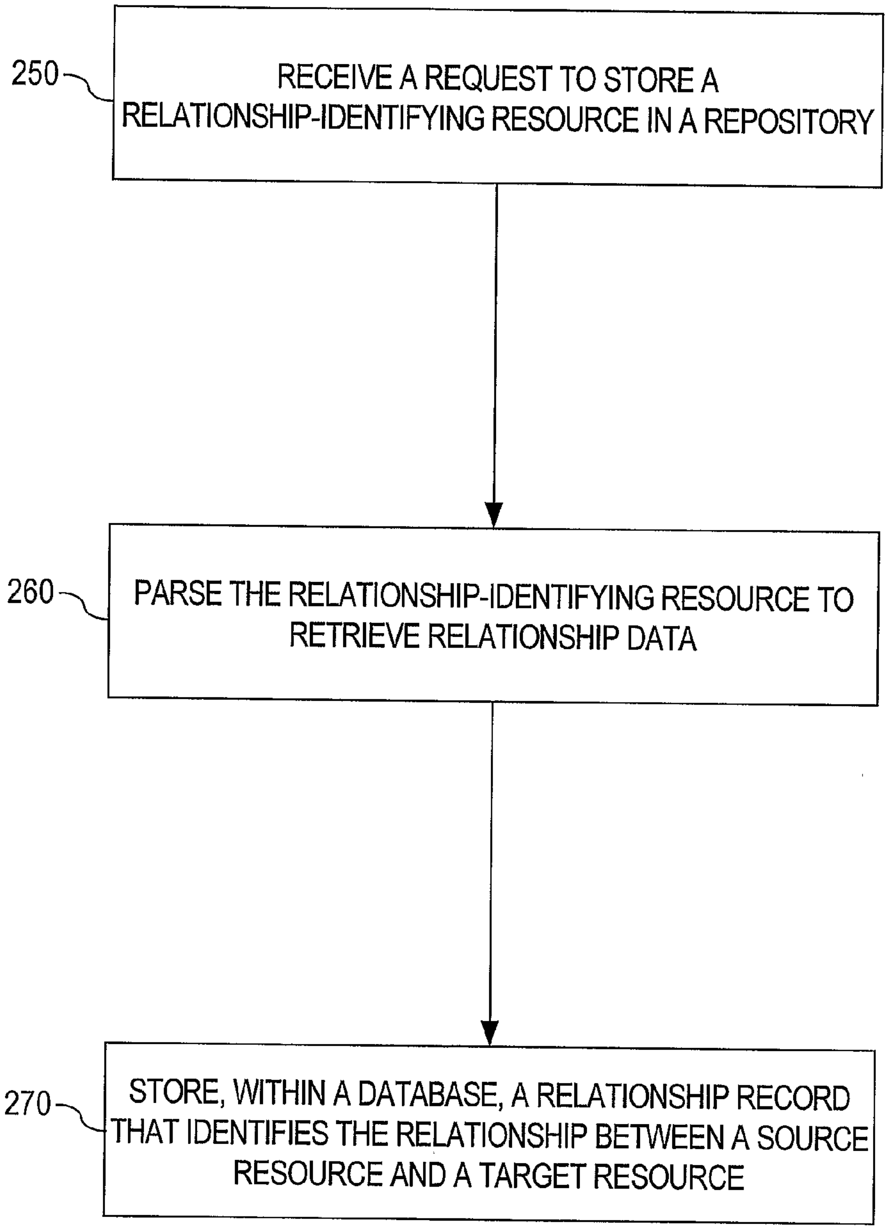

250
RECEIVE A REQUEST TO STORE A RELATIONSHIP-IDENTIFYING RESOURCE IN A REPOSITORY
260
PARSE THE RELATIONSHIP-IDENTIFYING RESOURCE TO RETRIEVE RELATIONSHIP DATA
270
STORE, WITHIN A DATABASE, A RELATIONSHIP RECORD THAT IDENTIFIES THE RELATIONSHIP BETWEEN A SOURCE RESOURCE AND A TARGET RESOURCE

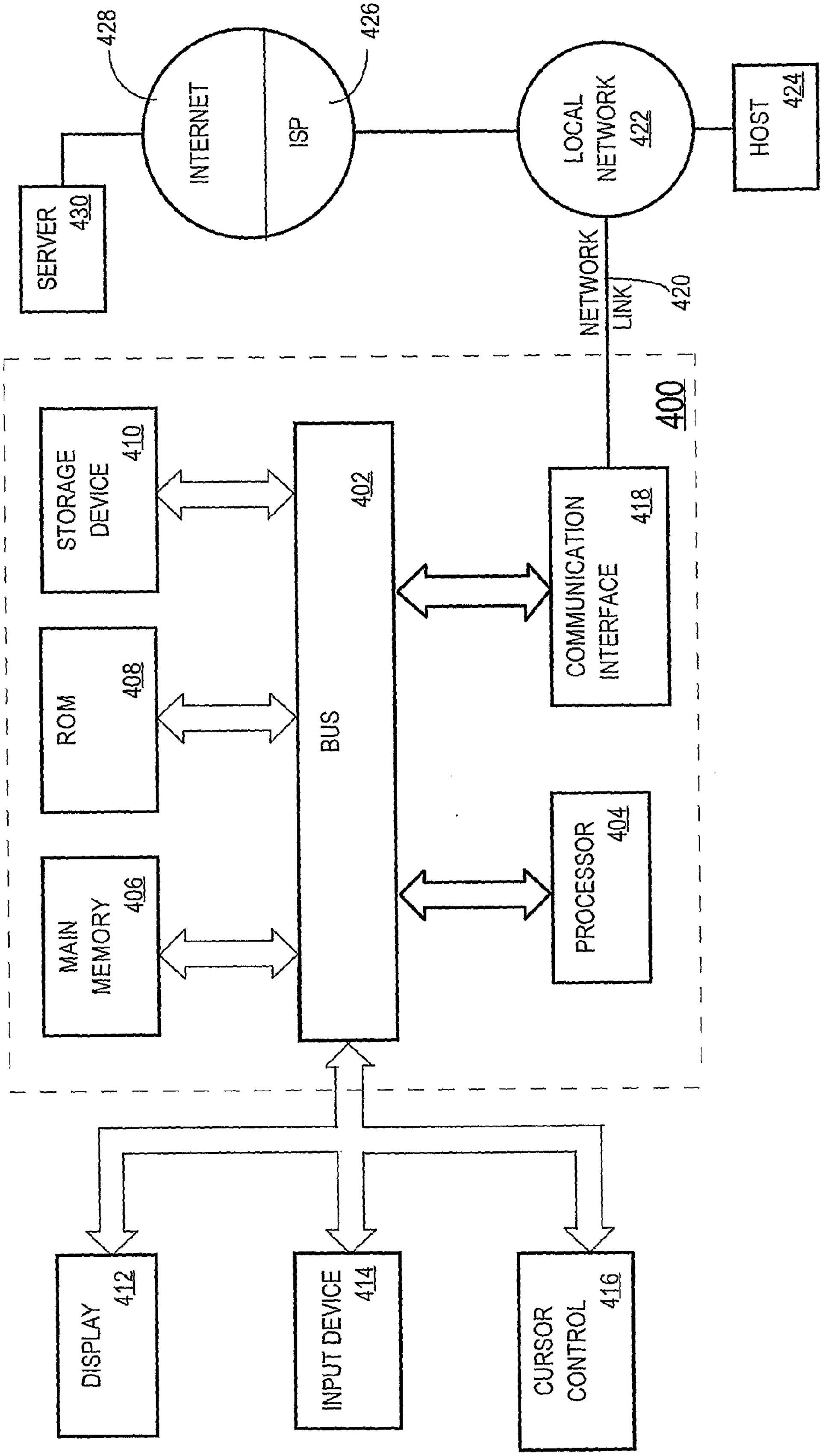
FIG. 3

Exemplary Schema of an Element of a Resource Configuration

```
<element name ="XLinkConfig">
 <complexType><sequence>
   <element name ="LinkType">
     <simpleType>
       <restriction base ="String">
         <enumeration value = "None"/>
         <enumeration value = "Hard"/>
         <enumeration value = "Weak"/>          302
         <enumeration value = "Symbolic"/>
       </restriction>
     </simpleType>
   </element>
   <element name ="PathFormat">
     <simpleType>
       <restriction base ="string">
         <enumeration value = "identifier"/>    304
         <enumeration value = "path"/>
       </restriction>
     </simpleType>
   </element>
</element>
```

300

FIG. 4


428
INTERNET
ISP
426
SERVER 430
LOCAL NETWORK 422
HOST 424
NETWORK LINK
420
400
STORAGE DEVICE 410
ROM 408
MAIN MEMORY 406
BUS 402
COMMUNICATION INTERFACE 418
PROCESSOR 404
DISPLAY 412
INPUT DEVICE 414
CURSOR CONTROL 416