Electronic Preprint for Journal of Information Processing Vol.20 No.3

Regular Paper

A Mobile Agent Approach for P2P-based Semantic File Retrieval

Naoki Fukuta[1,a)]

Received: September 15, 2011, Accepted: February 3, 2012

Abstract: Peer-to-peer (P2P) data sharing is a valuable approach for sharing data among people when they are belonging to different institutions. There are strong demands on both flexible, high-precision search and protection of privacy at peer-to-peer data retrievals. Especially, it is demanded for searching relevant files in P2P environment by using metadata while the terms in metadata that are used in such queries and annotations include some private information. In this paper, I present a mechanism and an analysis of P2P-based semantic file sharing and retrieval that uses mobile agents. The mechanism enables us to utilize private ontologies for flexible concept-oriented semantic searches without loss of privacy in processing semantic matching among private metadata of files and the requested semantic queries. The private ontologies are formed on a certain reference ontology with differential ontologies for personalization. In my approach, users can manage and annotate their files with their own private ontologies. Reference ontologies are used to find out semantically relevant files for the given queries that include semantic relations among existing files and the requested files. Mobile agent approach is applied for both implementing a system with less use of network bandwidth and coding it into a set of simple and small programs. I show the effectiveness of the use of private ontologies in metadata-based file retrieval. Also I show that the mobile agent approach has somewhat less overhead in execution time when the network latency is relatively high, while it is small enough even when the network is ideally fast.

Keywords: mobile agent, peer to peer, semantic retrieval, semantic matching, ontology

1. Introduction

P2P (Peer to Peer) approach is a way to share data among peers with little control of central servers. In P2P approach, data are mainly managed by each peer and also sufficient retrieval mechanisms are prepared to find out necessary data from peers [1]. For such purposes, DHT (Distributed Hash Table) and other mechanisms have been investigated for efficient and secure data retrieval [1]. However, there are further frontiers to improve such mechanisms when we need non-exact, flexible matching in the retrieval [2]. Furthermore, protecting unwanted reveals of secure data that includes protection of privacy for users is an important issue to be investigated [1], [2].

Mobile agent is an approach that makes each software agent capable to move from an execution environment to another when such environments are not on the same computer but connected via a certain network that might be sometimes disconnected. This approach is especially important to build P2P-based communication software. There are many approaches to implement better mobile agents and their platforms to realize faster execution and migration of agents, ease of programming, smaller footprint of agents and platforms, better security, etc [3], [4], [5].

P2P-based approaches are quite different from a traditional information sharing approach. For example, one traditional information sharing approach could be simply the use of a shared file server which stores common files for team members. Such file servers are often capable of preparing separate accounts for each member and to give proper access permissions for each file and folder. While file sharing policies should be shared with the project team members, it is also very difficult to force all members to follow the given policies when each member joined in many projects and each of which has different policies. Also in such cases, many members might not have enough time to learn and review all policies. Furthermore, when such policies have formed in an ad-hoc manner and they cannot be shared by all members, it makes it difficult to find out and access appropriate information, especially in the final phases of the project. This also makes it difficult to find out which data are deprecated, inconsistent, and being removed. Solving these issues may cause communication overheads that cannot be neglected [1].

Mobile agent-based approach is especially effective to be applied to build a software system that supports teaming tasks that may cover many organizations including companies, universities, communities, etc [6], [7], [8]. On such teaming tasks, better information sharing is crucial to reach better results in the tasks [9]. For example, there are many types of information to be shared in there. Their granularity is varied, from fine-grained low data files or messages to tacit knowledge that cannot easily be formed into a document that has explicit descriptions of them. There are already many researches about better information sharing in organizations [9], [10], [11], [12]. However, their early discussions were on a static organization whose members may have different

[1] Faculty of Informatics, Shizuoka University, Hamamatsu, Shizuoka 432–8011, Japan
[a)] fukuta@cs.inf.shizuoka.ac.jp

interests. Therefore, there are increasing discussions to share information among people who are in different organizations with different information sharing policies and their structures are dynamically changing [6], [13].

In Ref. [14], I discussed a preliminary idea, its possibility of implementation, and further issues on P2P-based file retrieval mechanism when each user has private (i.e., personalized) ontologies for their storage and annotation of the shared files. In this paper, I present a mechanism and an analysis of a P2P-based semantic file sharing and retrieval mechanism that enables us to use private ontologies for flexible concept-oriented semantic searches without loss of privacy in processing semantic matching among private metadata of files and the requested semantic queries. Mobile agent approach is applied for the purpose to implement a system with less use of network bandwidth and also to code it into simple programs. In this paper, I show the effectiveness of the use of private ontologies in metadata-based file retrieval. Also I show that the mobile agent approach has somewhat less overhead in execution time when the network latency is relatively high, while it is small enough even when the network is ideally fast.

2. Ontology-based Data Retrieval

2.1 P2P-based File Sharing and Retrieval

On P2P-based file sharing approaches, file management policies (e.g., the rules for naming files, where public files should be placed, etc.) are often published and managed by each primary owner who owns the entire file. Thus, such policies belong to each user (i.e., peer) and therefore they are made and managed on their own, and the files will be shared with those independently managed policies. On the other hand, P2P-based file sharing does not attempt to provide the shared files from primary owners. Some files will be provided by other peers which are located nearby the user who want to download them and had a piece of data that are stored by a distributed cache mechanism to efficient use of network bandwidth etc [*1]. Although there are many effective approaches to improve efficiency of transmitting data among peers, there are further needs to improve flexible data retrieval among peers that are managed by their own. Therefore, we sometimes use very primitive approaches to find out such files, e.g., by asking in a certain Web-based forum or via e-mails to obtain the exact file identifiers [1].

However, these primitive approaches do not always work sufficiently for sharing data among certain project members as mentioned before. Rather we need a flexible search mechanism that can find out sufficient data or files by showing their roles or partial information as hints.

2.2 Search Approaches with Semantic Metadata

In many text retrieval approaches, traditional indexing approaches have been used to allow us to find out all documents that include a certain set of terms. Such approaches often have been used for Web search engines and other local data storage software. On the other hand, in P2P-based file retrieval, sometimes it is difficult to make proper indexing for the files to be retrieved before a search query has been issued [15]. In this paper, since the discussions are about issues on P2P-based file retrieval, we consider a metadata-based semantic retrieval approach as an approach that does not need pre-indexing for the content of documents.

On retrieval approaches that use semantic metadata, search targets are not the files which include a certain keyword in their filenames or texts. Rather, they try to retrieve the files by the relations among the target files with other related files which are associated with certain conceptual backgrounds [16]. For example, this enables us to quickly find a file that contain the original data about a certain figure that is used in a specific file. Such retrieval can be realized by preparing sufficient semantic metadata for the files to be retrieved [16], [17]. Furthermore, such technologies can be extended for the use of service retrievals and automated compositions of them [18].

On P2P-based file retrievals, there are many granularities of targets, e.g., a peer, a set of files, and a part of file that might be encrypted [17]. While it is very difficult to cover all of them in a discussion within a single paper, I start the discussion with a case that seeks a set of files that can normally be managed by ordinary operating systems.

2.3 Ontology

Ontology based approaches are often discussed for a realization of flexible retrieval of data. In Gruber's definition, ontology is defined as an explicit process of conceptualization [19]. In recent works in the area, a simple taxonomy-like structure that captures conceptual hierarchies and relations is also considered as a (light-weight) ontology. Available ontologies can be found on the web by using Swoogle [20] and other ontology search services. Furthermore, conceptual networks in Wordnet can also be treated as an ontology. Also some automatically generated ontologies have been published and they were mainly generated from Wikipedia or other large-scale media that contain numerous instances (called *individual*s in the ontology research field) [21], [22]. Additionally, upper ontologies have also been proposed (e.g., SUMO [23], YATO [24], YAMATO [25], etc.). These ontologies have been used for defining certain business concepts and also for combining their business processes [26], [27]. Ontology description languages and frameworks have also been discussed and having growth in the use with OWL [28] and other description languages.

In this paper, ontologies that will have concepts, individuals, properties that denote relations among them are considered in the discussion. Since this construction is mostly equivalent to the OWL-DL framework [28], I omit some discussions about strict inference mechanisms and restrictions but rather focus on building a system that utilizes such ontologies and mappings among them effectively.

2.4 Personalized Ontologies: Their Uses and Issues

By associating semantic relations from ontologies to files, it is demanded to realize flexible semantic retrievals on searching files [16]. However, when such files are owned and maintained by their owners, the given annotations for such files and even the

*1 Modern P2P file sharing software, e.g., BitTorrent, and Winny have such a distributed cache mechanism.

used ontologies themselves should less or more depend on their owners' thoughts and policies [29].

Furthermore, there could be some annotations and associated part of ontologies to be kept secret from others even they are belonging to the same division of an institution and the entire files can be shared among them [14]. Therefore, such personal ontologies could be kept private, i.e., they cannot be openly accessed from others. For example, let us consider a file F is associated to a project X, and an annotation for the file F is associated to another secret project Y in the company. In this case, the file F might be shared to a member of X. However, the relations to project Y and any other information about the project Y itself should be kept secret from the members of X when they are not the employees of the company. There are demands to keep such metadata or the files be secret without complicated management of policies or permissions to such metadata and files while seeking an effective use of them for better retrievals [14], [29], [30], [31]. There are also demands about good implementations to realize such mechanisms efficiently and securely [14].

2.5 Assumptions

Domain dependent ontology development methodologies have been investigated in the domain ontology development field [32]. Also, investigations of sophisticated reusable ontology construction frameworks have been reported [33]. On the emergence of ontologies and their use, there are arising issues about making better mappings among such different ontologies and incorporating them into actual application systems [26], [34], [35].

The aim of this paper is to propose and show the potential advantages of a system that utilizes such ontologies and mappings. Before starting discussions about the system to be proposed in the next section, the underlying assumptions should be clarified. In this paper, the following assumptions has been applied:

Assumption 1) Any private ontology used in the proposed system has at least one connection to a publically accessible existing ontology, and the mappings among the two ontologies are defined.

Assumption 2) Any public ontology that is connected from a private ontology used in the proposed system, can reach any other public ontologies used in the system, via the connections among the ontologies, with the mappings among them.

Ontology mapping issues have been widely discussed [36] and various approaches have been proposed (e.g., clustering-based approach [37], logic-based approach [38], creating huge mappings [39], etc.). There are some series of international workshops [36] (e.g., EON2008), and several approaches have been investigated about calculating distances among similar concepts or ontologies themselves to map them each other [36], [37], [39], [40]. Furthermore, there are some approaches to define and describe effective mappings among concepts [37], [38], [40], [41]. SKOS is a good example for the purpose [41]. In SKOS, there are vocabularies to describe non-exact relations among concepts (e.g., `closeMatch`, `broadMatch`, `narrowmatch`, and `relatedMatch`), as well as exact match (`exactMatch`) of concepts. Furthermore, in SKOS, different labels can be defined for the same concept so that it enables us to separate labeling problems [42] from conceptualization problems.

The use of such techniques and vocabularies enables us to find out and also to describe relations among concepts in different ontologies [35]. According to these facts, having the two assumptions could be seen reasonable in at least the beginning of the discussion.

Furthermore, the following assumptions are applied in the paper.

Assumption 3) All files to be retrieved are sufficiently annotated by one or several private or publically accessible ontologies.

Assumption 4) All users of the proposed system have their own computing devices to run, and they are allowed to use the proposed system.

Assumption 5) All users of the proposed system can prepare appropriate access rights for the sharing files in some way (e.g., policy based, etc.).

Due to limited scope of the paper, further discussions about the all above assumptions are left for future work.

3. A Prototype System

3.1 Overview

I have implemented a prototype system that has some functions that can be useful to discuss about the issues presented in the earlier sessions. **Figure 1** shows an overview of the system.

The system realizes a P2P-based file sharing mechanism among same project members. A peer is assigned to each project member and the sharing policies are managed and maintained by each software agent that are associated with each peer. Thus, the peers can store and manage registered files to be shared with other members.

This system also allows users to make annotations for files and relations among files. For example, users can add a relation to a file to another file that denotes the file is an older version of the another file to prevent unexpected or unwanted sharing of older files, and denote two files as the one be the original figure and the other be the document that uses the figure to find out the original figures to modify them.

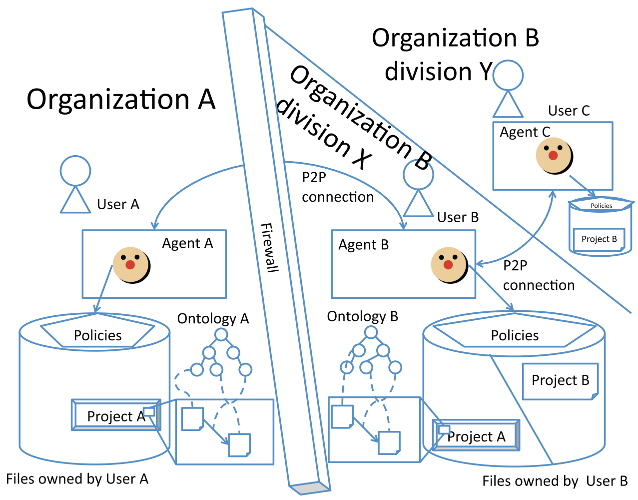

Fig. 1 An overview of prototype system.

3.2 Private Ontology

To make relations among concepts in different ontologies easily, the system prepared a reference ontology to relate such ontologies as a starting point of the discussion. Of course the reference ontology might not be shared to all users and multiple reference ontologies can be existed. However, in this case, at least one reference ontology is assumed to be associated to the personalized ontologies, and all reference ontologies are assumed to be accessed in open environment. This is a possible way to associate private ontologies by using one or several reference ontologies as a kind of scaffolding.

To this way, each peer has at least one reference ontology and associated private ontologies as differential ontologies to the reference ontology. Therefore, in this approach, each private ontology can be reproducible from the differential ontology and the associated reference ontology. This approach has another advantage that each associated software agent does not have to keep the whole reference ontology in them but just refers it when it needs. There are some big ontologies such as Wordnet. On the system, such ontologies are not fully loaded in each agent but just referred when necessary, and only some necessary parts of it is referred. In this model, the important part of the ontology is the ontology that denotes differences from its reference ontology.

3.3 File Retrieval and Transmission

Retrieval of files have been operated by the following way. First, the agent of a peer that wants to retrieve files issues a query as a mobile agent which has its differential ontology and moves itself to the peers which might have the target files. Then, the destination peer allows the agent to access reference ontologies and the agent can reproduce the complete ontologies from them. And then the agent will try to find out the files that are relevant to the query. The files found and permitted by the peer can be transmitted to the original peer and the obtained files are automatically associated to metadata that are inherit from the original one when it should be so (i.e., a metadata that denotes the file cannot be shared to other project members, it should be inherited to the transmitted files).

3.4 Implementation

The system has been implemented by using logic-based mobile agent platform *MiLog* [4]. The abstract structure of the implemented system is shown in **Fig. 2**.

Each peer has realized as an instance of *MiLog* execution platform. Each peer has Web-based user interface that has been implemented by *MiPage* Web API built on *MiLog*. The *MiLog* execution platform runs on each user's computer and it can be manipulated by the Web-based UI even when the computer has no connection to the network such as *google gear*.

Each peer has individual address and communicates to other peers based on the address. Here, the addresses for peers should be given when they have possibilities to share files each other. Accesses to the peers have been implemented by HTTP or HTTPS access and tunneling capability has also been implemented when peers are in the firewalls that prevent direct access to the peers via a relay server on the Internet [30].

Each peer has three types of agents. One is to control interactions to the users, another one is to control the whole behavior of the system, and the other is the agent that moves to another peer and retrieve files. When the system receives a request for retrieval, a clone agent is produced for the file retrieval and then

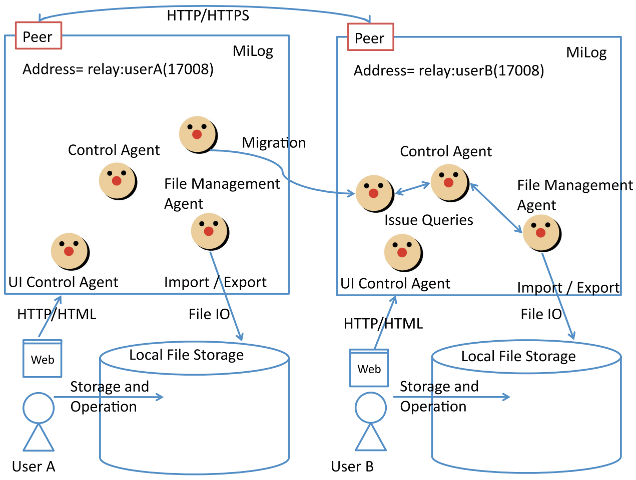

Fig. 2 The structure of the implemented system.

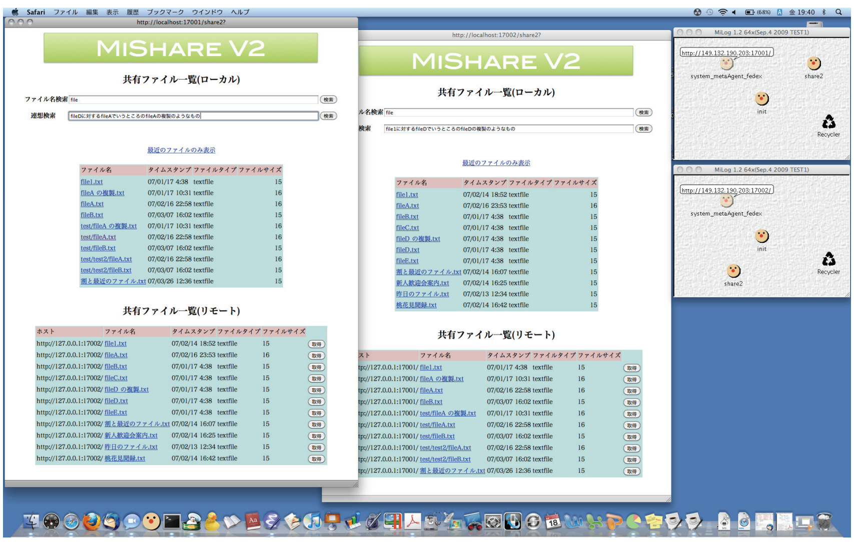

Fig. 3 An implemented web-based user interface.

moves to, and perhaps walks through some peers, and then back to the peer with obtained files and their metadata.

Migration among peers has been implemented by the mobility function implemented in *MiLog* platform. *MiLog* has implemented a very higher class of mobility, *strong mobility*, that allows all agents to move to other execution platforms while preserving the ones' whole internal execution states, including stacks and local variables in the code. On the *MiLog* implementation, communications for agent migrations can be put on HTTP or HTTPS so they can easily be passed through proxy servers and tunneling through firewalls when there is a relay server that is sufficiently configured and has a global IP address [*2].

Figure 3 shows the web-based user interface implemented on the prototype system [*3]. Each browser window is associated to each peer and the window at right side is the monitoring window implemented in *MiLog* that indicates how many agents are in the platform and what they are doing. Since the user interface has built as Web-based one, it is possible to use with any software or extensions that can capture a part of Web pages, including the capability to convert a part of Web page into a small widget. Further implementation details are shown in Ref. [14].

4. Evaluation

This section describes an evaluation about the proposed system [*4]. First of all, I evaluated the performance of search precision. Especially, I evaluated how the private ontology raises up the precision of the search. Since it is difficult to change the scale of a real ontology to an arbitrary size, artificially generated ontologies were used in the evaluation. Here, I prepared artificial ontologies which sizes are similar to real ontologies. To keep generality and simplicity of experiments, each ontology was constructed only by concept hierarchy that contains n_c of children in each node, and has depth n_d. In the experiment, I prepared artificially generated ontologies by the combinations of parameter $n_c \in \{2, 5, 10\}$ and $n_d \in \{3, 4, 5, 6, 10, 13, 16\}$. Note that the number of concepts in SUMO is nearly equal to the case of depth and siblings $<n_c : n_d>$ be $<13 : 2>$, $<4 : 10>$, and $<6 : 5>$. Also in Wordnet case, it is equal to the case that uses the depth and siblings be $<16 : 2>$, $<5 : 10>$, and $<7 : 5>$. Also, private ontologies were generated from the base reference ontology by applying the specified number of edit operators $n_e \in \{3, 5, 10\}$. In the experiment, I only used the edit operator that replaces its superclass. Then, a specified number n_e of virtual documents were prepared which have some conceptual tags that were automatically assigned to them based on the reference ontology. In the experiment, the number of assigned tags for each document is at most n_t concepts. In each search process, a virtual document is selected for the search target and the search query is generated based on the tags assigned to the target document. To emulate natural annotations, the tags used for the query are randomly shifted to neighbor concepts in the user's private ontology. Also,

*2 Further extensions about this capability have been discussed in Ref. [30]

*3 Here, to easily capture the screenshot with multiple peers, these runtimes were running on the same computer. Of course they also work well when they are deployed in different computers connected by a certain network

*4 A preliminary analysis has been presented in Ref. [43].

Table 1 Parameters to generate ontologies.

n_d	$\{3, 4, 5, 6, 10, 13, 16\}$
n_c	$\{3, 5, 10\}$
n_e	$\{1, 2, 3, 4\}$
n_t	$\{10, 100, 1000\}$

Table 2 Search result differences with and without mobile agents.

-n_d-n_c-n_e-n_t	with mobile agents	without mobile agents
-10-2-5-2	1st	9th
-3-10-5-1	1st	6th
-3-10-5-2	1st	5th

the tags assigned to documents are randomly shifted to neighbor concepts in the document owner's private ontology. Therefore, in each trial of retrieving documents, both the query and the assigned tags are not the original ones. Then the system retrieves the target documents at the document owner's peer. The experiment compares the rank of target documents in the search results whether it is using private ontologies in search algorithm or not. To cover both sparse and dense document case, I prepared experiments with $n_f \in \{10, 100, 1000\}$ depending on the purpose of the experiments. The parameters for generating ontologies and virtual documents are summarized in **Table 1**.

The similarity between a document and the conceptual tags in a query is calculated in the following formula.

$$\frac{\sum^{qc_i \in Q} \max_{vc_j \in V} sim(qc_i, vc_j)}{|Q|}$$

Here, $qc_i \in Q$ means the conceptual tags in the query Q, $vc_j \in V$ are conceptual tags assigned to the documents V, and $|Q|$ is the number of tags in the query, respectively. Although there are several approaches to compute conceptual similarity $sim(c_1, c_2)$, to keep simplicity of the discussion, I use the depth difference i to the deepest shared upper concept. For instance, when c_j is a depth i of upper concept of c_{j+i}, $sim(c_j, c_{j+i}) = \gamma_i$, where γ_i was set to $\gamma_0 = 1, \gamma_1 = 0.75, \gamma_2 = 0.10, \gamma_i = 0 \ \ s.t. \ \ i > 2$.

Table 2 shows the cases which have differences of ranks of retrieved documents when the use of private ontologies is considered. In this experiment, $n_t = 10$ is used so that the conditions could be harder for the approach. However, in all cases the ranks are better when the use of private ontologies is considered in the retrieval. When the size of ontology is large, due to the limited number of operation to generate a private ontology, the difference to a private ontology is relatively small. Therefore, there is no difference in the experiment in such case. The experiment shows that the approach works well even when the prepared ontology is small and documents are sparse in the peer.

When the system uses mobile agents to retrieve documents, it may take some computational and transfer overheads compared to use simple remote querying. Therefore I compared overheads of using mobile agents in some situations. Note that the experiments have been done with a preliminary implementation that is not deeply optimized for better speed or lesser data transfer.

The experiment conditions are equal to the previous experiments. Here, I used $n_f \in \{100, 1000\}$ and $n_c = 5$, respectively. The computer used for the experiment was a laptop computer running on MacOSX 10.6.4, with 8 GB memory and 3.06 GHz dual-core processor. For each peer, 256 MB of memory was as-

Table 3 Overheads of file retrieval within single computer.

	n_f = 100	n_f = 1000
using mobile agents [msec]	264.15	1,976.57
using remote query [msec]	211.42	1,816.28

Table 4 Overheads of file retrieval through firewall.

(n_f = 1000)	through firewall [msec]	local [msec]
using mobile agents	1,972.68	1,976.57
using remote query	4,428.72	1,816.28

Table 5 Data transfers of file retrieval between different hosts.

(n_f = 1000)	sent [kbytes]	received [kbytes]
using mobile agents	21.88	20.05
using remote query	20.88	6.02

signed for the Java VM. For experiments throughout firewalls, a relay host was used. The relay host computer was geographically located in approximately 800 km from the experiment place.

Table 3 shows the comparison in time when two peers are locally communicated. Here, the shown values are the averages in 100 times of the executions. Here, the results in "remote query" do not use private ontology so that they show the cases when private ontologies are shared before querying and there are no computational overheads to retrieve the target documents with private ontologies. Therefore, the shown "remote-query" case is the ideal case when it does not use mobile agent-based retrieval. In the case of $n_f = 100$, mobile agent-based approach took 50 msec of overheads compared to the remote query case. However, in $n_f = 1000$, although the overhead was still there, it only took 160 msec in 1976 msec of the total execution time. It can be said that when the number of documents are not small, the overhead on it could be relatively small in the overall execution time.

Table 4 shows the result in the case that it requires communications through firewalls to a distanced host. Here, the number of documents is $n_f = 1000$ and the results were average values in 100 times of executions. Since the mobile agent approach is robust for network latency [4], the overhead was far less than that on the remote query approach and the value was almost equal to the case of local communication.

Table 5 shows the observed communication overheads in transferred data size on the previous experimental condition. Here, although the transferred data sizes were still larger in the mobile agent approach, it can be said that it is reasonably small.

5. Conclusion

In this paper, I presented the mechanism and its analysis of ontology-based file retrieval approach that uses mobile agents. I demonstrated that the use of private ontology was effective in metadata-based file retrieval. Furthermore, the overhead of mobile agent approach is rather small in response time when the communication has done through a firewall with relatively long distance/high latency networks and it is also small enough even in a local communication environment.

In this paper, the evaluation has been done on an artificially generated dataset. Presenting a thorough evaluation of the system on a certain real-life data is future work. Also, in this paper, it was assumed that the several assumptions could be applied on the use of the proposed system. Discussions and detailed analyses concerning such assumptions are left to future work.

Acknowledgments This work was partly supported by Grant of The Telecommunications Advancement Foundation, and Grant of Artificial Intelligence Research Promotion Foundation in Japan. The author would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

[1] Cudré-Mauroux, P., Budura, A., Hauswirth, M. and Aberer, K.: PicShark: Mitigating metadata scarcity through large-scale P2P collaboration, *The VLDB Journal*, Vol.17, No.6, pp.1371–1384 (2008).

[2] Karnstedt, M., Sattler, K.-U., Hauswirth, M. and Schmidt, R.: A DHT-based infrastructure for ad-hoc integration and querying of semantic data, *Proc. 2008 International Symposium on Database Engineering & Applications* (*IDEAS '08*), pp.19–28 (2008).

[3] White, J.E.: Mobile Agents Make a Network an Open Platform for Third-Party Developers, *IEEE Computer*, Vol.27, No.11, pp.89–90 (1994).

[4] Fukuta, N., Ito, T. and Shintani, T.: A Logic-based Framework for Mobile Intelligent Information Agents, *Poster Proc. 10th International World Wide Web Conference* (*WWW 10*), pp.58–59 (2001).

[5] Cho, K., Hayashi, H., Hattori, M., Ohsuga, A. and Honiden, S.: picoPlangent: An Intelligent Mobile Agent System for Ubiquitous Computing, *Lecture Notes in Artificial Intelligence*, Barley, M.W. and Kasabov, N. (Eds.), Vol.3371, pp.43–56, Springer-Verlag (2005).

[6] Yu, L. and Ramaswamy, S.: Mining CVS Repositories to Understand Open-Source Project Developer Roles, *Proc. 4th International Workshop on Mining Software Repositories* (*MSR2007*), p.8 (2007).

[7] Gamma, E.: Agile, open source, distributed, and on-time: Inside the eclipse development process, *Proc. 27th International Conference on Software Engineering* (*ICSE2005*), pp.4–4 (2005).

[8] Zhou, Y. and Davis, J.: Open source software reliability model: An empirical approach, *Proc. 5th Workshop on Open Source Software Engineering* (*5-WOSSE*), pp.1–6 (2005).

[9] Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A. and Cohen, M.D.: Intelligent information-sharing systems, *Comm. ACM*, Vol.30, No.5, pp.390–402 (1987).

[10] e Maqsood, M. and Javed, T.: Practicum in software project management: An endeavor to effective and pragmatic software project management education, *Proc. 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering* (*ESEC-FSE2007*), pp.471–480 (2007).

[11] Fox, T.L. and Spence, J.W.: The effect of decision style on the use of a project management tool: An empirical laboratory study, *SIGMIS Database*, Vol.36, No.2, pp.28–42 (2005).

[12] Grudin, J.: Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces, *Proc. 1988 ACM Conference on Computer-supported Cooperative Work* (*CSCW '88*), pp.85–93 (1988).

[13] Wang, X., Zhang, L., Xie, T., Anvik, J. and Sun, J.: An approach to detecting duplicate bug reports using natural language and execution information, *Proc. 30th International Conference on Software Engineering* (*ICSE2008*), pp.461–470 (2008).

[14] Fukuta, N.: A Mobile Agent Approach for Flexible Peer-to-Peer File Retrieval, *Proc. IEEE/ACIS International Conference on Computer and Information Science* (*ICIS2010*), pp.599–604 (online), DOI: 10.1109/ICIS.2010.83 (2010).

[15] Moulin, C. and Lai, C.: Issues in Semantic File Sharing, *Advances in Intelligent Web Mastering*, Wegrzyn-Wolska, K.M. and Szczepaniak, P.S. (Eds.), Vol.43, Springer, pp.242–247 (2007).

[16] Siebert, M., Smits, P., Sauermann, L. and Dangel, A.: Increasing Search Quality with the Semantic Desktop in Proposal Development, *Proc. International Conference on Practical Aspects on Knowledge Management* (*PAKM2006*), pp.279–290 (2006).

[17] Tran, N., Beydoun, G. and Low, G.: Design of a Peer-to-Peer Information Sharing MAS Using MOBMAS (Ontology-Centric Agent Oriented Methodology), *Advances in Information Systems Development*, Magyar, G., Knapp, G., Wojtkowski, W., Wojtkowski, W.G. and Zupancic, J. (Eds.), Vol.2, Springer, pp.63–76 (2007).

[18] Berners-Lee, T., Hendler, J. and Lassila, O.: The Semantic Web, *Scientific American*, pp.35–43 (2001).

[19] Gruber, T.R.: A translation approach to portable ontologies, *Knowledge Acquisition*, Vol.5, No.2, pp.199–220 (1993).

[20] Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y. and Kolari, P.: Finding and Ranking Knowledge on the Semantic Web, *Proc. 4th International*

Semantic Web Conference (*ISWC2005*), pp.156–170 (2005).
[21] Auer, S., Bizer, C., Lehmann, J., Kobilarov, G., Cyganiak, R. and Ives, Z.: DBpedia: A Nucleus for a Web of Open Data, *Proc. 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference* (*ISWC2007 + ASWC2007*), pp.722–735 (2007).
[22] Yamaguchi, T. and Morita, T.: Building up a Large Ontology from Wikipedia Japan with Infobox and Category Tree, *Proc. 3rd Interdisciplinary Ontology Meeting* (*InterOntology '10*), the East Building of Mita Campus, Keio University, Japan, pp.121–134 (2010).
[23] Niles, I. and Pease, A.: Towards a Standard Upper Ontology, *Proc. 2nd International Conference on Formal Ontology in Information Systems* (*FOIS-2001*), pp.2–9 (2001).
[24] Mizoguchi, R.: Yet Another Top-level Ontology: YATO, *Proc. 2nd Interdisciplinary Ontology Meeting* (*InterOntology '09*), pp.91–101 (2009).
[25] Mizoguchi, R.: YAMATO: Yet Another More Advanced Top-level Ontology, Draft as of December 16, 2010, available from ⟨http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/YAMATO101216.pdf⟩.
[26] de Bruijn, J., Fensel, D., Keller, U. and Lara, R.: Using the web service modeling ontology to enable semantic e-business, *Comm. ACM*, Vol.48, No.12, pp.43–47 (2005).
[27] Haller, A., Oren, E. and Kotinurmi, P.: An ontology for internal and external business processes, *Proc. 15th International Conference on World Wide Web* (*WWW2006*), pp.1055–1056 (2006).
[28] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Petel-Schneider, P.F. and Stein, L.A.: OWL Web Ontology Language –Reference–, W3C Recommendation 10 February 2004, available from ⟨http://www.w3.org/TR/owl-ref/⟩.
[29] Sauermann, L., Bernardi, A. and Dangel, A.: Overview and Outlook on the Semantic Desktop, *Proc. ISWC2005 International Workshop on The Semantic Desktop* (2005), available from ⟨www.gnowsis.org⟩.
[30] Yamaya, T., Shintani, T., Ozono, T., Hiraoka, Y., Hattori, H., Ito, T., Fukuta, N. and Umemura, K.: MiNet: Building Ad-hoc Peer-to-Peer Networks for information Sharing based on Mobile Agents, *Proc. 5th International Conference on Practical Aspects on Knowledge Management* (*PAKM2004*), pp.59–70 (2004).
[31] Kapanipathi, P., Anaya, J., Sheth, A., Slatkin, B. and Passant, A.: Privacy-Aware and Scalable Content Dissermination in Distributed Social Networks, *Proc. 10th International Semantic Web Conference* (*ISWC2011*)*-PartII*, pp.157–172 (2011).
[32] Morita, T., Fukuta, N., Izumi, N. and Yamaguchi, T.: DODDLE-OWL: Interactive Domain Ontology Development with Open Source Software in Java, *IEICE Trans. Inf. Syst.*, Vol.E91-D, No.4, pp.945–958 (2008).
[33] Borgo, S. and Masolo, C.: Foundational Choices in DOLCE, *Handbook on Ontologies -2nd edition-*, Staab, S. and Studer, R. (Eds.), Springer, pp.361–382 (2010).
[34] Zhdanova, A.V., Krummenacher, R., Henke, J. and Fensel, D.: Community-Driven Ontology Management: DERI Case Study, *Proc. 2005 IEEE/WIC/ACM International Conference on Web Intelligence* (*WI2005*), pp.73–79 (2005).
[35] Cummins, R. and O'Riordan, C.: Learning in a pairwise term-term proximity framework for information retrieval, *Proc. 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (*SIGIR2009*), pp.251–258 (2009).
[36] Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P. and Trojahn, C.: Ontology Alighment Evaluation Initiative: Six Years of Experience, *Journal on Data Semantics*, Vol.XV, pp.158–192 (2011).
[37] Duan, S., Fokoue, A., Srinivas, K. and Byrne, B.: A Clustering-Based Approach to Ontology Alignment, *Proc. 10th International Semantic Web Conference* (*ISWC2011*)*-PartI*, pp.146–161 (2011).
[38] Jimenez-Ruiz, E. and Grau, B.C.: LogMap: Logic-Based and Scalable Ontology Matching, *Proc. 10th International Semantic Web Conference* (*ISWC2011*)*-PartI*, pp.273–288 (2011).
[39] Hu, W., Chen, J., Zhang, H. and Qu, Y.: How Matchable Are Four Thousand Ontologies on the Semantic Web, *Proc. 8h Extended Semantic Web Conference* (*ESWC2011*)*-PartI*, pp.290–304 (2011).
[40] Caracciolo, C., Euzenat, J., Hollink, L., Ichise, R., Isaac, A., Malaise, V., Melicke, C., Pane, J., Shvaiko, P., Stuckenschmidt, H., Svab, O. and Svatek, V.: Results of the Ontology Alignment Evaluation Initiative 2008, *Proc. 3rd International Workshop on Ontology Matching*, pp.73–119 (2008).
[41] Miles, A. and Bechhofer, S.: SKOS SImple Knowledge Organization System –Reference–, W3C Recommendation 18 August 2009, available from ⟨http://www.w3.org/TR/skos-reference/⟩.
[42] Ell, B., Vrandecic, D. and Simperl, E.: Labels in the Web of Data, *Proc. 10th International Semantic Web Conference* (*ISWC2011*)*-PartI*, pp.162–176 (2011).
[43] Fukuta, N.: A Preliminary Empirical Analysis of Mobile Agent-Based P2P File Retrieval, *Studies in Computational Intelligence*, Lee, R. (Ed.), Vol.365, pp.143–154, Springer (online), DOI: 10.1007/978-3-642-21375-5_12 (2011).

Naoki Fukuta received his B.E. and M.E. from Nagoya Institute of Technology in 1997 and 1999 respectively. He received his Doctor of Engineering from Nagoya Institute of Technology in 2002. Since April 2002, he has been working as a research associate at Shizuoka University. Since April 2007, he has been working as an assistant professor. In 2012, he received the IPSJ Yamashita SIG Research Award. His main research interests include Mobile Agents, SemanticWeb, Konwledge-based Software Engineering, Logic Programming, Applications of Auction Mechanisms, and WWW-based Intelligent Systems. He is a member of ACM (Association for Computing Machinery), IEEE-CS (IEEE Computer Society), JSAI (Japanese Society for Artificial Intelligence), IPSJ (Information Processing Society of Japan), IEICE (Institute of Electronics, Information, and Communication Engineers), JSSST (Japan Society of Software Science and Technology), and ISSJ (Information Systems Society of Japan).