OKBook: Peer-to-Peer Community Formation

Xi Bai[1], Wamberto Vasconcelos[2], and Dave Robertson[1]

[1] School of Informatics, University of Edinburgh, UK
[2] Department of Computer Science, University of Aberdeen, UK
xi.bai@ed.ac.uk, w.w.vasconcelos@abdn.ac.uk, dr@inf.ed.ac.uk

Abstract. Many systems exist for community formation in extensions of traditional Web environments but little work has been done for forming and maintaining communities in the more dynamic environments emerging from *ad hoc* and peer-to-peer networks. This paper proposes an approach for forming and evolving peer communities based on the sharing of choreography specifications (Interaction Models (IMs)). Two mechanisms for discovering IMs and collaborative peers are presented based on a meta-search engine and a dynamic peer grouping algorithm respectively. OKBook, a system allowing peers to publish, discover and subscribe or unsubscribe to IMs, has been implemented in accordance with our approach. For the meta-search engine, a strategy for integrating and re-ranking search results obtained from Semantic Web search engines is also described. This allows peers to discover IMs from their group members, thus reducing the burden on the meta-search engine. Our approach complies with principles of Linked Data and is capable of both contributing to and benefiting from the Web of data.

1 Introduction

Nowadays, providers care more about how communities (e.g. via eBay) rank their products and services than how search engine giants rank them. On the other hand, service requesters trust recommendations by other peers in social communities more than advertisements from service providers. Therefore, the community plays a more and more important role in addressing service discovery issues. Several systems exist for community formation in extensions of traditional Web environments but little is known about how communities might be formed and maintained in the more dynamic environments emerging from *ad hoc* and peer-to-peer networks. A difficulty in establishing communities in traditional peer-to-peer systems is that there is no structure that can be used as a basis for forming communities; there is nothing analogous to the relations used behind the scenes in Web-based social networking stems in order to infer community information (such as friend-of-a-friend relationships). Some recent peer-to-peer knowledge sharing systems have, however, used languages for specifying choreography between peers that can be used to provide the relations to build social networks. The OpenKnowledge[1] project has developed a peer-to-peer knowledge sharing system in

[1] http://www.openk.org/

L. Aroyo et al. (Eds.): ESWC 2010, Part II, LNCS 6089, pp. 106–120, 2010.

which peer interactions are described as Interaction Models (IMs, encoded in Lightweight Coordination Calculus (LCC) [1]). This paper proposes an approach for forming and evolving peer-to-peer communities based on their shared interactions with the assistance from the IM republication system [5]. Based on this approach we have developed OKBook - an open online platform for this new form of peer-to-peer communities.

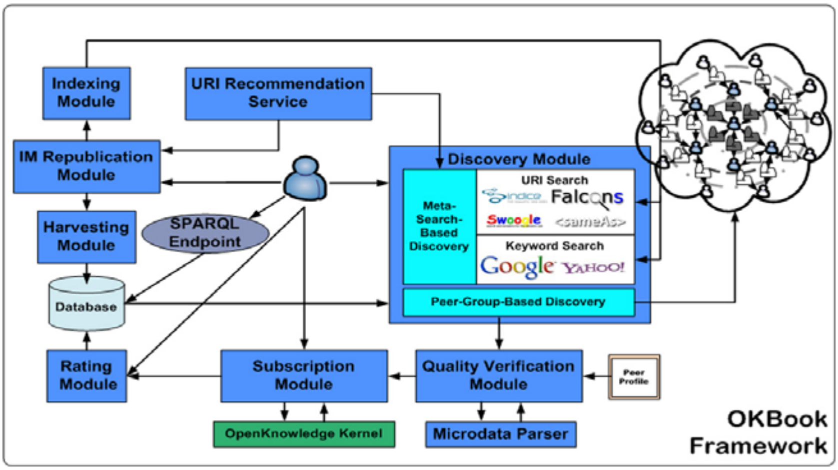


Fig. 1. Overlook of OKBook modules

Figure 1 shows the main components of the OKBook system. The *IM Republication Module* assists IM publishers in publishing IMs on Web pages using embedded micro-data such as RDFa [15]. Then the *Indexing Module* gets these IM pages indexed by informing search engine giants such as Google and Semantic Web Search Engines such as Sindice[2]. The *Discovery Module* offers a meta-search engine built on top of several SWSEs and ranks the search results by combining rankings from these search engines. Indices from Google and Yahoo!Search are used for assisting users in refining and adjusting their keyword-based queries when few URIs related to these queries are returned by SWSEs. Moreover, the *Discovery Module* also allows users to discover IMs that meet their requirements in terms of groups formed dynamically based on peers' historical interactions. Group members usually have some interests in common and this discovery mechanism can significantly cut down the IM search space compared with the meta-search-based mechanism. The *URI Recommendation Service* provides appropriate URIs for both the IM publisher within the annotation process and the IM requester within the IM querying process. The *Quality Verification Module* is in charge of discovering which role a peer is qualified to play by querying IM triples harvested by the micro-data parser as well as triples from the profile previously submitted by this peer. The *harvesting Module* gleans RDFa from the

[2] http://www.sindice.com

republished IM Web pages and stores triples into the back-end database. The *Group Discovery Module* analyzes the historical interactive data and discovers peer groups based on the criteria which will be discussed in Section 4. If a peer is qualified to play a role in an IM, it can subscribe to this IM through the *Subscription Module* which will display all roles this peer is capable of playing and subscription information will be finally forwarded to the OpenKnowledge Kernel (on a randomly selected peer) that coordinates this IM. After the execution, the *Subscription Module* forwards execution results from the OpenKnowledge Kernel to the *Rating Module* through which the peer can give some feedback about the IM based on its satisfaction with the execution. The ratings further inform IM selection beyond the ranking provided by the *Discovery Module.* Meanwhile, the execution result will be stored as one of historical interactions of this peer into the database. Moreover, a SPARQL endpoint is exposed and users can query and reuse triples (e.g., creating mashups) stored on OKBook via this endpoint so as to create more linked data and Web services.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction to the LCC choreography language used in OpenKnowledge and OKBook. Section 3 describes our way of representing knowledge such as capabilities of peers and links from IMs to the Web of data in OKBook. Section 4 proposes two mechanisms for discovering IMs helpful to users based on the meta-search engine and dynamically formed peer groups, respectively. The connection between the OKBook platform and the OpenKnowledge Kernel also is detailed in this section. Focusing on group-based IM discovery, Section 5 experiments with our approach and presents case studies that demonstrate how peers may acquire desired IMs from their group members, thus reducing the burden on the meta-search engine. Section 6 presents related work. Section 7 draws conclusions and outlines our future work.

2 Choreography Description on OKBook

OKBook uses the structure of interactions shared between peers to provide the basis for inferring social linkage. An interaction model is a specification of a peer interaction. In OKBook this is specified using the LCC language. LCC was developed in the OpenKnowledge project for describing choreographies for peer-to-peer networks. Compared with WS-CDL [11], LCC is a choreography description language but also executable. The LCC syntax is described elsewhere (e.g. in [1]) and a detailed definition of its semantics is not essential to this paper. For our purposes, all readers need know is that it is a process language that differentiates the specific roles taken by different peers in a given interaction and specifies the overall interaction through definitions (in the form of message sequences) of each role.

3 Knowledge Representation for OKBook

OKBook is an online platform built on top of the OpenKnowledge Kernel based on the Drupal framework. Through this platform, peers can import, publish, discover and subscribe/unsubcribe to IMs and join social groups which can help them to find out desired models for interactions and corresponding collaborators. The main modules of the system are described in this section.

3.1 Peer Capability Description and Its Storage

In the OpenKnowledge-based peer-to-peer community, each peer has a profile, in RDF, which will be uploaded to the discovery service as soon as this peer signs up to OKBook. A profile contains its owner's public information such as which roles it can play and proprietary information such as which constraints it can solve and corresponding methods in the form of OpenKnowledge Components (OKCs). OKBook can be regarded as a discovery server. In the peer-to-peer network, a discovery server behaves like a super peer and more than one of this type of peers may co-exist. When a peer registers on one of them, other discovery servers will also get copies of this peer's profile and the peer can log on to any of them using an identical username. Currently, OKBook supports the OpenID [18] standard which allows users to log on to different services with the same digital identity.

Centralized storage of triples is inefficient and incompliant with the peer-to-peer network for two reasons: firstly, it normally takes several hours to load in a billion RDF triples [16] so the centralized storage will be slow and impracticable; secondly, some data are proprietary and sensitive, which should be stored separately when copyright and security issues are taken into considerations. OKBook stores peers' profiles in a distributed way. Since Peer profiles are more concerned with providing information that can benefit interactions, they comply with choreographies in the whole OpenKnowledge system.

3.2 Linking Elements of Interaction Models to the Web of Data

In [5], a micro-data-based approach for republishing IMs has been proposed and a prototype has been implemented. OKBook further improves this prototype and integrates it as a module aimed at linking IMs to the Web of Data. As described in [5], currently IMs are republished using RDFa on XHTML Web pages and this task is fulfilled by the *IM Republication Module* as shown in Figure 1.

Wikipedia is a worldwide encyclopedia which has 262 language editions and naturally provides a knowledge base represented by natural languages. DBpedia [2] extracts structured information from Wikipedia and can be taken as a knowledge base represented by structured data such as CSVs, RDF triples which link structured knowledge to the World Wide Web. DBpedia assigns *http://dbpedia.org/resource/Name* (*Name* is taken from Wikipedia's *http://en.wikipedia.org/wiki/Name* like URLs) like URIs to all entities that have been crawled from Wikipedia. A URI is used for identifying a specific entity so we can also make use of URIs generated by DBpedia to annotate elements in IMs. However, sometimes more than one URI indicate the same individual and they will introduce differently within the annotation process in picking up a suitable URI and within the discovery process in matching annotations against the peer' profile. In order to allow IM publishers to efficiently find appropriate URIs to annotate elements in IMs, we take advantage of the DBpedia Lookup service which employs Lucene's string-similarity-based ranking and relevance metric having been discussed in [3] to do the URI recommendation. As shown in Figure 1, the *URI Recommendation Service* wraps the above functionality. This IM republication adheres to the four principles of Linked Data because of the followings: URIs are used as annotations and they are dereferenceable HTTP URIs; employed URIs are curated by DBpedia and each of them can be provided with an RDF file and a human-readable Web page from Wikipedia; each IM finally becomes an RDF resource on the Web of data and it is also assigned with a URI linking to other URIs of annotations. The *Discovery Module* (discussed in Section 4) also uses this lookup service.

We employ the property *rdfs:comment* to allow publishers to add more human-readable details about published IMs. Sometimes, users want to use diverse vocabularies or even their own. Therefore, three types of search services are offered to users for selecting diverse URIs to annotate IMs. The first service wraps in SW search engines such as Sindice, Falcons[3], Swoogle[4], which crawl and index URIs from the Web continuously. The second service wraps in general search engines such as Google and Yahoo!Search, which can assist users in refining and adjusting their keyword-based queries when too few relevant URIs are returned by above SW search engines. The third service wraps in a co-references search engine sameAs[5], which can help users find equivalent URIs for a specific URI inputed by themselves.

When users publish new IMs, the embedded triples will be harvested automatically by the *Harvesting Module* as shown in Figure 1 and stored in the back-end database by OKBook using the ARC2 library[6]. Obtained triples will be also exposed to users via a SPARQL endpoint based on HTTP bindings. This gives an opportunity for others to reuse the RDF repository derived from republished IMs and establish their own applications of interest (e.g., IM mashups).

4 Discovery of Interaction Models and Collaborative Peers

IMs describe choreographies between peers as protocols which guide peers to interact with one another and achieve collaborations with different service providers. Peers collaborate by subscribing to an IM but finding an appropriate one for guiding peers' interaction is still a challenge. The keyword-based IM publication [4] limits IM discovery in the OpenKnowledge system. Therefore, we connect to the broader Semantic Web discovery effort by using URIs to republish IMs. Based on this strategy, two mechanisms for discovering desired IMs and collaborative peers are proposed in this section: meta-search-based discovery and peer-group-based.

[3] `http://iws.seu.edu.cn/services/falcons/`

[4] `http://swoogle.umbc.edu/`

[5] `http://www.sameas.org/`

[6] `http://arc.semsol.org`

The former mechanism is a generic solution for discovering IMs. The latter one discovers other collaborative peers which have common interests with the requester. The meta search works well especially when requesters' desired IMs are out of the scope of their groups' interests. The *Discovery Module* in Figure 1 is in charge of this IM discovery task.

4.1 Meta-search-Based Discovery

RDFa is a type of serialization for RDF so RDF triples parsed from the RDFa-embedded page can be indexed by SW search engines. OKBook provides a meta-search engine that allows users to input queries and access several SW search engines. When a user submits a new IM to OKBook, the submission will also trigger a request message to be sent to SW search engines. For example, Sindice supports the RPC ping API that is developed according to the specification of the Pingback [17] mechanism. So an IM submission or a submission of its revised version will ping Sindice for indexing or re-indexing this IM. On the OKBook query interface, users submit their queries to our meta-search engine by typing in several keywords. Normally, different SW search engines use different ranking mechanisms so ranking results are combined before being displayed to users. Suppose Q denotes a querying phrase that a user types into our meta-search engine and U denotes the URL minted for a specific IM indexed by one or more SW search engines. S_i denotes the ith SW search engine. The rank of this URL returned by S_i is denoted by $rank_{S_i}(U,Q)$. If S_k has not previously indexed U, then $rank_{S_k}(U,Q) = 0$. We take the weighted average of ranks of U returned by different search engines as the overall rank of U on our meta-search engine, which is calculated by the following equation:

$$rank(U,Q) = \frac{\sum_{i=0}^{N} \chi_i \times rank_{S_i}(U,Q)}{N} \qquad (\sum_{i=0}^{N} \chi_i = N)$$

Here, N denotes the overall number of search engines on which our meta-search engine is built. χ_i denotes the weight on the rank returned by S_i and it actually reflects the user's preference. χ_i is equal to one by default. However, if a user prefers some search engines over others, he or she can inform our meta-search engine of this by changing values of weights on the OKBook search panel. Using this equation, ranks of all URLs will be synthesized and shown to users, in descending order of relevance. If more than one URL has identical ranking value, they will be displayed together as a group on the result page. Nonetheless, their provenances will be retained in case users want to further explore relevant information (e.g., snippets of search results) provided by diverse SW search engines.

On the search panel of OKBook, the URI recommendation service having been used in the *IM Republication Module* is employed as well within the query suggestion process. When users type in keywords, the recommendation service will forward them to the DBpedia Lookup service which will calculate the similarities between the keywords and stored URIs and select out the most suitable URIs for users to refine their queries. Since we use the same recommendation service in both the *IM Republication Module* and the *Discovery Module*, it is more likely that both the IM publisher and the IM requester will choose the same URI for the same individual. Needless to say, this will benefit the IM discovery because under this circumstance (a single URI is used for both annotating and querying), OKBook can just do the precise match between annotations and refined queries without employing any ontology matching or reasoning algorithms.

On the other hand, if the IM publisher and the IM requester represent a single object using heterogenous URIs, we align them by employing the sameAs service which has collected predicates of co-reference such as *owl:sameAs*, *skos:exactMatch* and *skos:closeMatch* from diverse vocabularies. Many more sophisticated techniques have been proposed for ontology matching and these also could be used on the OKBook platform for assisting users in discovering more useful services. Discussion of this is outside the scope of this paper.

4.2 Peer-Group-Based Discovery

Our peer-to-peer community is established based on peers' historical interactions. As mentioned earlier, when enough peers fill roles inside an IM, the *Submitting Module* will forward the IM and relevant subscription information to a coordinator randomly selected by the OpenKnowledge Kernel. After executed, the IM and the original subscription information will be sent back to OKBook which maintains a table to record the subscription history for each registry peer in the database. Peer groups will be discovered by analyzing the interaction history on the fly. For instance, the authorized peer will know peers with which it has been involved in executions of IMs and what other IMs these peers have subscribed to. This will facilitate IM discovery and peers can find more IMs with which they may potentially interact. After letting OKBook analyze IMs in which its group members and itself have been involved, a peer can subscribe to IMs by claiming to play a specific role inside. This method is actually derived from the Friends-Of-A-Friend (FOAF) idea. If the $PeerA$ has interacted with the $PeerB$, it is possible that the $PeerA$ will be also interested in other interactions not having involved itself but have involved the $PeerB$ and vice versa. Therefore, we use this method to group peers and name it the Interactions-From-An-Interaction (IFAI) method. The algorithm for the peer-group-based discovery is described in Algorithm 1. It is notable that Algorithm 1 describes a situation in which just peers that $PeerAP$'s friends know directly are considered. Actually, peers are allowed to do a deeper search depending on their preferences by invoking our IFAI method, which is helpful especially for peers who newly registered and have not gotten many friends. This can be also achieved by making use of newly discovered friend peers and running Algorithm 1 recursively.

4.3 Subscription Information Submissions and Feedback

After an appropriate IM is discovered, peers can subscribe to it via the *Subscription Module* as shown in Figure 1. In the OpenKnowledge Kernel, the

Algorithm 1. IFAI Algorithm (single step)

```
Input: the URI of current authorized peer, apeer_uri and the historical
       interaction record, record.
Output: URIs of group members, fpeer_uris and URIs of IMs these members
        were involved in, im_uris.
begin
    IM_URIs = getInvolvingIMs (apeer_uri, record);
    for each im_uri $\in$ IM_URIs do
        partner_peer_uris = getInvolvedPeerURIs(im_exec);
        for each peer_uri $\in$ partner_peer_uris do
            if apeer_uri equals peer_uri then
                continue;
            else
                fpeers = fpeers $\cup$ {peer_uri};
                IM_URIs' = getInvolvingIMs(peer_uri, record);
                for each im_uri' $\in$ IM_URIs' do
                    if im_uri' $\in$ im_uris then
                        continue;
                    else
                        im_uris = im_uris $\cup$ {im_uri'};
end
```

Distributed Discovery Service (DDS) will finally send subscription-related data (such as which peer fills which role) and the subscribed IM itself to a randomly selected coordinator, which takes charge of bootstrapping the interaction and executing the LCC protocol. In order to comply with this submission of subscription information and the OpenKnowledge Kernel, when a peer decides to subscribe or unsubscribe to an IM, OKBook will record relevant information such as the peer's user account, the URL of the IM, the time when this subscription occurs as well as some auxiliary information such as the peer's contact details. Before IM submissions, OKBook also checks if each role in the submitted IM is filled by at least one peer.

5 Experiments

After the execution of an IM, each involved peer may have a chance to extend its group information by making friends with other involved peers. This interaction-based community expansion will encourage peers to interact with others and also make them benefit from these interactions. In this section we describe some preliminary experiments on peer community formation using our approach.

5.1 Acquiring IMs from Discovered Group Members

OKBook provides peers with not only interactions in which they were involved but also interactions in which their participants were involved. These interactions can be actually taken as expansion seeds via which peers are likely to interact with more others whom they are difficult if not possible to know only based the searching mechanism offered by Web sites (e.g., eBay). Suppose Alice bought a product from Bob via the trade IM described in Figure 2 via OKBook. This figure depicts an interaction in which a client purchases a product referenced by a product code from a shop using his or her credit card.

/* A client, *C*, sends out a message to a potential shop, *S*, in order to buy a product with a code *PC* using his or her credit card *CC*. Then *S* sends the receipt of the product back to *C*. When *S* receives the message from *C*, it checks if *CC* has enough credit to buy the product. If the credit is enough, *S* completes *C*'s order by generating a receipt and send it back to *C*. There is one peer playing the client role and may be more than one peer playing the shop role. */

```
r(client, initial, 1, 1)
r(shop, necessary, 1)

a(client(PC, CC), C)::
   buy(PC, CC)⇒ a(shop, S)← payby(CC)∧ lookup(S) then
   receipt(R)⇐ a(shop, S)

a(shop, S)::
   buy(PC, CC)⇐ a(client(_),C) then
   receipt(R)⇒ a(client(_), C)← enough_credit(CC, PC)∧
                                complete_order(PC, CC, R)
```

Fig. 2. Simple trade IM in LCC

Focused on this IM, the sequence diagram in Figure 3 shows that how the group-base IM discovery drives peer interactions. Suppose Bob is a retailer who bought the same product from another peer Carol (the original manufacturer of this product) using cash via another IM similar to the above one previously. By logging on to OKBook, Alice can reach and subscribe to the latter IM based on the automatically discovered peer groups. Therefore, when Alice intends to buy the same product next time, she has a chance to interact with other peers such as Carol via the latter IM instead of with Bob via the former IM. It is likely to happen that Alice will get a lower price this time. On the other hand, from the perspective of Carol, OKBook assists her in discovering a new customer. Once their interaction finishes, Alice's group will be enlarged by absorbing a new group member and a new IM.

We experimented with the IFAI algorithm by simulating the peer interactions. Firstly, 100,000 peers and 10 IMs were generated for our experiment. Since each interaction will involve at least two peers, we assumed each IM owns two roles and 80,000 interactions occurred in the end. In each interaction, two peers and one IM were randomly selected to play roles and model the interaction respectively. Moreover, a bidirectional link was generated to connect every two collaborative peers. Then we made each peer randomly select an IM to subscribe to and calculated how many peers can find desired IMs making use of our peer-group-based discovery rather than the assistance of the meta-search engine on OKBook.

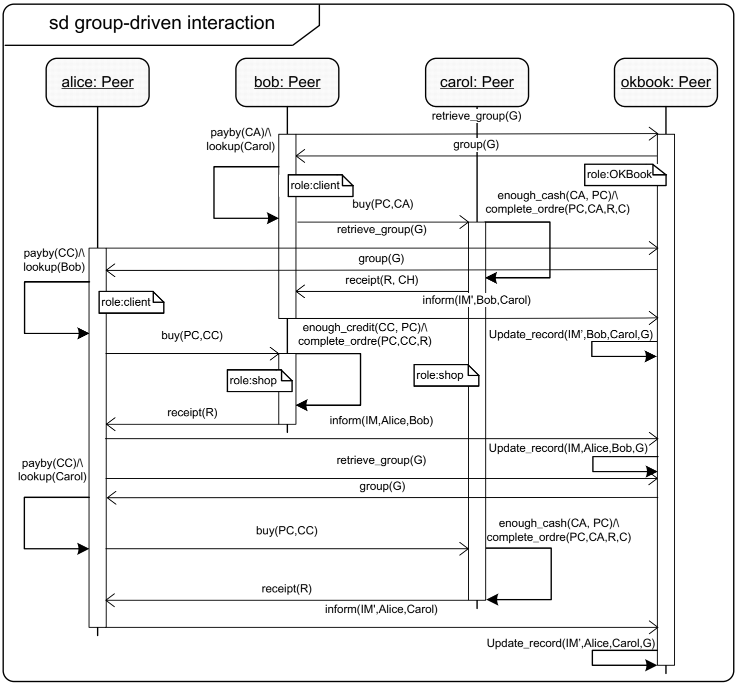
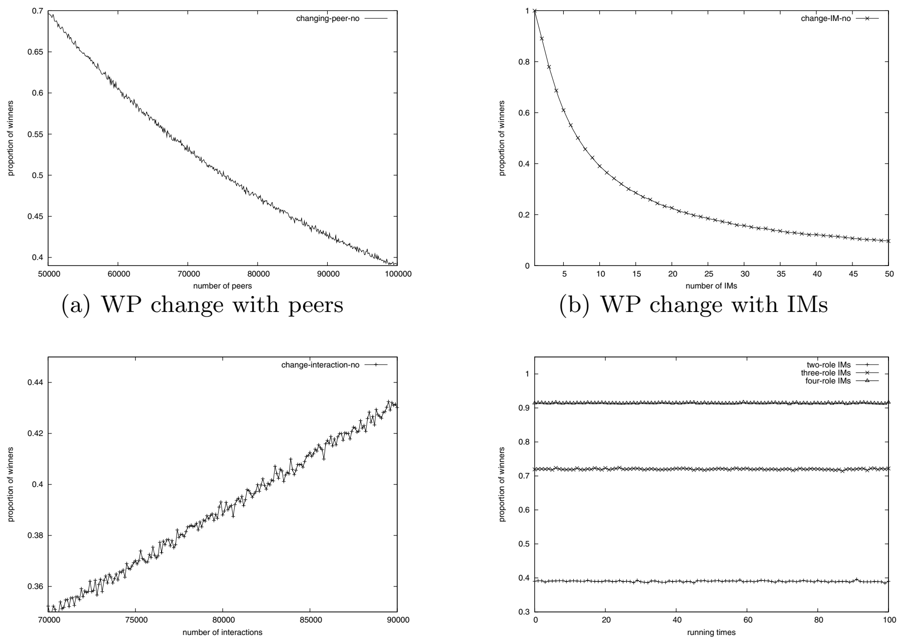


Fig. 3. Sequence diagram for a bunch of interactions driven by peer groups

We name the percentage of all peers the peers that can get desired IMs from their group members account for as the Winning Proportion (WP). Then we ran this experiment 1,000 times and found that on average, 38.99% of peers got desired IMs from their own group members without searching on the meta-search engine. The result indicates that peer groups can reduce the burden on OKBook as well as other SWSEs which have indexed republished IMs.

Secondly, since this proportion was probably related to the parameters such as the number of peers, the number of IMs and the number of interactions, we redid the above experiment by changing one parameter but keeping other two fixed and calculating the WP in each case. The results are depicted in Figure 4(a), Figure 4(b) and Figure 4(c). From them, we can see that although the augmentation of the whole peer-to-peer network is uncontrollable, peers can still try to employ as small numbers of IMs as possible to maximize the WP. Moreover, as time goes by, more and more interactions will occur in the peer-to-peer community and the WP will also go up in terms of our experimental results.

Thirdly, IMs may contain more than two roles and the WP value in reality is probably higher than the above one because under this circumstance, more group members and links may come up after each interaction. Base on the previous configuration of our experiment (100,000 peers, 10 IMs and 80,000



(a) WP change with peers

(b) WP change with IMs

(c) WP change with peer interactions

(d) WP change with IM roles

Fig. 4. Experimental results of the group-based IM discovery

interactions), we also investigated the performance of the peer-group-based discovery in the situations that interactions involved three peers or four peers. The discovery programme was ran for 200 times in each case and the result is shown in Figure 4(d). For the three-role case, on average, 72% of peers got desired IMs from their own group members and for the four-role case, this proportion went up to 91.43%. Therefore, in each interaction, if more peers are involved, more community members will gain in the near future.

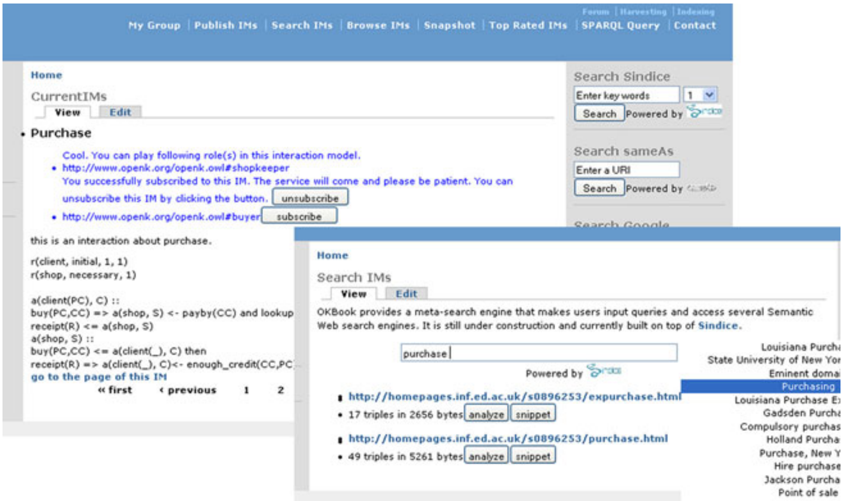
5.2 Peer Subscriptions and IM Consumptions

Conventionally, when a user accesses to a shopping Web site such as eBay, he or she searches a desired product by typing in relevant keywords via the search UI. But this is based on the precondition that providers have already logged on to this Web site and published adverts for this kind of products on the server. On the other hand, keyword-based search has its natural limitations due to the synonymity and the ambiguity of phrases. Employed as annotations, URIs can provide more disambiguous identifiers to concepts that convey meanings users want search services to be truly knowledgable about. In the peer-to-peer network, peers are more autonomous and there is no central server in this distributed environment. On the OKBook platform, all a user has to do is search for an appropriate IM (recommended by OKBook) and subscribe to it no matter if collaborative peers exist or not. Then OKBook will try to find other peers automatically who can collaborate with this peer and fulfill the interaction. Even if there is no provider providing the desired product for the time being, the subscription of this user will be still valid for a period of time (each subscription has an expiry time). As soon as enough collaborative peers have subscribed to an IM, this user will be informed of this and meanwhile, this IM goes into the execution procedure. However, for most conventional Web sites, this temporal“no result” is likely to end up with a page indicating the HTTP 404 error.

```
<html
xmlns=‘‘http://www.w3.org/1999/xhtml’’
xmlns:openk=‘‘http://homepages.inf.ed.ac.uk/s0896253/openk.owl#’’
xmlns:dbpedia=‘‘http://dbpedia.org/resource/’’
>
...
<div typeof=‘‘openk:InteractionModel’’>
 <span property=‘‘openk:has_declaration’’>r(client, initial, 1, 1)
 </span><br/>
 <span property=‘‘openk:has_declaration’’>r(shop, necessary, 1)</span>
 <div rel=‘‘openk:has_role’’>
  <div typeof=‘‘openk:Role’’ property=‘‘openk:has_roletype’’
  content=‘‘initial’’>a(<span property= ‘‘openk:has_name’’>client
  </span>(<span rel=‘‘openk:has_arg’’ typeof=‘‘openk:Argument
  dbpedia:Universal_Product_Code’’>
  <span property=‘‘openk:has_name’’>PC</span></span>), C)::<br/>
  <span rel=‘‘openk:sendout’’>
   <span typeof=‘‘openk:Message’’>
    <span property= ‘‘openk:has_name’’>buy</span>(
    <span rel=‘‘openk:has_arg’’>
     <span typeof=‘‘openk:Argument dbpedia:Universal_Product_Code’’>
      <span property=‘‘openk:has_name’’>PC</span></span>,
     <span typeof=‘‘openk:Argument dbpedia:Credit_card’’>
      <span property=‘‘openk:has_name’’>CC</span></span>)
...
```

Fig. 5. Republished trade IM in XHTML

Figure 5 gives the excerpt of proportional source codes of a Web page on which the trade IM described in Figure 2 has been republished. In Figure 6, OKBook helped a peer consume a Web page on which an IM has been republished and informed this peer of which roles it can play when the *Analyze* button was pushed. As shown at the bottom right of this figure, peers can also choose to make OKBook do the analysis as soon as they get the search result through the search panel. As mentioned previously, this analysis was based on the local profile of the peer as well as the harvested micro-data. With the movement of Linked Data, more and more peer-side applications will come up, which can parse and harness micro-data in a variety of ways but further discussion of this is outside the scope of this paper.



- Purchase

Cool. You can play following role(s) in this interaction model.

- http://www.openk.org/openk.owl#shopkeeper

You successfully subscribed to this IM. The service will come and please be patient. You can unsubscribe this IM by clicking the button.

- http://www.openk.org/openk.owl#buyer

this is an interaction about purchase.

```
r(client, initial, 1, 1)
r(shop, necessary, 1)

a(client(PC), C) ::
buy(PC,CC) => a(shop, S) <- payby(CC) and lookup
receipt(R) <= a(shop, S)
a(shop, S) ::
buy(PC,CC) <= a(client(_), C) then
receipt(R) => a(client(_), C)<- enough_credit(CC,PC
```

go to the page of this IM

OKBook provides a meta-search engine that makes users input queries and access several Semantic Web search engines. It is still under construction and currently built on top of Sindice.

- http://homepages.inf.ed.ac.uk/s0896253/expurchase.html
- 17 triples in 2656 bytes
- http://homepages.inf.ed.ac.uk/s0896253/purchase.html
- 49 triples in 5261 bytes

Fig. 6. Consumption of a republished IM

6 Related Work

From the perspective of orchestration, several approaches have been proposed for semantically describing Web services, such as OWL-S [6], WSDL-S [7] and SAWSDL [8]. Accompanying these, several matchmakers have been built, such as OWLS-MX [9] and SAWSDL-MX [10]. However, insufficient attention has been paid to semantically enhancing service descriptions from the perspective of choreography. WS-CDL [11] has been proposed for describing Web service choreography but it lacks appropriate support for semantics. Moreover, existing SWS choreography description languages such as WSMO [12] are heavyweight for portable devices such as mobile phones and personal digital assistants whose computation capabilities have been hampered by limitations of batteries and memories. OKBook employs LCC which is a lightweight and compact but expressive language for describing services provided by peers from the perspective of choreography.

Micro-data provides us with a more lightweight way of adding semantics to Web content for the purpose of making the content both human-readable and machine-readable. Several solutions for embedding micro-data into Web pages have been proposed. Davis proposed eRDF (embedded RDF) [13] from Talis, which can be used with any version of HTML but it restricts itself to the existing HTML attributes and does not support full RDF (e.g. there is no data type and no blank node). Microformat [14] also makes use of existing HTML and XHTML tags to convey metadata and other attributes but new formats require new data models. RDFa was proposed by Adida and Birbeck, which not only takes advantage of existing HTML attributes but also invents several new attributes in XHTML for flexibility and disambiguation. It supports full RDF and can reuse data models created for RDF. In OKBook, a micro-data-based strategy is employed within the process of IM republication and users are encouraged to use URIs maintained by DBpedia to annotate IMs. This strategy works for any RDF vocabularies because the micro-data we employ in OKBook is RDFa that supports the full RDF data model. When peers face IMs republished using diverse URIs indicating the identical individual, they can use the sameAs service from OKBook or any local ontology matchmakers to fulfill the alignment task.

Compared with other online services such as eBay and Facebook, OKBook can dynamically form peer groups which assist peers in discovering more IMs of interest and corresponding collaborative peers. In the distributed environment, there is no central peer and peers are more autonomous. Therefore, with OKBook, users only have to search and subscribe to desired IMs and leave the remaining work to OKBook and the OpenKnowledge Kernel.

7 Conclusions

This paper proposes an approach to forming peer communities based on peer interactions. A system has been created as a preliminary implementation of this approach built on top of the OpenKnowledge Kernel. This system provides peers with a platform for publishing, discovering and subscribing or unsubscribing to IMs. Within the publishing process, RDFa is used for annotating elements inside IMs that will be finally published on Web pages. Within the discovery process, two mechanisms are proposed based on our meta search engine and dynamic peer grouping algorithm respectively. In the established peer community, within the subscription process, peers are allowed to discuss on and add rates to IMs through modules such as forums and voting. Nowadays, more and more service providers begin to look for customers instead of waiting for customers to look for their services. So we believe that in the near future, we will not search for services but services will find us in one way or another (e.g., through the peer community). OKBook is a prototype trying to achieve this goal via the distributed knowledge sharing environment.

References

[1] Robertson, D.: Multi-agent coordination as distributed logic programming. In: Demoen, B., Lifschitz, V. (eds.) ICLP 2004. LNCS, vol. 3132, pp. 416–430. Springer, Heidelberg (2004)

[2] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a Web of open data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)

[3] Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Bizer, C., Lee, R.: Media meets Semantic Web - How the BBC uses DBpedia and Linked Data to make connections. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 723–737. Springer, Heidelberg (2009)

[4] Kotoulas, S., Siebes, R.: Adaptive routing in structured peer-to-peer overlays. In: Proceedings of the 3rd International IEEE Workshop on Collaborative Service-Oriented P2P Information Systems (WETICE 2007). IEEE Computer Society, Los Alamitos (2007)

[5] Bai, X., Robertson, D.: Service choreography meets the Web of data via microdata. In: Proceedings of the AAAI Spring Symposium on Linked Data Meets Artificial Intelligence (LINKEDAI 2010). AAAI Press, Menlo Park (2010)

[6] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K.: OWL-S: Semantic markup for Web Services. W3C Member Submission (2004), http://www.w3.org/Submission/OWL-S/

[7] Akkiraju, R.: Web service semantics-WSDL-S (Version 1.0) (2005), http://www.w3.org/Submission/WSDL-S/

[8] Farrell, J., Lausen, H.: Semantic annotations for WSDL and XML schema. W3C Recommendation (2007), http://www.w3.org/TR/2007/REC-sawsdl-20070828/

[9] Klusch, M., Fries, B., Sycara, K.: Automated semantic web service discovery with OWLS-MX. In: Proceedings of the Internatinoal Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2006), pp. 915–922. ACM Press, New York (2006)

[10] Klusch, M., Kapahnke, P.: Semantic web service selection with SAWSDL-MX. In: Proceedings of the International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR^2 2008) at ISWC 2008, pp. 3–18 (2008), CEUR-WS.org

[11] Kavantzas, N., Burdett, D., Ritzinger, G., Lafon, Y.: Web services choreography description language version 1.0. W3C Working Draft (2004), http://www.w3.org/TR/ws-cdl-10

[12] Lara, R., Roman, D., Polleres, A., Fensel, D.: A conceptual comparison of WSMO and OWL-S. In: Zhang, L.-J., Jeckle, M. (eds.) ECOWS 2004. LNCS, vol. 3250, pp. 254–269. Springer, Heidelberg (2004)

[13] Davis, I.: RDF in HTML (2005), http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml

[14] Suda, B.: Using microformats. O'Reilly Press, Sebastopol (2006)

[15] Adida, B., Birbeck, M.: RDFa primer (2008), http://www.w3.org/TR/xhtml-rdfa-primer/

[16] Large triple stores (2009), http://esw.w3.org/topic/LargeTripleStores

[17] Langridge, S., Hickson, I.: Pingback 1.0 (2002), http://hixie.ch/specs/pingback/pingback-1.0

[18] Recordon, D., Reed, D.: OpenID 2.0: a platform for user-centric identity management. In: Proceedings of the 2nd ACM Workshop on Digital Identity Management (DIM 2006), pp. 11–16. ACM Press, New York (2006)