

Users-Groups Matching in an Annotation System: Ontological and URL Relevance Measures

Danilo Avola, Paolo Bottoni, Amjad Hawash
Department of Computer Science, Sapienza University of Rome
Via Salaria 113, 00198, Rome, Italy
(avola, bottoni, hawash)@di.uniroma1.it

March 5, 2014

Abstract

The MADCOW annotation system supports a notion of group, facilitating focused annotations with respect to a domain. We argue that the use of an ontology to represent knowledge about the domain allows posting more refined annotations to a group, while the use of ontology concepts as tags facilitates the formulation of semantically significant queries for retrieving annotations on specific topics. Services for promoting participation to groups of potentially interested users can also leverage the adoption of domain ontologies, by matching tags users freely employed in their annotations to terms proper to some domain ontology. To this end, we propose a combination of existing relevance measures for matching users to domains.

Keywords: Web annotation, Ontologies, Matching.

1 Introduction

Digital annotation is the process of adding information to contents of a multimedia document, enriching it with additional valuable information, without altering the original content. Reasons for annotating can vary: to create mementos or clarifications of interesting content; to entertain discussions with other users; to construct or integrate documents [1, 2, 3]. On-demand training and education processes also use annotations, e.g. for learning foreign languages [4, 5, 6].

MADCOW (Multimedia Annotation of Digital Content Over the Web) is a three-tiered client/server architecture Web-based system for annotation of Web pages (see Fig. 1).

MADCOW supports the annotation of (portions of) text, image or video, with textual content, links to other resources, and user-defined tags [1].

Annotations can be published in MADCOW in one of three ways: (1) *public*: viewable by any user, (2) *private*: viewable by their submitters only, or

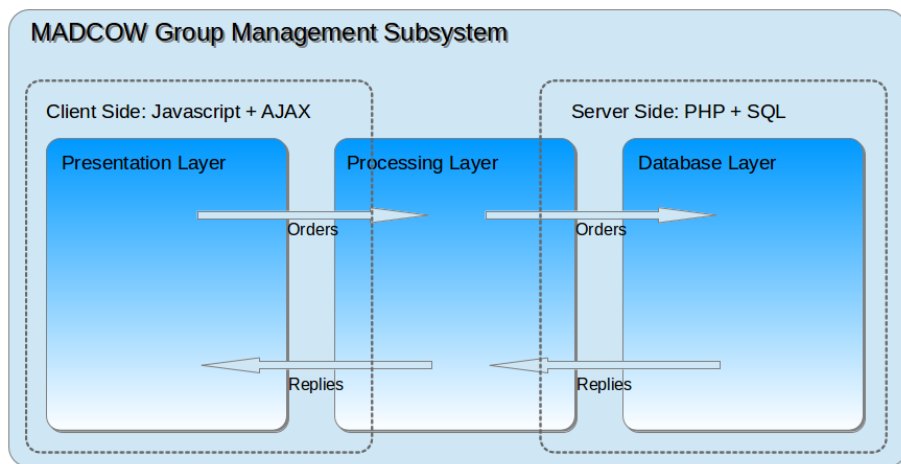


Fig. 1: MADCOW 3-tier Architecture.

(3) *group-related*: viewable only by members of the group to which they are posted. After selecting the Web content to be annotated, a pop-up window appears containing data related to the selected part and offering the possibility of adding some textual comment (the annotation), and a set of tags. In addition, it allows users to set the annotation type, the annotation visibility (private, public, or grouped), and it enables users to write down a list of groups' titles to which they can submit annotations (see Fig. 2).

Groups, introduced to MADCOW in [7], are typically focused on some topic of common interest and restrict publication of relevant annotations so that they can be read by all and only the group members, thus enhancing focusing and privacy of annotations, and supporting collaborative group activities.

Group owners deciding to invite users to their groups share similar, if symmetrical, problems with users looking for relevant groups. Owners want to attract users interested in the group topic, in order to promote collaboration, while users want to share their thoughts with people who can provide interesting feedback. In both cases, they do not want to lose time with users and annotations not relevant to their interests.

Prior to this work, the implementation of groups in the MADCOW system did not provide owners and users with automated support to identify potentially interesting matches for their interests, so that they had to manually search for them. Owners had to list all MADCOW users (possibly looking at the annotations they submitted publicly) and select some of them as receivers of invitations. This process becomes rapidly unwieldy as the number of users and annotations increases. In a similar way, users could refer only to scant information about the subject of the group, namely the title and a textual description of the group topic, as they cannot access the content of the annotations posted to the group.

In an experimental test, 152 students from eight different undergraduate

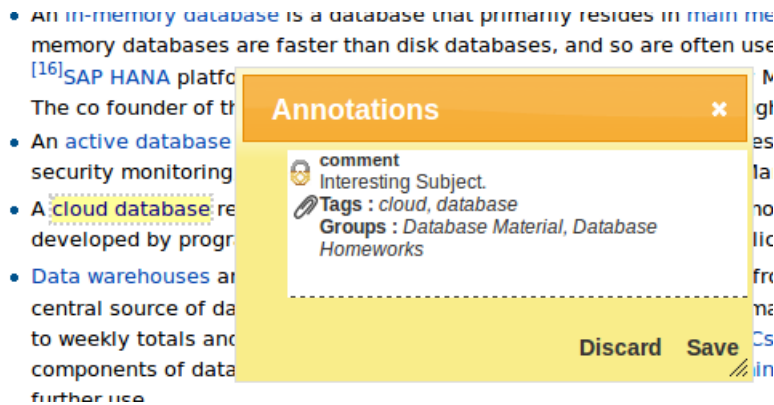


Fig. 2: Pop-up Annotation Window.

courses with their tutors were asked to use the system in order to check its feasibility and to measure a set of Human Computer Interaction metrics, such as *Understandability* of the different steps and *Ease of use* for the interface. Course tutors created different groups, all group-related services were tested, and results were gathered. Table 1 presents information about the number of times the different operations on groups were used and the average time needed to complete the operation. It is clear that manual invitation takes more time than any other operation.

Table 1: Number and average duration (secs) for operations.

	<i>Create</i>	<i>Update</i>	<i>Invite</i>	<i>Join</i>
# of times	72	51	719	125
Average	37.3	15.9	99.25	5.6

We argue that associating groups with publicly available representations of knowledge relevant to the group objectives, besides overcoming these problems, provides additional advantages to the whole annotation activity. Indeed, we rely on the selection of well defined terms in which to express domain knowledge [8], which can both be recognised as significant by users, and automatically searched for and manipulated by automated services [9, 10, 11, 12]. In particular, in the context of the Semantic Web, domain knowledge is typically represented in the form of ontologies expressed in XML-based languages, such as RDF [13] or OWL [14].

In this way, creators of groups become able to associate them with existing ontologies, by browsing among the available ones. The process can be assisted by letting creators provide sets of terms which reflect the intent of the group. The **MADCOW Ontology Browser** matches these terms with the collections of terms¹ associated with existing ontologies to propose those possibly appropriate for

¹From now and on we will refer to ontology classes or concepts as terms.

the new group. Group owners can ask the system to retrieve a list of ontologies ranked according to relevance for their groups and select the ontology which better reflects the group’s domain of interest.

Currently, the association between groups-ontologies is 1-1 from the group side and 1-n from the ontology side, i.e. a group can be associated with only one ontology, while an ontology can be assigned to more than one group. Fig. 3 represents the group-ontology association.

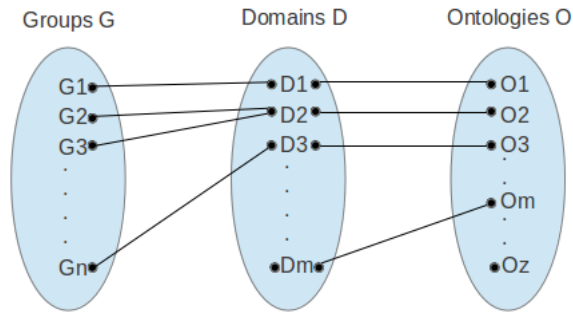


Fig. 3: Associations between groups, domains, and ontologies.

On the annotator’s side, users can tag their annotations with terms. A group member can select terms from the group ontology to better reflect the content of the annotation, facilitating subsequent retrieval. Users looking for groups can equally submit terms reflecting their interests to the **MADCOW Ontology Browser**, which will propose groups associated with matching ontologies. Similarly, owners can look for users whose public annotations contain terms in the group ontology.

A group owner can identify an ontology reflecting the main focus of the domain for which the group was formed. The tags in the set of annotations produced by a user, not yet in the group, can then be compared with the terms associated with the ontology, to extract a relevance measure, based on the **Class Match Measure (CMM)** [15].

Another relevance measure is based on the collection of Web sites a user targets for annotation. Users annotating specialised sites usually share the same interests. Discovering that some fellow annotators already belong in some group could encourage others to be members in that group. On the other hand, group owners can look for users to invite among those who more frequently annotate Web sites which are targets of annotations for group members.

In this paper we illustrate these concepts and describe the implementation of the **MADCOW Ontology Browser** and its relation with mechanisms for group support. After considering related work in Section 2, we introduce relevance measures for matching and discuss their use in Section 3, sketching the storage of ontologies and describing the behavioral aspect of matching in Section 4. Section 5 presents an applicative scenario, with experimental results shown in

Section 6, and Section 7 concludes the paper.

2 Related work

Ontologies were proposed as representational schemes for domain knowledge to enhance the retrieval process by Paralic and Kostjal [16], who compared the efficiency of retrieval in an ontology-based approach (implemented in the Webocrat system) with the vector and the latent semantic indexing models, obtaining promising results. In [17] a retrieval agent is described that provides access to information from multiple domains based on domain ontologies and users' interests.

Patel *et al.* explored the use of ontologies in order to automate common clinical tasks, e.g. selection of a patient cohort for clinical trials, by considering the matching of patient records to clinical trials as a problem of semantic retrieval [18]. In their approach, clinical trial criteria are formulated as queries to be matched against a knowledge base (represented as an ontology) to retrieve eligible patients.

The work in [19] aimed at designing an ontology mapping algorithm for effective product matching between heterogeneous classifications of products. In electronic commerce, each shopping mall has its own vocabulary and product hierarchy, increasing the semantic interoperability problem. Their primary goal is to minimize the number of retrieved products and to increase their relevance for customer searches.

Tangmunarunkit *et al.* match resources to application requests, comparing attributes advertised by resources with those required by jobs [20]. An ontology-based matchmaker performs resource selection by creating ontologies to declare resources and job requests and performing a semantic match between terms defined in the two ontologies. Similar to this, but in the area of human resource management, the work of [21] proposed an ontology-based approach to effectively match job seekers and job advertisements and applying a similarity-based approach to rank applicants.

Cantador *et al.* used representations of semantic user preferences for collaborative content retrieval, combining ontology-based user profiles to generate a shared semantic profile for a group of users [22]. They studied the feasibility of applying strategies for combining multiple individual preferences in a personalisation framework from a knowledge-based multimedia retrieval system. In their framework, user preferences are gathered in user profiles according to ontology concepts and used to retrieve ranked lists of items to be shown to users.

Several works deal with measuring the relevance of an ontology to a collection of terms. In [23], a similarity is defined between sets of concepts belonging to a common ontology. They defined a similarity between a single concept and a set of concepts and between two sets of concepts establishing some criteria that should be met, and implemented their algorithms in the Jena framework².

²<http://jena.sourceforge.net>

Sentence similarity based on semantic nets and corpus statistics is treated in [24], focusing directly on computing the similarity between very short – one sentence long – texts. Their algorithm took into account semantic information and word order information implied in the sentences. The semantic similarity of two sentences is calculated using information from a structured lexical database and from corpus statistics.

In [25], the similarity relevance was extended to include the relations between completely different resources. They proposed a methodology to measure semantic relevance between resources based on ontological representation. Their proposed measurement took into account the different meaningful relations and rules for its calculations.

Most of annotation tools do not support the grouping of related users into one unit, while those which support it do not present the whole set of group related services provided by the MADCOW system. Diigo Toolbar³ lets users create their own groups and invite other members, and provides some services for group archiving and groups dissemination, such as adding group links to users' websites and publishing group contents in blogs. A.nnotate⁴ takes a snapshot of a document, Web page or image, to produce a read-only copy, which can be annotated and shared with other users. This does not amount exactly to group annotation, but the tool implements it in a group-like fashion. In Bounce⁵, users produce, and collaborate on, notes for snapshots of Web pages. In Stickis, user groups implement the idea of private annotation submission. In Reframe It⁶, users can comment on a Web page via a sidebar and reply to others. Crocodoc⁷ does not implement groups, but users can invite others to collaborate in real-time by sharing folders.

MADCOW differs from such systems in that it uses group-related services to facilitate the annotation process as a whole and automates the group joining process minimising time and effort and maximising the relevance of the suggestions.

Users' interests can be measured by studying the Web sites they visit. In [26], the Web access patterns of more than 280,000 users were studied for a 3G mobile network, observing the users' interests in applications. The accessed URLs were classified into broad categories such as social networking, dating, music, gaming, trading, etc. We argue here that annotations on the same URLs by different users can also reflect commonality of their interests.

This paper develops previous work in [27, 28], where ontological based groups-users matching was introduced and portions of the MADCOW architecture relevant to group support were presented. We also presented a sketch of a set-theoretic formal definition of groups and of the actions involved in their management. Our contribution in this work is the implementation and the testing of the mechanisms needed to establish ontologies as a basis for user-group

³<http://www.diigo.com/>

⁴<http://a.nnotate.com/>

⁵<http://www.bounceapp.com/>

⁶<http://reframeit.com>

⁷<http://crocodoc.com>

matching, starting with the matching measurement techniques introduced in previous work. We also introduce another matching measurement (**URL-Based Matching**), whose results can be used as a further indicator in the user-group matching process.

3 Matching Measurements

Among the several measurements for evaluating the relation between a term and an ontology, we used and implemented two independent ones: ontology-based and URL-based.

3.1 Ontology-based measurement

The **Class Match Measure (CMM)** is meant to evaluate the coverage of an ontology for a given set of search terms by quantifying the relevance of these terms to the ontology itself. Relevance is calculated by performing binary matches (exact and partial) between each search term and each term in the ontology, and accumulating the results of these matches to obtain one single value for the whole process.

Definition 1 Let $C[o]$ be the set of terms for an ontology o , $T[u]$ the set of terms for an entity u ; then:

$$I(c, t) = \begin{cases} 1 & \text{if } c = t \\ 0 & \text{otherwise} \end{cases}$$

$$E(C[o], T[u]) = \sum_{c \in C[o]} \sum_{t \in T[u]} I(c, t)$$

$$J(c, t) = \begin{cases} 1 & \text{if } c \text{ contains } t \\ 0 & \text{otherwise} \end{cases}$$

$$P(C[o], T[u]) = \sum_{c \in C[o]} \sum_{t \in T[u]} J(c, t)$$

where $E(C[o], T[u])$ and $P(C[o], T[u])$ are the exact and partial matches between o and u terms, respectively. We then obtain $CMM(o, u) = \alpha E(C[o], T[u]) + \beta P(C[o], T[u])$.

So, for a given ontology o and two different sets of terms, t_1 and t_2 , $CMM(o, t_1) > CMM(o, t_2)$ indicates that the set t_1 is more relevant to the ontology o than the set t_2 .

Assuming that domains have been associated with proper ontologies, we introduce the following definitions in order to define three different relevance measures based on CMM.

1. $D = \{D_1, \dots, D_f\}$ is a set of domains with each D_i described by a set of terms, $\Delta_i = \{l_1^i, \dots, l_{n_i}^i\}$.
2. $T[G] = \{t_1, \dots, t_c\}$ is the set of terms for group G .

3. $U = \{u_1, \dots, u_m\}$ is a set of MADCOW users.
4. $TGP[u_i] = \{tgp_1^i, \dots, tgp_{q_i}^i\}$ and $TGV[u_i] = \{tgv_1^i, \dots, tgv_{s_i}^i\}$ are the sets of tags used in public, resp. private, annotations by user u_i , for $i = 1, \dots, m$.
5. $R = [r_1, \dots, r_k]$ is the array of all result matches, with $k = f$ or $k = m$, depending on whether we are matching towards domains or users.

We introduce three measures:

1. **Group-Domain Relevance:** This measure is used in the group-domain association process, to support groups' owners in identifying appropriate domains for their groups. The evaluation of the match between G and one of the domains in D , say D_i , requires checking the existence of a term from $T[G]$ in the set Δ_i , taking into account exact and partial matches. Fig. 4 depicts the group-ontology matching.

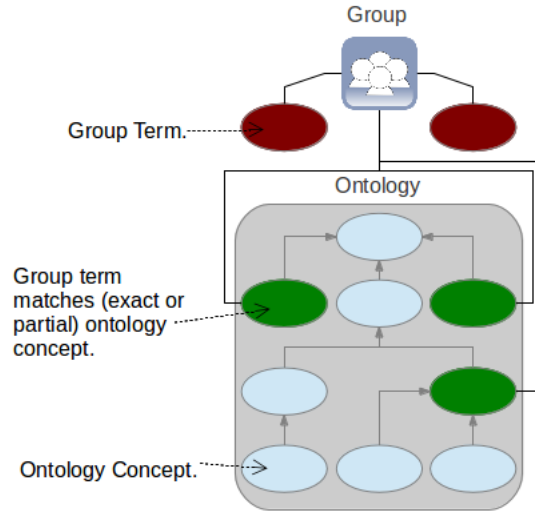


Fig. 4: Group-Ontology Matching.

The relevance of the domains in D for a group G is stored in R , where r_w represents the match between G and Δ_w , and it is calculated as⁸

$$CMM(\Delta_w, T[G]) = \alpha E(\Delta_w, T[G]) + \beta P(\Delta_w, T[G]).$$

2. **Domain-Users Relevance:** This measure supports groups' owners in identifying potential members for their groups. Matching a group G (associated with the domain D_i) to a user $u_w \in U$ requires checking the existence

⁸The coefficients α and β are determined here to be 0.6 and 0.4 respectively (thus giving more importance to exact matches).

in $\{l_1^i, \dots, l_{n_i}^i\}$ of a tag from $TGP[u_w]$ (as depicted in Fig. 5). To rank u_w 's relevance for G , we store in R the weighted sum of total and partial matches between Δ_i and $TGP[u_w]$, calculated as⁸ $CMM(\Delta_i, TGP[u_w]) = \alpha E(\Delta_i, TGP[u_w]) + \beta P(\Delta_i, TGP[u_w])$.

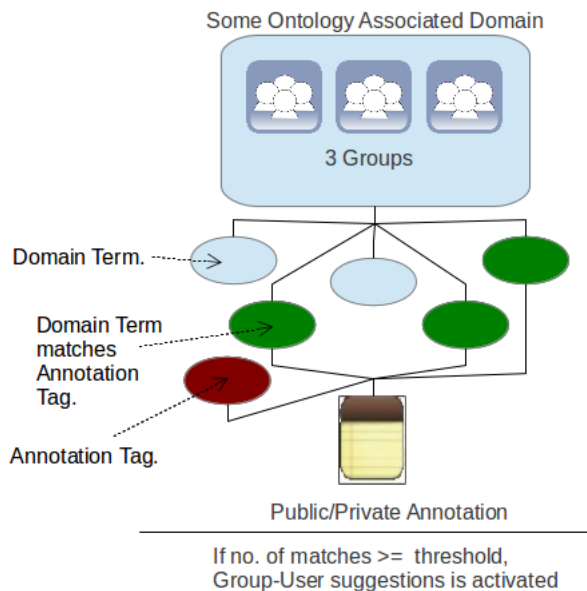


Fig. 5: Domain-User Matching.

3. **User-Domains Relevance:** This measure supports users in identifying relevant groups according to their interests. It is calculated in the same way as Domain-Users Relevance, but the matching now is executed between the sets $TGV[u_w]$ and $TGP[u_w]$ for all the domains in the system. In this case, elements in R represent the relevance for all domains in the system with the user u_w and each element equals $CMM(\Delta_i, TGP[u_w]) + CMM(\Delta_i, TGV[u_w])$, i.e.⁸ $\alpha E(\Delta_i, TGP[u_w]) + \beta P(\Delta_i, TGP[u_w]) + \alpha E(\Delta_i, TGV[u_w]) + \beta P(\Delta_i, TGV[u_w])$. All groups associated with the same domain have the same rank.

For the last two measures, the occurrences of tags in different annotations by the same user are counted, so that $I(c, t)$ and $J(c, t)$ in Definition 1 are multiplied by the number of occurrences of t .

The following pseudocode provides a representation of the process by which the relevance of ontologies from a given set is calculated for a group G represented by its terms⁹. We assume that `ontologies` is an array whose elements maintain the information on the different ontologies.

⁹The code uses nested loops to facilitate understanding. Loops are actually implemented through suitable SQL statements.

```

ln = ontologies.length; ranks = new array[ln] of int;
relevance = new array[ln] of float;
exactMatch = partialMatch = 0;
for(i = 0; i < ln; i++)
    relevance[i] = matchMeas(G.terms, ontologies[i].terms)
ranks = computeRanking(relevance);

float matchMeas(gTerms[], oTerms[]) {
for(i = 0; i < gTerms.length; i++)
for(j = 0; j < oTerms.length; j++)
    if(gTerms[i] == oTerms[j]) exactMatch++;
    elseif(isTokenIn(gTerms[i], oTerms[j])) partialMatch++;
return exactMatch*0.6 + partialMatch*0.4;
}

```

The following example illustrates the process of matching a group with two ontologies.

3.1.1 Example

Given the fragments of the **Animal** and **Plant** ontologies in Figs. 6 and 7, let G be a group with terms (**creature**, **plant**, **woody**, **vascular**). We need to check the relevance of the two ontologies for the group.

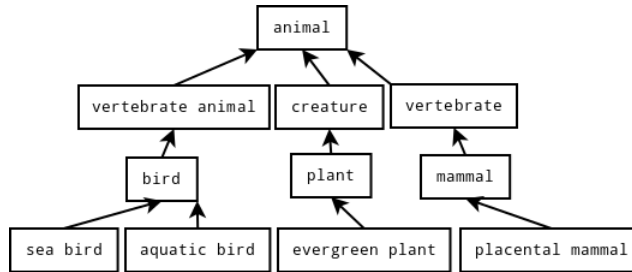


Fig. 6: A fragment of the **Animal** Ontology.

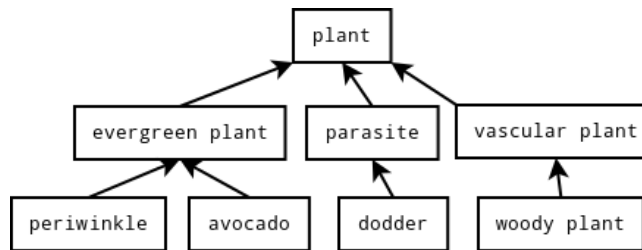


Fig. 7: A fragment of the **Plant** Ontology.

The Exact and Partial Match Measures for the sets of terms are given in Table 2, from which we derive: $CMM(G, Animal) = (1+1+0+0)*0.6+(0+1+0+0)*0.4 = 1.6$ and $CMM(G, Plant) = 0+1+0+0)*0.6+(0+3+1+1)*0.4 = 2.6$.

Table 2: Exact and Partial Matches
with Animal Ontology

	<i>creature</i>	<i>plant</i>	<i>woody</i>	<i>vascular</i>
Exact	1	1	0	0
Partial	0	1	0	0

with Plant Ontology

	<i>creature</i>	<i>plant</i>	<i>woody</i>	<i>vascular</i>
Exact	0	1	0	0
Partial	0	3	1	1

Based on these values, we can then rank the two ontologies:

$Relevance(G, Animal) = 1.6, Rank(Animal) = 2,$

$Relevance(G, Plant) = 2.6, Rank(Plant) = 1,$

with resulting array $R = [1.6, 2.6]$.

3.2 URL-based measurement

URL matching is based on the following considerations:

1. A group owner may be interested in users who annotate pages which are also annotated by group members, ranking them according to the number of annotations.
2. A user may be interested in groups whose members consistently annotate pages also annotated by him or her, ranking them according to the number of annotations.

For URL matching, the system excludes the file name from the URL, assuming that the path to the directory containing the page is still representative of the topic.

URL matching is used in both the processes of suggesting users to a group owner and of suggesting groups to a user. For the first case, given a group G and the set of MADCOW users $U = \{u_1, \dots, u_s\}$, let $URL_w = \{url_1^w, \dots, url_{n_w}^w\}$ be the set of URLs annotated by the user u_w , who is not member in G , with $A_i^w = \{a_1^i, \dots, a_k^i\}$ the set of public annotations for url_i^w by user w . Let $\Upsilon = \{v_1, \dots, v_m\}$ be the set of URLs annotated by all group members. Let $R = [r_1, \dots, r_s]$ be the array of URL-based relevances to group G for the users in U , with $r_w = \sum_{i=1}^{n_w} M(url_i^w, \Upsilon)$, and:

$$M(url_i^w, v_j) = \left\{ \begin{array}{ll} |A_i^w| & \text{if } url_i^w \in \Upsilon \\ 0 & \text{otherwise} \end{array} \right\}$$

In the process of suggesting groups to a user, the same matching is executed taking into account user's private and public annotations. In this case we consider all URLs which have been annotated by user u_w , irrespective of whether public or private annotations were used. Calculating user relevance depends on

$|A_i^w|$, the number of submitted annotations to a matched URL for a user, so that $r_i > r_j$ means that the first user submitted more annotations than the second one with respect to the matched URLs.

The URL matching process for suggesting users to a group owner is computed as follows⁹, assuming we have available the array `users` where the content of U is stored.

```

void executeURLMatch(group, int[] ranks) {
    usrNmb = users.length;
    relevance = new array[usrNmb] of int;
    groupURLs = loadURLs(group);
    for (i = 0; i < usrNmb; i++) {
        userURLs = users[i].loadUserURLs();
        for (j = 0; j < userURLs.length; j++)
            if (groupURLs.contains(userURLs[j]))
                relevance[i] += userURL[j].getNoOfPubAnns();
    }
    ranks = computeRanking(relevance);
}

```

In Fig. 8, suppose the MADCOW user is not a member in a group. The rounded squares represent the sets of URLs annotated by the user and the set of group members. Numbers inside URLs indicate how many annotations the user posted for URLs also annotated by group members. The sum of these numbers (in all shared URLs) represents how much this user's interests are close to those of the group members.

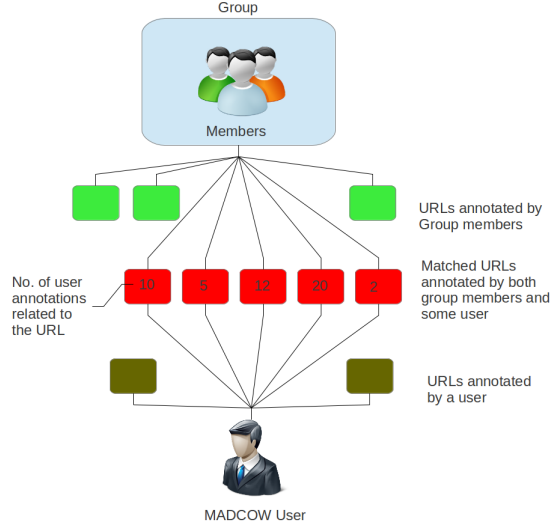


Fig. 8: URL Matching Illustration.

3.2.1 Example

Suppose we have 3 users who are not members in the group G . Every user has number of annotations in set of URLS as follows:

$user1 = \{ur11, ur12\}$ (with 7 and 15 annotations).

$user2 = \{ur11, ur14, ur16\}$ (with 3, 3, and 10).

$user3 = \{ur16, ur13\}$ (with 6 and 12).

and suppose members of G annotated the set $URL = \{ur11, ur12, ur13, ur14, ur15\}$. Then, we have:

$Relevance(user1, G) = 22, Rank(user1) = 1,$

$Relevance(user2, G) = 6, Rank(user2) = 3,$

$Relevance(user3, G) = 12, Rank(user3) = 2,$

with resulting array $R = [22, 6, 12]$.

3.3 Filters

The system excludes some users/groups from matching:

- **Users:** for a group owner requesting candidate members, the system excludes users who are already members in the group, those who have already been invited to join the group, or with pending join requests.
- **Groups:** for a user requesting candidate groups the system excludes groups owned by the user, those of which the user is a member, and those to which join requests were sent or from which invitations have been received.

4 Ontologies Repository and Behavioral Description

We describe the ontologies used in the testing sessions and illustrate the behavior of the system in executing the two ontological and URL based matches.

4.1 Ontologies Repository

We built a repository from 6 different ontologies: **Animals** (with 899 concepts), **Plants** (709 concepts), **Finance** (2037), **Artificial Intelligence** (2386), **Vehicles** (168) and **Viruses** (296) gathered from [30]. The ontologies are composed of a set of terms and IS_A relationships, with a text describing each term. Ontologies are represented via MySQL tables for faster access [31] and fast calculation of exact and partial matches. A fragment of the Entity-Relationship diagram for the repository is shown in Fig. 9.

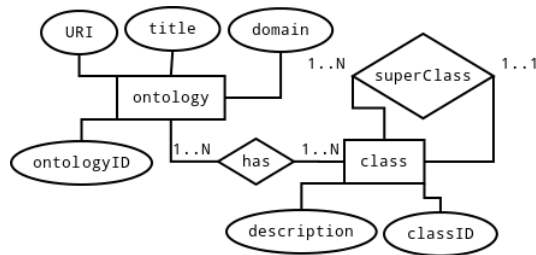


Fig. 9: A fragment of the Ontology Repository E-R Diagram.


4.2 Behavioral Description

Fig. 10 represents the behavioral aspect of the matching process for CMM and URL matches. The actions begin when the group’s owner issues a request `suggestMembers(group)` to the `pt1:MADCOWPortal` object, which responds by executing the functions `executeCMMMatch(group)` and `executeURLMatch(group)`. Both functions are executed upon the object `mch:MADCOWMatcher`.

When `executeCMMMatch(group)` is launched, `mch` contacts `dbc:MySQLServer`, launching the function `loadTerms(group)` that loads all terms for `group`. Then the execution of `loadUsersTags()` loads all tags for all public annotations for all users in the system, after executing a filtering process as described in 3.3. A list of tags is returned to `mch` and the function `executeMatch(terms, tags)` produces a ranked list of users, which is sent to the `pt1` object. In a similar way, the activation of `executeURLMatch(group)` causes `mch` to start `loadURLs(group)` that loads all URLs annotated by all members of the `group`. The `mch` object then executes the function `loadUsersURLs()` that loads all URLs annotated by all (filtered) users in the system. `mch` can then execute the function `executeMatch(groupURLs, usersURLs)` to produce a ranked list of users that is returned to the `pt1` object. Both lists of users are displayed to the group owner who can then select interesting users to whom to send invitations.

5 Working Scenario

In Fig. 11 the beta version of the MADCOW toolbar is shown, containing buttons for logging in and out of the system, reaching the system portal by the “Home” button, and controlling annotation creation and display.

A university uses MADCOW as a coordination tool for its faculty members and students. Taya is a teacher interested in several computing subjects, and she would like to direct her interest to a new field related to robotics and create a group to gather other interested members. She creates a group titled “Introduction to Robotics” and browses the existing domains by clicking the  icon in front of the title of her new group, but finds out that no suitable domain exists (see Fig. 12).

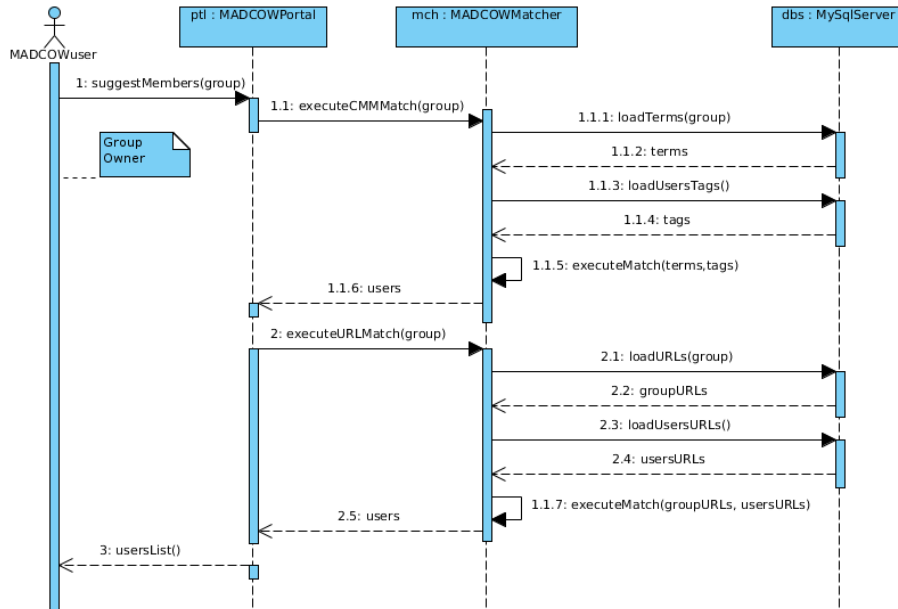


Fig. 10: CMM and URL Matches sequence diagram.

Hence, she enters relevant terms such as “robot”, “place”, “direction”, “motion” to represent the group intent and asks the system to suggest related ontologies for this group by clicking on “Search”. The system looks for the most suitable ontologies available and generates a ranked list. Taya picks the “AI” ontology and clicks on “Associate” to refer her group to this domain. The terms in the “AI” ontology become available for users to tag annotations (see Fig. 12).

Taya invites some users she knows interested in her group and who start submitting annotations to the group. After a while, Taya asks the system to suggest members for her new group by clicking the 🧑 icon in front of the group title. The system performs ontology and URL matching and shows Taya a ranked list of users from the two types of matches, from which she selects and invites users¹⁰, as shown in Fig. 13.

Conversely, Daniel has created several annotations related to different subjects and would like to find interesting groups. From his portal menu, he chooses “Suggest Groups”. The system looks at all his public and private annotations and matches them to all the available domains, presenting him with a ranked list of these domains with their associated groups. Fig. 14 illustrates this process.

¹⁰The absence of checkboxes before the users *tea* and *dando* in the URL suggestions list is due to their presence in the first list, preventing Taya from sending duplicate requests.

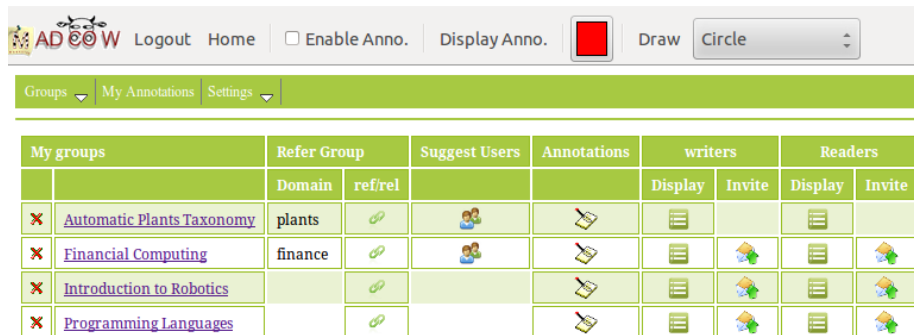


Fig. 11: MADCOW Toolbar.

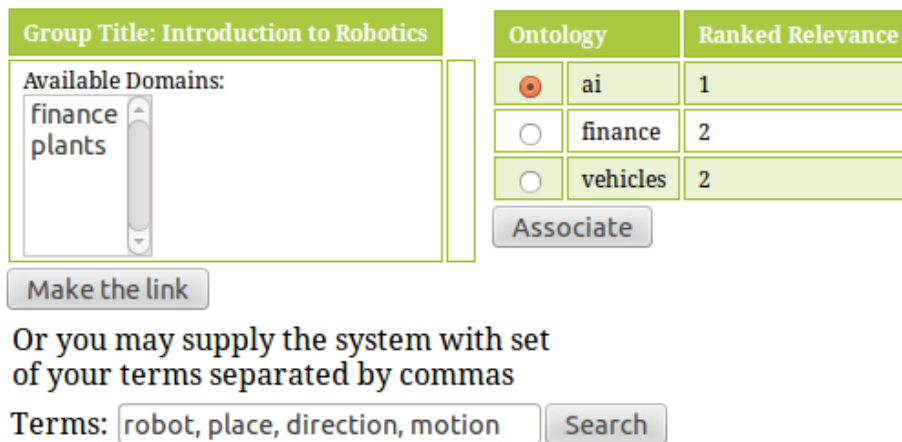


Fig. 12: Taya lists available domains.

6 Experimental Tests

We conducted a pilot test on the matching process. 17 participants were divided into 3 disjoint sets (of numerosity 6, 6, 5). Users from the first set created 16 MADCOW groups and assigned domains to them, then manually invited participants from the second set to join the groups (creating 20 invitations). The invited participants accepted the invitations and submitted annotations to the joined groups (26 annotations). Participants in the third set submitted private and public annotations (31 annotations), associating tags of their choice with them. We asked users to annotate a set of similar websites to check URL matching (using 10 different websites). Group owners requested the system to suggest members and invitations were sent (9 invitations). Participants of the third set required the system to suggest proper groups and sent membership requests (12 requests). A first observation indicated that the average invitation

Members Suggestions

Group Title: Introduction to Robotics		URL Match	
Username	Ranked Relevance	Username	Ranked Relevance
<input checked="" type="checkbox"/>	dando 1	<input checked="" type="checkbox"/>	calvitto 1
<input checked="" type="checkbox"/>	cristina 2		tea 2
<input type="checkbox"/>	tea 3	<input type="checkbox"/>	eduardo 3
<input type="checkbox"/>	daniel 3		dando 3

Fig. 13: Taya sends invitations to most relevant users.

Groups Suggestions:

Group	Domain	Ranked Relevance
<input type="checkbox"/> Financial Computing	finance	1
<input type="checkbox"/> Automatic Plants Taxonomy	plants	2
<input type="checkbox"/> Introduction to Robotics	ai	3

Suggestions of groups containing annotations on URLs of interest to the user:

Group	Domain	Ranked Relevane
<input type="checkbox"/> Introduction to Robotics	ai	1

Fig. 14: List of the most relative groups for Daniel.

duration for group owners of the whole process – obtaining user suggestions / selecting users / sending invitations – was decreased to 10.6 seconds from the 99.25 seconds required in average for the manual process.

Despite the observation of the decrease in time and effort consumed in joining groups and the increase in groups-users relevancy, we identify the following threats to the validity of the the pilot study:

1. Lack of participants from different disciplines (all participants were related to computer science).
2. Using only 6 ontologies limits group diversity, even though they reflected quite different domains.
3. Group owners could have a bias towards specific kinds of group related to their interests.
4. The annotation tags might not be relevant to the actual contents of the annotation.

It is to be observed, however, that the latter two phenomena might be common also in real-life usage of the system.

After the test, we asked participants to fill a questionnaire to test the usability and understandability of groups-users matching. Questions are divided into two groups: for groups' owners and for users. We report on some of these results in Tables 3-7, and also illustrate them in Figs. 15-19.

As a group owner:

1. How do you find the Domain-Group referring process?
2. How do you find the Users-to-Group suggestion process?
3. Are you satisfied with the results of “members suggestions”?
4. What do you recommend for joining your group?
5. Which ranked domain did you choose (in group-domain reference)?
6. Which ranked user did you choose (in members suggestion)?

As a user:

7. How do you find the Group-to-User suggestion process?
8. Are you satisfied with the results of “groups suggestions”?
9. What do you prefer?
10. Which ranked group did you choose (in group suggestion)?

Table 3: Percentage answers values for questions 1, 2 and 7.

Question#	1	2	7
Easy	94.12%	88.24%	94.12%
Moderately Hard	5.88%	5.88%	5.88%
Hard	0%	0%	0%
Vague	0%	5.88%	0%

Table 3 and Fig. 15 show the percentage of answers for testing the understandability of the process of referring groups to domains and of the matching process for both owners and users. We observe that most participants find it easy.

Table 4: Percentage answers for questions 4.

Question#	4
<i>Manual</i>	11.76%
<i>Members' Suggestions</i>	70.65%
<i>Join Requests</i>	0%
<i>System Suggestions</i>	17.65%

Table 4 and Fig. 16 illustrate the percentage of answers of owners for the preferred method of joining to their groups. A high percentage is noticed for the *Members' Suggestions* option, reflecting the owners' preference to use the system ability to suggest members for their groups as a result of the matching between their groups and system users.

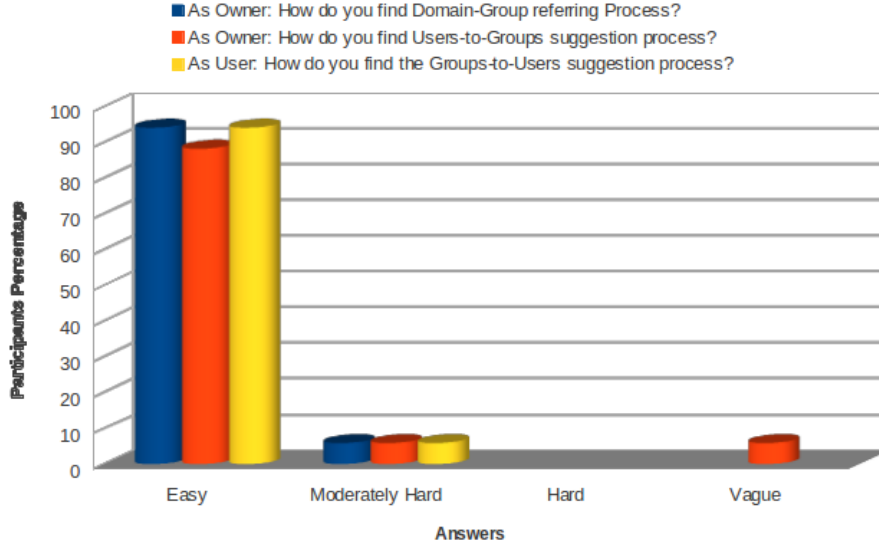


Fig. 15: Table 3 data representation.

Table 5: Percentage answers values for questions 5, 6 and 10.

Question#	5	6	10
<i>First</i>	82.35%	82.35%	100%
<i>Second</i>	17.65%	17.65%	0%
<i>No Choice</i>	0%	0%	0%

In this test, we wanted to know the preferences of participants with respect to the methods for selecting users and/or groups from the generated ranked lists. Table 5 and Fig. 17 illustrate that most of participants selected the groups / users suggested as first by the matching process, thus indicating the validity of the matching algorithm.

Table 6: Percentage answers values for questions 3 and 8.

Question#	3	8
<i>Yes</i>	94.12%	100%
<i>No</i>	5.88%	0%

In measuring the amount of satisfaction with the results of generated ranked lists, we obtained good indications from participants. Table 6 and Fig. 18 present these results.

Finally, Table 7 and Fig. 19 highlight that most of the users prefer to use the ability of the system to suggest proper groups for them rather than using

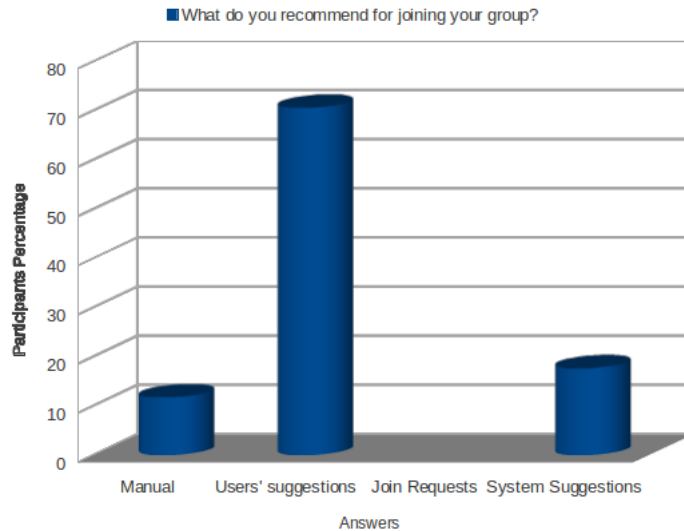


Fig. 16: Table 4 data representation.

Table 7: Percentage answers values for questions 9.

Question#	9
<i>Searching for a group, then send join request</i>	23.53%
<i>Use system groups suggestion feature, then send join request</i>	76.47%

the ordinary and manual method of groups joining.

7 Conclusions and future work

Ontologies have been introduced to the MADCOW annotation system to complement the notion of group, and tools to find matches between groups and potentially interested users have been realised. Ontologies are used to represent domain knowledge relevant to the formation of MADCOW groups. The ontology associated with a domain provides terms which can be used as tags for annotations posted to groups referring to that domain, facilitating a better structuring of the knowledge shared within the group. Tools for facilitating the retrieval of interested users (or of interesting groups) are provided based on matches between tags freely used by submitters of annotations and terms contained in the different ontologies integrated in the MADCOW system.

Referring MADCOW groups with universal knowledge representatives and matching them with users helped mitigate the time, effort and irrelevance problems. This improvement introduced a novel way to enhance the group joining process in an annotation system and improves the process of attracting inter-

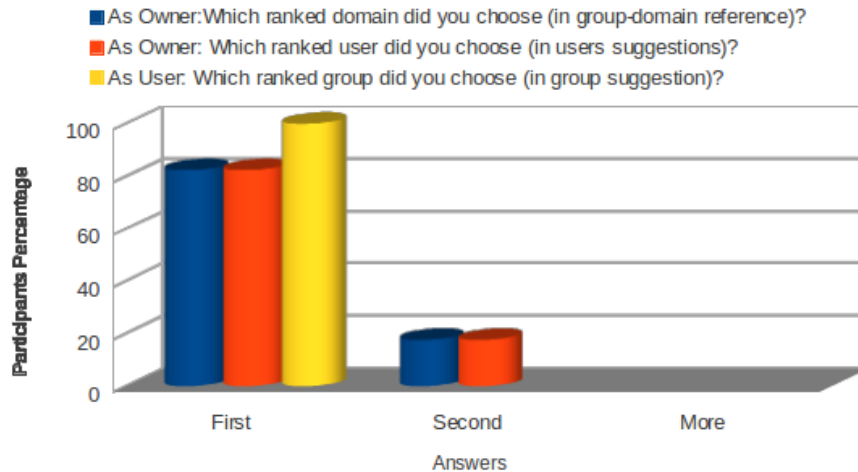


Fig. 17: Table 5 data representation.

esting users to relevant groups.

As future work, we plan to proceed with deeper experimental investigation and explore possible enhancements to the use of CMM for relevance analysis. An interesting avenue seems to be its integration with ontology quality measures, e.g. the density measure that takes into account the amount of ontology coverage for a given term. We will also explore the possibility of allowing a group to be associated with several ontologies, and verify its impact on groups-users matching.

References

- [1] P. Bottoni, R. Civica, S. Levialdi, L. Orso, E. Panizzi, and R. Trinchese, “Madcow: a multimedia digital annotation system,” in *Proc. AVI’04*. ACM, 2004, pp. 55–62.
- [2] D. Avola, P. Bottoni, and R. Genzone, “Light-weight composition of personal documents from distributed information,” in *Proc. IS-EUD 2011*, ser. LNCS, M. Costabile, Y. Dittrich, G. Fischer, and A. Piccinno, Eds. Springer Berlin / Heidelberg, 2011, vol. 6654, pp. 221–226. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21530-8_17
- [3] R. Heck, S. Luebke, and C. Obermark, “A Survey of Web Annotation Systems,” 2008. [Online]. Available: http://www.math.grin.edu/~rebelsky/Blazers/Annotations/Summer1999/Papers/survey_paper.html
- [4] D. Barger, J. Grudin, A. Gupta, E. Sanocki, F. Li, and S. Leetiernan,

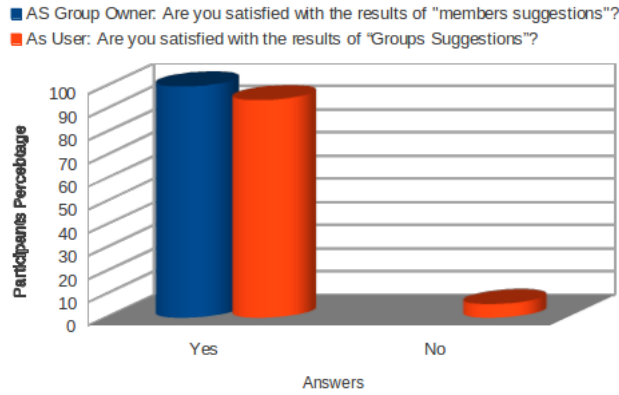


Fig. 18: Table 6 data representation.

“Asynchronous collaboration around multimedia applied to on-demand education,” *J. Manage. Inf. Syst.*, vol. 18, no. 4, pp. 117–145, Mar. 2002. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1289722.1289729>

- [5] A. Sakar and G. Ercetin, “Effectiveness of hypermedia annotations for foreign language reading,” *Journal of Computer Assisted Learning*, vol. 21, no. 1, pp. 28–38, 2005. [Online]. Available: <http://dx.doi.org/10.1111/j.1365-2729.2005.00108.x>
- [6] Y.-S. Lai, H.-H. Tsai, and P.-T. Yu, “Integrating annotations into a dual-slide powerpoint presentation for classroom learning.” *Educational Technology & Society*, vol. 14, no. 2, pp. 43–57, 2011. [Online]. Available: <http://dblp.uni-trier.de/db/journals/ets/ets14.html#LaiTY11a>
- [7] D. Avola, P. Bottoni, M. Laureti, S. Levialdi, and E. Panizzi, “Managing groups and group annotations in madcow,” in *Proc. DNIS 2010*, ser. LNCS, vol. 5999, 2010, pp. 194–209.
- [8] C. Brewster, K. O’Hara, S. Fuller, Y. Wilks, E. Franconi, M. A. Musen, J. Ellman, and S. B. Shum, “Knowledge representation with ontologies: The present and future,” *IEEE Intelligent Systems*, pp. 72–81, January 2004. [Online]. Available: <http://eprints.soton.ac.uk/262481/>
- [9] B. Chandrasekaran, J. R. Josephson, and V. Benjamins, “What are ontologies, and why do we need them?” *IEEE Intelligent Systems*, vol. 14, no. 1, pp. 20–26, Jan. 1999. [Online]. Available: <http://dx.doi.org/10.1109/5254.747902>
- [10] R. Gil, A. Borges, and L. Contreras, “Shared ontologies to increase systems interoperability in university institutions,” in *Proc. IMCSIT 2007*, 2007,

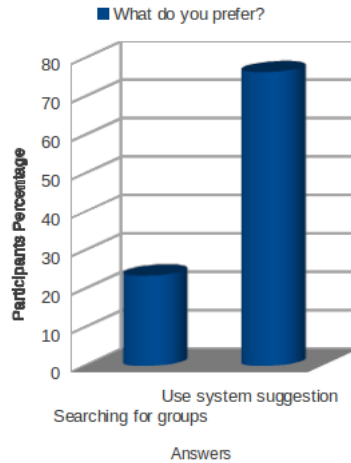


Fig. 19: Table 7 data representation.

pp. 799–808. [Online]. Available: <http://www.proceedings2007.imcsit.org/pliks/126.pdf>

- [11] D. Vallet, M. Fernandez, and P. Castells, “An ontology-based information retrieval model,” in *Proc. ESWC 2005*. Springer, 2005, pp. 455–470.
- [12] Y. Zhang, W. Vasconcelos, and D. Sleeman, “Ontosearch: An ontology search engine,” in *Research and Development in Intelligent Systems XXI*, M. Bramer, F. Coenen, and T. Allen, Eds. Springer London, 2005, pp. 58–69. [Online]. Available: http://dx.doi.org/10.1007/1-84628-102-4_5
- [13] RDF Working Group, “RDF/XML Syntax Specification (Revised),” <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>, OMG, Tech. Rep., 2004.
- [14] OWL Working Group, “OWL Web Ontology Language,” <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>, OMG, Tech. Rep., 2004.
- [15] H. Alani, C. Brewster, and N. Shadbolt, “Ranking ontologies with aktiverank,” in *Proc. ISWC’06*. Springer, 2006, pp. 5–9.
- [16] J. Paralic and I. Kostial, “Ontology-based information retrieval,” in *Proc. IIS 2003*, 2003, pp. 23–28.
- [17] R. Braga, C. Werner, and M. Mattoso, “Using ontologies for domain information retrieval,” in *Proc. DEXA 2000*, 2000, pp. 836–840.

- [18] C. Patel, J. Cimino, J. Dolby, A. Fokoue, A. Kalyanpur, A. Kershenbaum, L. Ma, E. Schonberg, and K. Srinivas, “Matching patient records to clinical trials using ontologies,” in *Proc. ISWC’07/ASWC’07*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 816–829.
- [19] S. Park, W. Kim, S. Lee, and S. Bang, “Product matching through ontology mapping in comparison shopping,” in *Proc. iiWAS 2006*, ser. books@ocg.at, vol. 214. Austrian Computer Society, 2006, pp. 39–49.
- [20] H. Tangmunarunkit, S. Decker, and C. Kesselman, “Ontology-based resource matching in the grid - the grid meets the semantic web,” in *Proc. ISWC 2003*, ser. LNCS, vol. 2870, 2003, pp. 706–721.
- [21] M. Fazel-Zarandi and M. Fox, “Reasoning about skills and competencies,” in *CProc. PRO-VE 2010*, ser. IFIP AICT. Springer Berlin Heidelberg, 2010, vol. 336, pp. 372–379.
- [22] I. Cantador, P. Castells, D. Vallet, and E. Politcnica, “Enriching group profiles with ontologies for knowledge-driven collaborative content retrieval,” in *Proc. STICA 2006 at WETICE 2006*, 2006, pp. 358–363.
- [23] V. Cordi, P. Lombardi, M. Martelli, and V. Mascardi, “An ontology-based similarity between sets of concepts,” in *Proc. WOA 2005*. Pitagora Editrice, 2005, pp. 16–21.
- [24] Y. Li, D. Mclean, Z. Bandar, J. O’Shea, and K. Crockett, “Sentence similarity based on semantic nets and corpus statistics,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 8, pp. 1138–1150, 2006.
- [25] S. K. Rhee, J. Lee, and M. woong Park, “M.w.: Ontology-based semantic relevance measure,” in *Proc. 1st Int. Wks. on Semantic Web and Web 2.0 in Architectural, Product and Engineering Design*, 2007.
- [26] I. Trestian, S. Ranjan, A. Kuzmanovic, and A. Nucci, “Measuring serendipity: connecting people, locations and interests in a mobile 3g network,” in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, ser. IMC ’09. New York, NY, USA: ACM, 2009, pp. 267–279. [Online]. Available: <http://doi.acm.org/10.1145/1644893.1644926>
- [27] D. Avola, P. Bottoni, and A. Hawash, “Using ontologies for users-groups matching in an annotation system,” in *Computer Science and Information Technology (CSIT), 2013 5th International Conference on*, 2013, pp. 38–44.
- [28] A. Hawash, “Introducing groups to an annotation system,” in *CHIItaly2013, Trento, 16-19 September*, 2013.
- [29] D. Avola, P. Bottoni, and A. Hawash, “Groups-users matching in an annotation system using ontologies (class match measure),” in *CHIItaly2013, Trento, 16-19 September*, 2013.

- [30] P. Velardi, S. Faralli, and R. Navigli, “OntoLearn reloaded: A graph-based algorithm for taxonomy induction,” *Computational Linguistics*, vol. 39(3), 2013.
- [31] P. Atzeni, S. Paolozzi, and P. D. Nostro, “Ontologies and databases: Going back and forth,” in *Proc. ODBIS 2008*, 2008, pp. 9–16.