Groups-Users Matching in an Annotation System Using Ontologies (Class Match Measure)

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

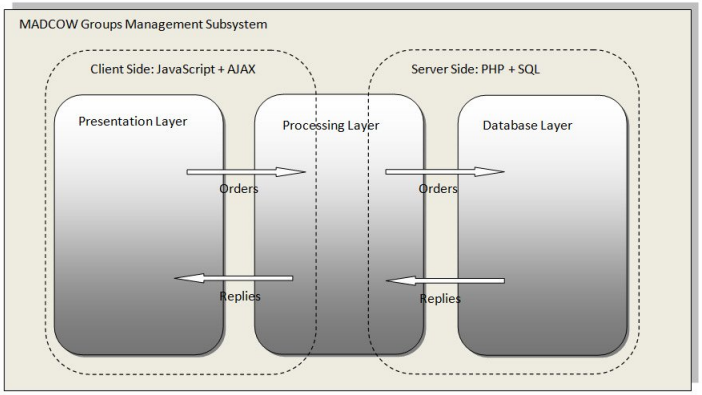

Figure 1: MADCOW 3-tier Client-Server Architecture.

CHItaly 2013, 16-19 September, 2013, Trento, Italy.

Abstract
Introducing groups to the MADCOW annotation system solves the privacy-collaboration problem for users, but a new challenge emerges for the group joining process in terms of irrelevance, time and effort, for both users and groups' authorisers. We introduce MADCOW *domains* and link them to domain ontologies. Users can then use tags derived from ontology terms to synthesize the intent and scope of their annotations. We propose the use of matching measures between domains and users' annotations as the basis for suggestions of groups to users and of users to groups authorisers. A reduction of the mentioned problems ensues.

Author Keywords
Annotation; Groups; Joining Process; Ontologies.

ACM Classification Keywords
H.5.3 [Information interfaces and presentation]: Group and Organization Interfaces. Web-based interaction.

MADCOW System
MADCOW[1] supports the annotation of (portions of) texts, images and videos with textual content, links to other resources, and user-defined tags [4], through a 3-tier architecture (see Figure 1). Annotations published in

[1]Multimedia Annotation of Digital Content Over the Web

MADCOW are of three types: (1) *public*: viewable by any user, (2) *private*: viewable by their submitters only, or (3) *group-related*: viewable by any member of the group to which they are posted, and nobody else.

Users Collaboration-Privacy Conflict

Public annotations increase collaboration among users while private ones increase their privacy. Introducing groups to MADCOW solved this conflict through the notion of group-related annotation [3]. A group owner can select one of three different policies to allow users to join it: (1) `Public`: any user can join the group, (2) `Invite`: users can join a group only if invited by some group member with *authoriser* status[2], or (3) `Apply`: users apply for joining the group, and admission is subject to approval by one of the authorisers.

Scenario

A teacher of a course on Web design metrics wants his students to perform collaborative evaluation of Web sites (site structure and graphical quality) by annotating their contents. He starts by letting students submit private and public annotations, but later on he notices limited and unfocused collaboration patterns. He decides to partition them into two sets according to their interests. Hence, he creates two groups, for the domains "Site Structure" and "Graphics Quality", respectively. He asks the system to suggest students for each group, based on the interests shown by the use of specific tags. A ranked list of students is generated for each group, based on which the teacher enlists students in the two groups and asks them to submit annotations only within groups.

Manual Group-Join Problems

The manual joining process introduces two main problems:

[2]This is given to the group owner and to group moderators.

Table 1: Number of operations and average duration.

	# of times	*Mean*
Create	72	37.3
Update	51	15.9
Invite	719	99.25
Join	125	5.6

Irrelevance Problem

Authorisers deciding to invite users to their group and users looking for relevant groups share similar, if symmetrical, problems: "**how do authorisers know who could be interested in joining their group?**" or "**what groups exist which might interest a user?**". This causes users to be faced with a wealth of irrelevant data when choosing target groups or users.

Time-effort problem

Without matching, authorisers have to list all MADCOW users (possibly looking at their public annotations) and select some of them as receivers of invitations. Depending only on the title and a textual description of the group topic, users can send join requests to the groups considered relevant to their interests. Both processes consume time and effort and become unwieldy as the number of groups or users increases. Table 1 presents information gathered from an extensive test involving 162 participants to measure the usability of mechanisms for group management, showing 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 is a lengthy process.

Groups-Users Automatic Suggestion

Manual-group joining problems can be solved by proper groups-users matching so that authorisers are presented with the most relevant users and users with the most relevant groups. Associating groups with publicly available representations of knowledge relevant to the group objectives is done by the selection of well-defined terms in which to express domain knowledge [5]. To achieve this goal, a group creator has to associate it with some

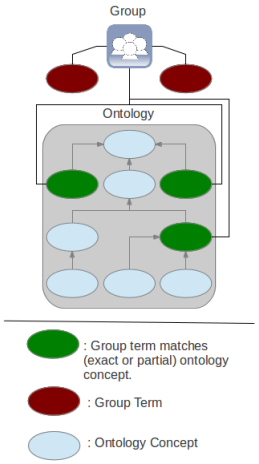

Figure 2: Group-Ontology Matching.

existing ontology, either manually, by selecting the most appropriate ontology, or by providing set of terms which reflect the intent of the group. On the annotator side, users can complement their annotations with tags to represent the intent of these annotations. The **MADCOW Ontology Browser** executes a matching between the terms of the representative domain and the tags of public annotations in the users-to-groups suggestion process. The same matching is done for a user searching for proper groups, where the match will include all of the user's public and private annotations.

Relating ontologies and groups

We introduce some definitions to illustrate the role of ontologies in the process of groups-users matching: **Domain**: a unique name designating the area of knowledge to which an ontology refers. **Concept**: the name associated with a node in some ontology. **Terms**: lexicalisations of the concepts in some ontology. **Tags**: words (possibly terms in some ontology) provided by a user to characterise the annotation content [2].

Group-Ontology Assocoiation

As described in the following pseudocode, if a group owner provides a name for a domain the system checks whether an ontology associated with this name exists, in which case it would create an association. Otherwise, the owner is asked to input a set of terms characterising the group and which might define its domain. The system then presents the ontologies for which a match with a significant number of terms exists (Figure 2).

```
selectedGroup = selectGroup();
domains = loadDomains();
repeat {
 selectedDomain = selectDomain(domains);
 if(checkDomain(selectedDomain.getTerms())) {
  associate(selectedGroup,selectedDomain);
  break; }
} until(allDomainsConsidered());
if(!group.checkAssociation()) {
 matchedOntologies = new List();
 terms = askForTerms();
 foreach (ontology in MADCOW.Ontologies)
  if(CMM(terms,ontology.getTerms())>0)
   matchedOntologies.add(ontology);
 matchedOntologies.rank();
 selectedOntology = selectOntology();
 if(!selectedOntology.isAssociated())
  selectedOntology.associateWithDomain(selectedOntology.Title);
 associate(selectedGroup,selectedOntology.getDomain());
```

Annotation Submission

For annotations submitted as public or private, users can use any set of tags as metadata associated with the annotation. Tags attached to these annotations will be used in the matching process to propose groups to users, and users to authorisers (only for public annotations).

Matching Process

The algorithms to match annotations with domains, to which sets of groups correspond.

In case a group owner asks for potential members, a matching is executed between terms of the referring domain and tags of public annotations of users not already members in the group. The appropriate users are proposed, ranked according to their relevance to the domain. If a user asks for potentially interesting groups, the matching process considers all of the user's annotations (public and private) and ranks the groups related to the most significant domains.

The Class Match Measure (exact and partial) is used to assess relevance of a group terms to domains in the process of group-domain association, and of annotations to domains in the users-to-group and domain-to-user suggestions [1]. Figure 3 schematises the matching process in the domain-user association process.

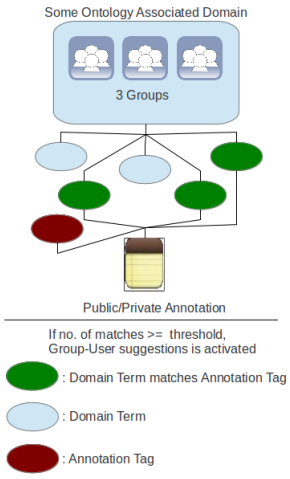

Figure 3: Domain-User Matching.

Given a domain D and an annotation A, let $L_D = \{r_1, r_2, \ldots, r_n\}$ be the set of domain terms and let $T = \{t_1, t_2, \ldots, t_m\}$ be the set of tags. The matching between D and A is the process of checking the existence, within A, of a term from L_D, considering exact and partial matches. Currently, a $\#matches > 0$ condition is used to filter out spurious domains.

Results

An extensive test has been conducted to compare the joint duration for group owners of the processes of obtaining user suggestions / selecting users / sending invitations with that of the manual invitation process (Table 1). The average invitation duration is decreased from 99.25 to 10.6 seconds. An online questionnaire on system usability was filled by participants, where (*E* is for Easy, *M* for Moderately Hard, *H* for Hard and *V* for Vague):

As a group owner:

1. How do you find Domain-Group referring process? *(*E:*94.12%,* M:*5.88%,* H:*0%,* V:*0%)*
2. How do you find Users-to-Group suggestion process? *(*E:*88.24%,* M:*5.88%,* H:*0%,* V:*5.88%)*
3. What do you recommend for joining your group? *(Manual:11.76%, Users' Suggestions:70.59%, Join Requests:0%, System Suggestions:17.65%)*

As a user:

1. How do you find the Group-to-User suggestion process? *(*E:*94.12%,* M:*5.88%,* H:*0%,* V:*0%)*
2. Are you satisfied with the results of "groups suggestions"?*(yes:94.12%, no:5.88%)*
3. What do you prefer? *(Searching for a group, then send join request:23.53%, Use system groups suggestion feature, then send join request:76.47%)*

Conclusions and future work

The introduction of groups to MADCOW system represents a solution to the users' privacy-collaboration conflict. Problems related to manual groups-join are solved by users-groups matching based on involving ontologies as representatives for groups knowledge. Results from experiments and participants' feedbacks are promising. As future work, we plan to allow group-domain multiple associations, tuning the ranking values by Fuzzy Logic, and to study other relevance measurements.

References

[1] Alani, H., Brewster, C., and Shadbolt, N. Ranking ontologies with aktiverank. In *Proc. ISWC'06*, Springer (2006), 5–9.

[2] Avola, D., Bottoni, P., and Hawash, A. Using ontologies for users-groups matching in an annotation system. In *Proc. CSIT 2013* (2013), 38–44.

[3] Avola, D., Bottoni, P., Laureti, M., Levialdi, S., and Panizzi, E. Managing groups and group annotations in madcow. In *Proc. DNIS 2010*, vol. 5999 of *LNCS* (2010), 194–209.

[4] Bottoni, P., Civica, R., Levialdi, S., Orso, L., Panizzi, E., and Trinchese, R. Madcow: a multimedia digital annotation system. In *Proc. AVI'04*, ACM (2004), 55–62.

[5] Brewster, C., O'Hara, K., Fuller, S., Wilks, Y., Franconi, E., Musen, M. A., Ellman, J., and Shum, S. B. Knowledge representation with ontologies: The present and future. *IEEE Intelligent Systems* (January 2004), 72–81.