GroupMe! - Where Semantic Web meets Web 2.0

Fabian Abel, Mischa Frank, Nicola Henze, Daniel Krause,
Daniel Plappert, and Patrick Siehndel

IVS – Semantic Web Group, University of Hannover, Hannover, Germany
{abel,frank,henze,krause,plappert,siehndel}@kbs.uni-hannover.de

Abstract. Grouping is an attractive interaction metaphor for users to create reference collections of Web resources they are interested in. Each grouping activity has a certain semantics: things which were previously unrelated are now connected with others via the group. We present the GroupMe! application which allows users to group and arrange multimedia Web resources they are interested in. GroupMe! has an easy-to-use interface for gathering and grouping of resources, and allows users to tag everything they like. The semantics of any user interaction is captured, transformed and stored as adequate RDF descriptions. As an example application of this automatically derived RDF content, we show the enhancement of search for tagged Web resources, which evaluates the grouping information to deduce additional contextual information about the resources. GroupMe! is available via http://www.groupme.org.

1 Introduction

The success of the so-called Web 2.0 has shown that people enjoy features like tagging or collaborative spaces. Community platforms (e.g. wikis, blogs, social bookmarking systems) provide users with high *intercreativity* and form some kind of *collective intelligence*. Although some platforms offer public interfaces to access the community knowledge, sharing knowledge across community boundaries is still limited. One of the reasons for this can be attributed to the fact that Semantic Web technologies are rarely used here. GroupMe! combines approaches from these two areas.

GroupMe! extends the idea of social bookmarking systems like *del.icio.us*[1] and systems that allow (re-)organizing web content like *combinFormation* [1] in many aspects: (1) users are able to build groups of arbitrary (multimedia) Web resources by simple drag & drop operations, (2) resources contained in such groups can be (re-)arranged by the users (the visualization of the resources is adapted to content type), and (3) the grouping and tagging activities produce RDF descriptions in an easy and collaborative way (e.g. dropping a Web resource into a group effects the transformation of the resource's community-specific attributes into attributes adhering to common ontologies). RDF content

[1] http://del.icio.us

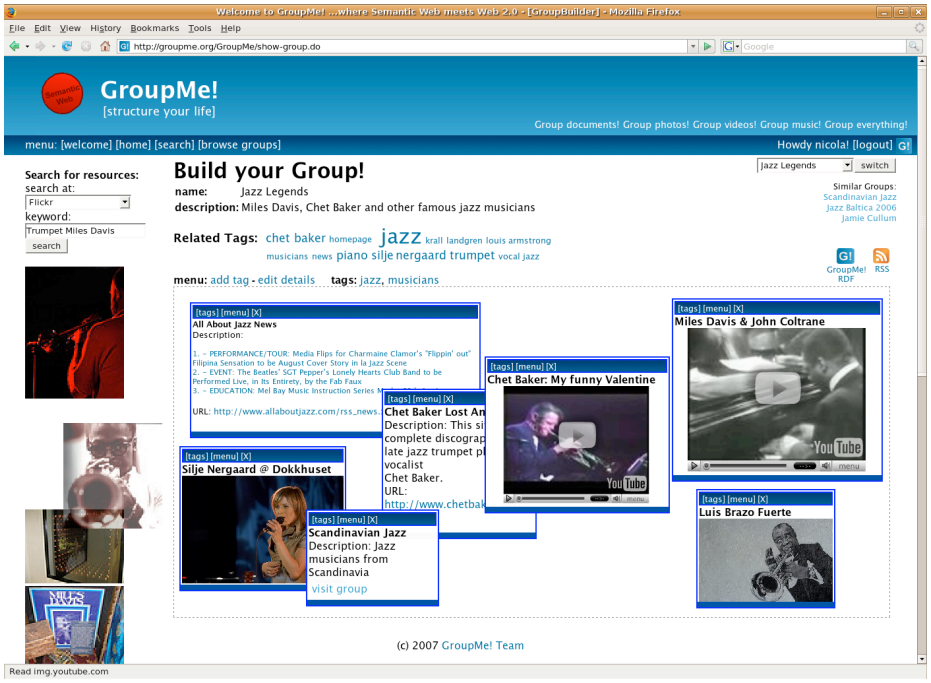

Fig. 1. Screenshot: group builder of the GroupMe! application

that is created in GroupMe! is given back to the Web via an *RESTful* [2] API using well known vocabularies like *FOAF*[2], *RSS*[3], or *DCMI element set*[4], and a GroupMe! vocabulary, which captures new concepts like groups, relation of groups and resources, etc.

The GroupMe! approach provides the possibility to overcome the gap between Web 2.0 and the Semantic Web, and offers with the grouping facility new and interesting strategies for search and content discovery. In the following section, we introduce the GroupMe! application with an example scenario. Afterwards, we discuss a context-aware search strategy that takes grouping information into account. Section 4 briefly discusses technical issues of the GroupMe! architecture, and Section 5 summarizes the paper.

2 The GroupMe! Approach - Scenario

Figure 1 shows the group builder functionality of GroupMe!. In the illustrated example the user is building a group which he tags with *Jazz Legends*. The search form on the left hand side in Fig. 1 enables the user to utilize several search

[2] http://xmlns.com/foaf/spec/
[3] http://web.resource.org/rss/1.0/
[4] http://dublincore.org/documents/dces/

engines for gathering resources of interest (in arbitrary format). With a simple *drag & drop* operation he adds suitable resources to the group (right hand side in Fig. 1), and arranges them within the group as he likes. In the depicted scenario the user has already added some resources to the group: two YouTube[5] videos of live performances, two Flickr[6] photos of musicians, a news feed, a website about *Chet Baker*, and a GroupMe! group, which deals with *Scandinavian Jazz*. All these resources are visualized according to their content type, e.g. the news feed displays the three latest news items, videos are embedded, etc. Every time the user adds a resource to a group, GroupMe! produces RDF: Metadata is extracted from the aggregated resources (e.g. a Flickr image title is transformed to the DCMI element *title*, etc.), and the grouping activities are captured as RDF (cf. section 4.2). Produced RDF can immediately be accessed via our API or static links (see buttons *"GroupMe! RDF"* and *"RSS"*).

Besides grouping and arranging resources within groups, the user can tag resources and groups. These tags are used to provide enhanced navigation possibilities: by clicking on tags of the group-specific *tag cloud* (see *related tags* in Fig. 1), GroupMe! lists matching resources and groups, and by clicking on a similar group (see top right in Fig. 1) the user is directed to the selected group.

3 Folksonomies and Search Strategies

3.1 GroupMe! folksonomies

The core data of GroupMe! evolves over time by tagging resources and groups, and by grouping resources and (re-)arranging them. Tagging is done by users (*folks*), and results in a collection of concepts (*"taxonomy"*) that is called *folksonomy*. In [3] a folksonomy is formally defined as follows:

Definition 1 (Folksonomy).
A folksonomy is a tuple $\mathbb{F} := (U, T, R, Y, \prec)$*, where:*

- U*,* T *and* R *are finite sets that contain instances of users, tags and resources*
- Y *defines a relation (*tag assignment*) between these sets:* $Y \subseteq U \times T \times R$
- $\prec$ *defines a user-specific subtag/supertag-relation between tags:* $\prec \subseteq U \times T \times T$

The GroupMe! application uses an adapted version of definition 1: Users are able to tag resources (*tag assignment*), and $\prec$-relations can in principle be deduced using ontology learning techniques [4]. In GroupMe!, we understand the notion of *resources* in a general sense, e.g. a resource can either be a resource or a more complex object: a set of resources.

Definition 2 (Group). *A group is a set of resources.*

As a group is a resource as well, groups can contain groups (which was e.g. the case in the scenario described in Section 2). With this definition of groups, we extend definition 1 to be able to tag resources and sets of resources (groups):

[5] http://www.youtube.com
[6] http://www.flickr.com

Definition 3 (Extended Folksonomy).
An extended folksonomy is a tuple $\mathbb{F} := (U, T, \breve{R}, Y, \prec)$*, where:*

- $\breve{R} = R \cup G$ *(the union of the set of resources* R *and the set of groups* $G \subseteq 2^R$*)*
- *consequently the definition of* Y *is replaced by:* $Y \subseteq U \times T \times \breve{R}$*,* $\prec$ *analogously.*

According to Fig. 1, the tag assignment for group *Jazz Legends* can be modeled with the following statements:

```
(gm:nicola, gm:jazz, gm:jazzLegendsGroup)
(gm:nicola, gm:musicians, gm:jazzLegendsGroup)
```

Furthermore, resources can be tagged in context of a certain group. This enables us to gain additional knowledge about the tag assignment. The definition of a GroupMe! folksonomy, which considers such a group context, is formalized as:

Definition 4 (GroupMe! Folksonomy).
A GroupMe! folksonomy is a tuple $\mathbb{F} := (U, T, \breve{R}, G, \breve{Y}, \prec)$*, where:*

- $\breve{Y}$ *defines the extended tag assignment:* $\breve{Y} \subseteq U \times T \times \breve{R} \times G$*,* $\prec$ *analogously*

By introducing the group context in addition to the $\prec$-relation, we obtain other kinds of relations between tags which can be used for many purposes (to deduce tags for untagged resources, to derive a "neighborhood" of a tag, to mine frequently occurring neighborhoods, etc.). To continue our example, allocating tags to the resources grouped in Fig. 1 is expressed via:

```
(gm:nicola, gm:trumpet, gm:chetBakerVideo, gm:jazzLegendsGroup)
(gm:nicola, gm:jazz, gm:chetBakerVideo, gm:jazzLegendsGroup)
(gm:nicola, gm:vocalJazz, gm:siljeNergaardPhoto, gm:jazzLegendsGroup)
...
```

3.2 Search Strategies

When searching for resources by providing a set of query terms, the GroupMe! application considers not only the tags that are directly assigned to a resource by a user, but also contextual information of resources. Different strategies, which have potential to enhance search strategies as proposed in [5], are possible. Fig. 2 shows an example of such a strategy.

(1) In the first step, all resources that are directly tagged with one of the query terms are collected. The weight which is associated with these fitting resources depends on the number of users that tagged resource r with tags contained in T_{query}, e.g.:

$$directWeight(T_{query}, r) = \sum_{t \in T_{query}} resourceWeight(t, r), \text{where}$$

$$resourceWeight(t, r) = \frac{number\ of\ users\ who\ tagged\ resource\ r\ with\ t}{number\ of\ users\ who\ tagged\ resource\ r}$$

rankResources(T_{query})**:**
T_{query}: *tags which represent the search query;*
$\breve{R}_{result}$: *set of fitting resources;*
G_{temp}: *set of fitting groups; //it is:* $G_{temp} \subseteq \breve{R}_{result}$

(1) for each resource $r \in \breve{R}$ **:**
if($\exists t \in T_{query}$: *r is tagged with t*)**:**
$r.directWeight = directWeight(T_{query}, r)$;
$\breve{R}_{result} = \breve{R}_{result} \cup \{r\}$;
if($r \in G$)**:**
$G_{temp} = G_{temp} \cup \{r\}$;

(2) for each group $g \in G_{temp}$ **:**
for each resource $r \in g$ **:**
$n = |\{g' \in G_{temp} | r \in g'\}|$;
$r.contextWeight \mathrel{+}= contextWeight(T_{query}, r, g) \cdot \frac{1}{n}$;
$\breve{R}_{result} = \breve{R}_{result} \cup \{r\}$;

(3) for each $r \in \breve{R}_{result}$**:**
$r.weight = \alpha \cdot r.directWeight + \beta \cdot r.contextWeight$;

(4) $ranking$ = order $\breve{R}_{result}$ by $r.weight$;
return $ranking$;

Fig. 2. Search strategy which considers the group context

(2) Afterwards all *fitting groups* (i.e. those groups that are tagged with at least one of the query terms) are determined, and the resources in these groups are weighted according to their appearances in groups. There are a several ways to compute such *context weight*, e.g.:

$$contextWeight(T_{query}, r, g) = \sum_{t \in T_{query}} resourceWeight(t, r) \cdot groupWeight(t, g),$$

$$\text{where } groupWeight(t, g) = \frac{number\ of\ resources\ in\ g\ that\ are\ tagged\ with\ t}{number\ of\ resources\ in\ g}$$

As a resource may appear in several groups $g \in G_{temp}$, we compute the average of the corresponding $contextWeights$.

(3) In the next step, the values of both weights are combined into an *overall weight*. α or β can be used to emphasize either direct or context weight.

(4) Finally, all found resources are ranked according to their overall weight and returned as an ordered list.

4 The GroupMe! system

At a glance the GroupMe! system provides three core functionalities for users:

1. Creation of groups. Users can add arbitrary Web resource to a group (including other groups), arrange them within the group, and tag both resources and groups.

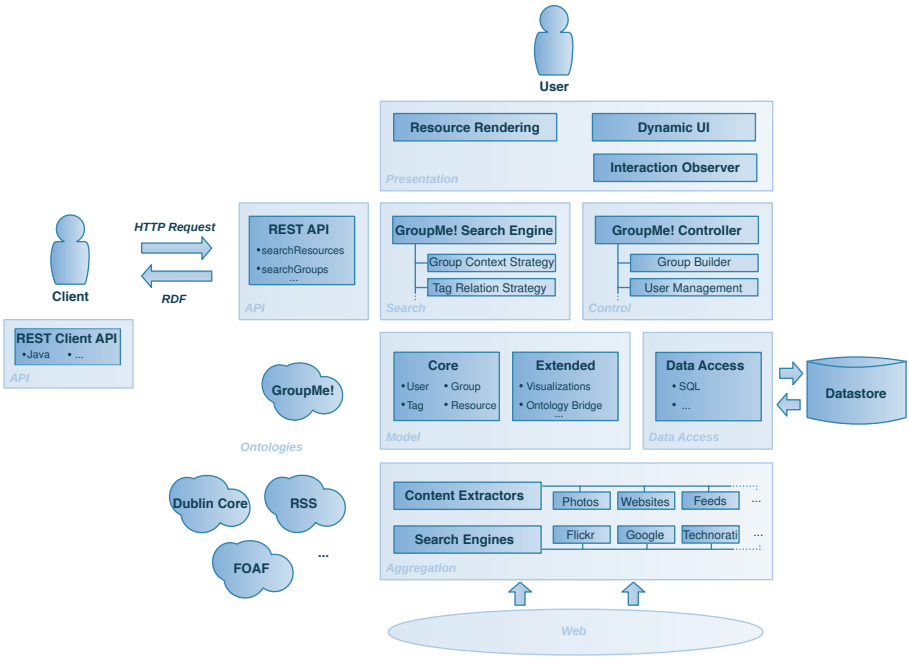

Fig. 3. Technical overview of the GroupMe! application

2. **Search and navigation.** Ranking strategies that utilize information about tags and groups are applied for search and navigation facilities.
3. **RDF export.** RDF content which is produced within the process of creating groups and arranging resources within groups is made available to the public.

4.1 Architecture

The architecture of the GroupMe! application is outlined in Fig. 3. In technical terms, GroupMe! is a Web application, which adheres to the Model-View-Controller pattern. It consists of four basic layers:

Aggregation. The aggregation layer provides functionality to search for resources that should be included into GroupMe! groups (*Search Engines*). *Content Extractors* allow us to process gathered resources in order to extract useful metadata and convert them into RDF resources with semantically well defined descriptions. When e.g. adding a result from the Flickr search engine into a group, a *Photo content extractor* converts Flickr-specific descriptions into a well defined RDF description using Dublin Core vocabulary.

Fig. 4. Example: process of RDF content creation

Model. The core GroupMe! model is composed of four main concepts: *User*, *Tag*, *Group* and *Resource*. These concepts constitute the base for the GroupMe! folksonomy (cf. section 3). In addition, the model covers concepts concerning the users' arrangements of groups, etc. The *Data Access* layer cares about storing model objects.

Application logic. The logic layer provides various controllers for modifying the model, exporting RDF, etc. The internal GroupMe! search functionality is made available via a RESTful API. It enables third parties to benefit from the improved search capabilities, and to retrieve RDF descriptions about resources, even such resources that were not equipped with RDF descriptions before they were integrated into GroupMe!. To simplify the usage of exported RDF data, we further provide a lightweight Java *Client API*, which transforms RDF into GroupMe! model objects.

Presentation. The GUI of the GroupMe! application is based on AJAX [7] principles, and is highly modular and extensible. For example, the visualization of group elements is adapted to the content type (see Fig. 1). When creating or modifying groups, each user interaction (e.g. moving and resizing resources) is monitored and immediately communicated to the responsible GroupMe! controller with the effect that e.g. the actual size or position of a resource within a group is stored.

4.2 Ontologies and content creation

Almost every user interaction with the GroupMe! systems implies the creation of RDF content. Figure 4 illustrates the general process of RDF content creation. On the left the original resource – a Flickr photo in this example – is presented. It is described with a community-specific vocabulary instead of a well defined ontology. Hence, when integrating the resource into GroupMe!, this deficit has to be compensated. To do so, we use ontologies consistent with the type of resource. The photo in Fig. 4 can for the most part be described by applying Dublin Core metadata elements. For example, *dc:subject* represents the tag that was provided by Flickr. Which ontology to apply depends highly on the resource type and further on the content provider. To capture the process of grouping and tagging we rely on the *GroupMe! ontology*[8] which essentially models the *GroupMe! folksonomy* as defined in definition 4:

[7] http://www.adaptivepath.com/publications/essays/archives/000385.php
[8] http://groupme.org/rdf/groupme.owl

User (U): GroupMe! users are simply modeled as *rdfs:subClassOf* of *foaf:Person*.

Resources ($\breve{R}$): Resources have at least *dc:title* and *resourceURL* properties. Additionally they can be equipped with attributes of any other domain ontology. Grouping of resources is modeled via the object property *isInGroup*, which points to the *Group* instances the resource is included in. For tagging of resources the GroupMe! ontology provides an object property named *tagAssignment*, which refers to *TagAssignment* instances.

Groups (G): *Group* is a subclass of *Resource* and extends its superclass with an inverse property of *isInGroup*, namely *hasResource*.

Tag (T): The class *Tag* defines a functional property *keyword*. As future versions of GroupMe! should also aim on bridging from folksonomies to taxonomies, Tag is also equipped with a property *relatesToConcept*, which should refer to such concepts of domain ontologies that are denoted by the corresponding Tag instance.

TagAssignment (Y): This concept implements the assignment of a tag by a user within the context of a group and consequently has four object properties: *user*, *tag*, *resource* (inverse of *tagAssignment*), and *group*.

The result of applying the GroupMe! ontology is shown shortened in the right box of Fig. 4. Such semantically enriched resources, which evolve naturally while users are interacting with the GroupMe! system, can in turn be processed by other systems via the GroupMe! API.

5 Conclusions

The GroupMe! application is at the edge between Web 2.0 and Semantic Web and enables users to easily group and arrange multimedia resources they are interested in. We believe that this kind of interaction is enjoyable to use and will, in combination with the automatic capturing of the semantics of the users interactions, support the wide-spread use of RDF.

References

1. Kerne, A., Koh, E., Dworaczyk, B., Mistrot, J.M., Choi, H., Smith, S.M., Graeber, R., Caruso, D., Webb, A., Hill, R., Albea, J.: combinFormation: A mixed-initiative system for representing collections as compositions of image and text surrogates. In Marchionini, G., Nelson, M.L., Marshall, C.C., eds.: JCDL, ACM (2006) 11–20
2. Fielding, R.T., Taylor, R.N.: Principled design of the modern web architecture. In: Proc. of ICSE '00, New York, USA, ACM Press (2000) 407–416
3. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: BibSonomy: A social bookmark and publication sharing system. In de Moor, A., Polovina, S., Delugach, H., eds.: Proc. of ICCS '06, Aalborg (2006) 87–102
4. Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S.: Learning taxonomic relations from heterogeneous sources of evidence. In: Ontology Learning from Text: Methods, Evaluation and Applications. Frontiers in AI. IOS Press (2005) 59–73
5. Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., Su, Z.: Optimizing web search using social annotations. In: Proc. of WWW '07, New York, USA, ACM Press (2007) 501–510