Unlock Your Data: The Case of MyTag

Thomas Franz, Klaas Dellschaft, and Steffen Staab

University of Koblenz, 56070 Koblenz, Germany
{franz,klaasd,staab}@uni-koblenz.de
http://isweb.uni-koblenz.de

Abstract. The business model of Web2.0 applications like FaceBook, Flickr, YouTube and their likes is based on an asymmetry: Users generate content, Web2.0 application providers own, *(i)*, the access to user content, *(ii)*, the user profiles and, *(iii)*, user interaction data. We argue in this paper that such asymmetry disadvantages the users and prevents innovative applications. We demonstrate an application, MyTag, that is based on a layer for cross-application user profiling and personalization and that exploits web service access to user data. Presenting this application, we conclude that such applications offer additional value to users and usage of such applications on content generated by the users should not be at the disposal of the application provider, but should be a part of users' rights.

Keywords: User-generated content, Web2.0, Mash-ups, User rights.

1 Asymmetry of Efforts and Rights

The success of the internet and in particular of the Web as the most prominent application of the internet is based deeply on the variety of Web applications, stakeholders and users. In fact, this variety has been such a driving force that it is now taken for granted rather than a surprise.

We, however, argue in this paper that such a variety may be fragile and we need innovation of technology and applications as well as development of laws and regulations in order to maintain this variety.

The Winner Takes It All. For example, it has become visible in the Web search market that there is not much room for a variety of search engines. The market is dominated by one player eventually owning a near-monopoly. It has been argued reasonably that such a monopoly may create severe economic and societal problems [6].

User-generated Content, Personalization and Interaction. Until recently, the area of Web2.0 applications exhibited a very different picture. Applications such as Flickr, Delicious or YouTube have been developed by start-up companies creating a new variety of stakeholders and a new set of Web applications attracting site visits at enormous rates. However, (partial) acquisitions of several of the most successful of these companies have led to a situation where not only the index to Web content, but also the user-generated content itself as well as the interaction

J. Domingue, D. Fensel, and P. Traverso (Eds.): FIS 2008, LNCS 5468, pp. 117–129, 2009.

of users with this content is now owned by a few oligopolists. With recent, new types of applications such as Google docs, this development will be further accelerated.

Unlocking Your Data. The oligo-/monopoly in the field of applications is based on the monopoly of data ownership by the oligo-/monopolist — instead of data ownership by the users who generated the data. For example, it will be virtually impossible for an everyday user to unlock his own data from Gmail (Google's mail programme) in order to continue the usage of his own data in a different email client.

To substantiate the discussion, we present in this paper an example application we have developed, i.e. MyTag (`http://mytag.uni-koblenz.de`). MyTag demonstrates, *(i)*, the need of data ownership by users, *(ii)*, a new application allowing versatile usage of user-generated content and, *(iii)*, further requirements for a variety-rich internet of the future.

In the remainder of this paper, we first introduce our running scenario, based on which we will determine requirements for cross-application usage of content and personalization (i.e. profiles and user interaction). Section 4 describes the MyTag application from the user point of view, Section 5 the architecture of MyTag and Section 6 the personalization capabilities. We describe some lessons learned, before we conclude with a discussion of open technical, application and legal issues to be handled for the future of Web applications.

2 Scenario

The scenario is centered around two global travellers named Tim and Tony.[1] When Tim and Tony approach a town, they use their mobile phone and personal digital assistant (PDA), respectively, in order to access the internet and search and retrieve impressions about the town that may be found in images, videos and Web pages.

Tim also carries a digital camera and often puts images online to share his impressions with friends at home. He has an account for a web application for image sharing for that purpose. Tim likes the application as it is easy to use, enables him to organize his images by simple tags, and offers him to browse for similar images shared by others.

Tony has a digital camera too that she also uses to record short movies. Being interested in professional fotography, she has an account for a web application commonly used by fotographers where members can also discuss about images, e.g. camera settings required to take a photo. Besides that, the application enables sharing and organizing of photos similar to the application used by Tim. For her videos, Tony has an additional account for a web application supporting videos.

After travelling Tony and Tim compile a photo album consisting of the most spectacular places they visited together. As they always tag their photos with

[1] The scenario partly overlaps with the scenario developed by the W3C incubator group [1].

the name of the place where it was taken they can use the search functionality of the web applications they use to display such photos.

3 Issues and Requirements Collection

Investigating the scenario, a number of issues with current Web applications, Web2.0 applications in particular, may be noticed:

Lack of task focus: The *task* of the travellers is the retrieval of impressions about a town. As data are locked into several distinct applications, they need to unlock the data manually, because there is no cross-application and cross-media search facility available that could support such a task. A similar case is true for updates. When Tony adds a photo and a video taken at the same place, she needs to create the same geo tag at both the photo and video application. When Tony wants to add a tag *Tim* to photos and videos that show Tim, she needs to execute the same kind of update on both platforms. And also for sharing, Tim and Tony cannot work seamlessly with both photo applications in order to compile an album of their trip. Consequently, users try to avoid switching between applications as much as possible.

Lack of cross use of user-interaction data: Most of the applications they use do not personalize their search to reflect their preferences. Though some more sophisticated applications allow personalization at the level of the individual application (e.g., `http://www.bibsonomy.org`), the applications do not allow for moving the user-interaction and -personalization data across different applications. Consequently, users have to spend a multiple of user-interaction efforts for unoptimized interaction with different applications.

Lack of cross-application profiles: The applications they use do not allow them to define profiles that are applicable across applications. For sharing content about their travel, they have to explicitly maintain their profile in form of social network data and access rights at different applications. Consequently, users have to spend extra effort for maintaining multiple profiles (if possible at all).

In order to deal with these issues, we have captured the following user requirements. First, we consider traditional ones that exist for single applications:

Personalized Result Ranking: Given personal structures that provide personalized access to content, we derive the requirement for *personalized* ranking to increase task completion efficiency (here, in particular: search). For instance, personalized rankings enable Tony and Tim to find interesting things about a town more quickly as web pages that are more relevant to them will be ranked higher.

Second, we consider user requirements that are derived from the cross-application usage scenario:

Cross-Application Search: To support the task of searching, the travelers need a possibility for cross-application search and results display.

Cross-Application Ranking: Cross-application results display requires intelligent integration of results, e.g. a fusion of collections as discussed in [7].

Cross-Application Profile Management: The sharing of user profiles needs a corresponding cross-application profile management allowing for the sharing of profile data, such as social network data, contact data, login data, etc.

Cross-Application Personalization: Finally, the efficient interaction requires cross application personalization such that the individual application can adapt itself easily according to preferred application-specific and cross-application interactions by the user.

From this requirements collection we may recognize that full benefits from user-generated content in these Web2.0 applications can only be harvested if either one application provider owns a virtual monopoly on these applications and facilitates comprehensive interactions (such as Google does for Google docs) or we may have to come up with new types of applications that fulfill the special needs for cross-application search, ranking, personalization and profiling in a system of distributed applications.

4 MyTag

MyTag [2] is an example of a novel kind of cross-application platform we have built for experimenting and solving limitations of current Web 2.0 applications as discussed in Section 3. MyTag features cross-application search for images, videos, and social bookmarks including capabilities for profiling and personalization. As of now, MyTag integrates Flickr, YouTube, and Delicious to offer transparent access to the information provided by these applications (cf. Fig. 1).

On its start page, MyTag offers its users a single field where search terms can be entered. By default, all platforms currently integrated into MyTag are then searched based on the user input. Search results for each media type are presented as separate result sets in different columns that can be ranked by means such as popularity and creation date (cf. Fig. 1).

For every resource in the result sets, its title, a preview, and the creation date are shown. For bookmarks, the preview is a snapshot of the website that is provided by a mash-up with the service of `Snap.com`. The snapshot is shown by moving the mouse over the icon next to the title of the bookmark. Furthermore, for each resource its associated tags are shown. Clicking on the preview of a resource opens a new window that offers additional details about the resource and links to its occurrences on the integrated application, e.g. Flickr. Above the results, a tag-cloud is shown that summarizes the most frequent tags in the result set. The font size of each tag is proportional to its occurrence frequency.

In MyTag, clicking on a tag refines the previous search query by adding the tag supporting faceted search for disambiguation of queries. The user can easily explore different directions in adding or removing tags.

Two further modes of operation are available for registered users of MyTag. First, a mode in which only personal media are shown, i. e. only media one

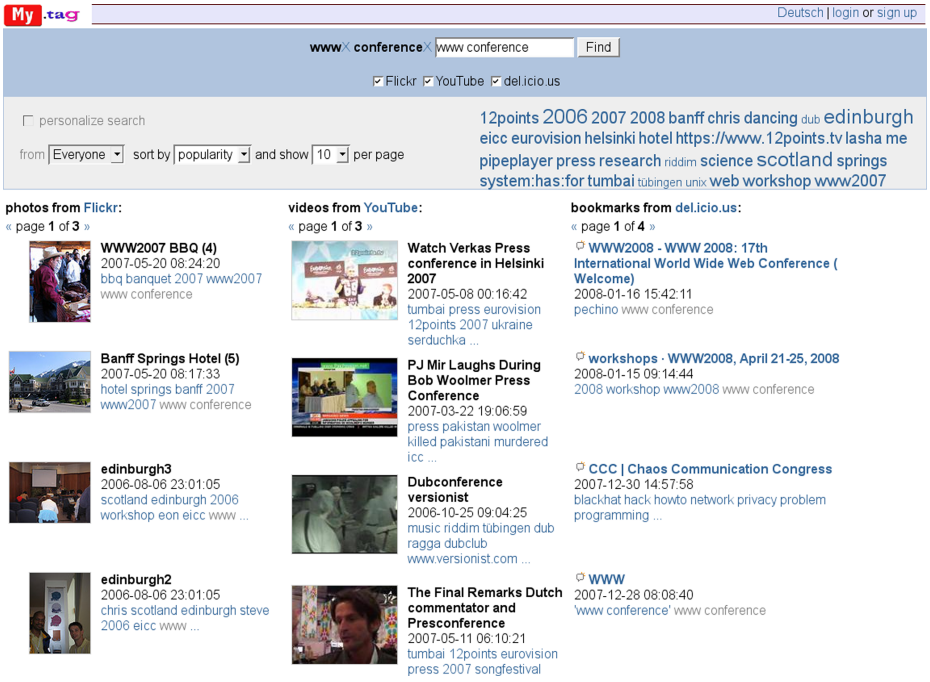

Fig. 1. Screenshot of a MyTag Search Result

contributed to any of the integrated applications. Second, a mode where search results are ranked based on the personomy of a user. Both search modes are explained in more detail in Section 6 that deals with personalization features.

5 MyTag Architecture

In the following, we present the conceptual architecture of MyTag and explain how that architecture supports the integration of existing web applications. Additionally, we point out conceptual improvements we foresee.

5.1 Layers of the Architecture

MyTag was developed utilizing the web-development framework *Ruby on Rails*[2] to benefit from the maturity, tool support, and lively user community of that framework. The MyTag architecture realizes the model-view-controller paradigm (MVC) distinguishing between three conceptual layers as indicated in Fig. 2. The view layer (shown on the top) is responsible for handling the interaction with the user such as rendering user interfaces and retrieving user input. The control layer in the middle implements logical operations and processes data from the model

[2] `http://www.rubyonrails.org/`

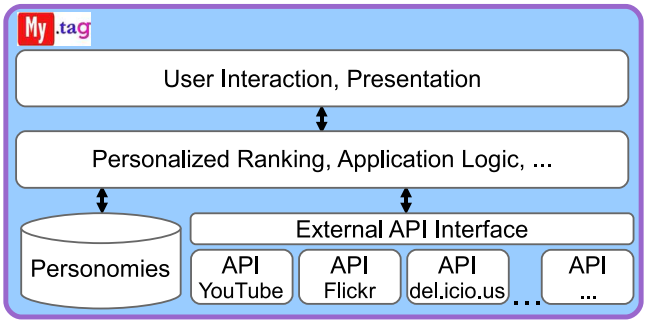

Fig. 2. MyTag Architecture

layer as well as user input from the view layer, e.g. by computing personalized rankings for a search term entered by a user. The model layer provides access to the information processed by the upper layers and consists of two major components: First, the interface to the local database that contains user profiles and personomies. Second, components that integrate applications into MyTag.

5.2 Application Integration

MyTag provides users with unified access to content they contributed to the Internet by the use of different web applications. The integration of such applications is based on the use of web-based application programming interfaces (APIs) that are provided by the tagging platforms integrated into MyTag. Web-based APIs are a common feature of Web 2.0 applications, intended to leverage system integration by enabling programmatic access to the data the applications provide. The utilization of such APIs confronts developers with common issues of integration such as mapping between data structures, data models, and API specifications as well as with further issues, because only a subset of user-generated data is available via the API.

Accordingly, for the development of MyTag, a mechanism was required that maps between the API of an integrated application and the data structures and method calls used by components of the controller layer (cf. Figure 2). MyTag provides a plugin architecture for the integration of applications that provides an abstraction of core aspects of tagging applications as we illustrate in Figure 3. The class `TaggingSystem` provides a common representation for integrated applications while subclasses extend it to provide application-specific details like access credentials. MyTag components communicate to integrated applications by means of common representations of application requests and responses. A `ListRequest` represents a search query that is distributed to integrated applications. It returns a `ListResponse` that contains for each of the integrated applications a `ResultList` pointing to single `Result` instances that provide a common representation for results from different applications. The abstraction of preview requests and responses, i.e. the request for details about a single

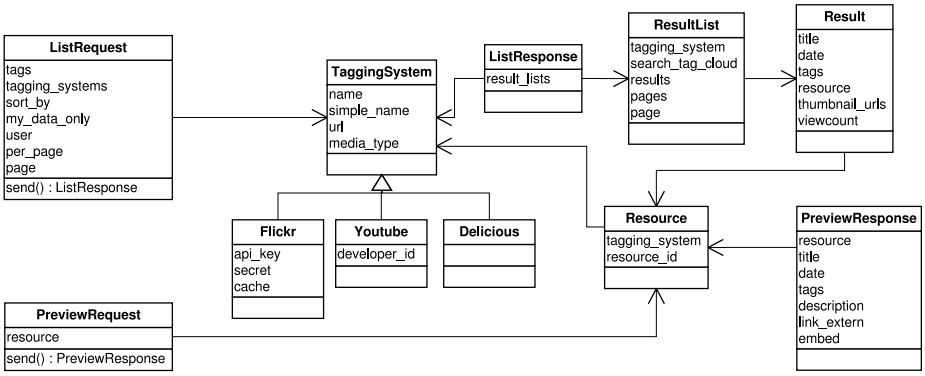

Fig. 3. Unified Model for System Communication

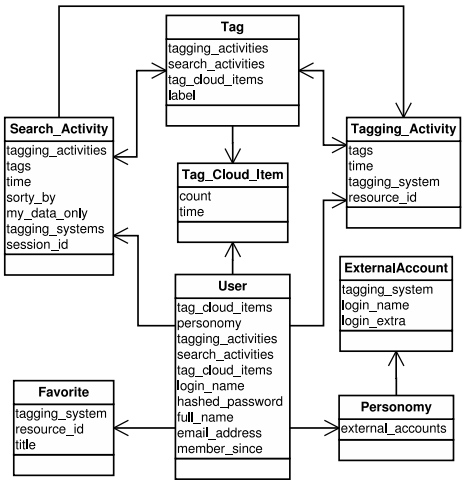

Fig. 4. Unified Model for User and Tagging Data

`Resource` such as an image on Flickr or a bookmark on del.icio.us, is modeled by the classes `PreviewRequest` and `PreviewResponse`.

Next to an abstraction for cross-application communication, MyTag components build upon a unified data model for users, resources, and taggings (i.e. the association of a tag to a resource by a specific user). Figure 4 shows the classes and properties in UML notation. The class `User` represents a MyTag user identified by its `login_name`. A user is associated to accounts on systems integrated by MyTag via his/her `Personomy`. Favorite resources – e.g. images, bookmarks, videos – are modeled by instances of the class `Favorite`. The class `Tag` models a tag that is identified by its `label`. Tags can be associated to users in three different ways: First, by a `Search_Activity` that models a user's search including the tags entered (`tags`), the applications considered for the search

(`tagging_systems`), and the time at which the search was executed (`time`). Second by a `Tagging_Activity`, that stands for a users click on a tag in a result listing of MyTag. Third, tags are associated to users by personal tag clouds that represent the most frequently entered or clicked tags. A tag cloud is represented by instances of a `Tag_Cloud_Item` that stores how often a tag was used by a given user (modeled by the property `count`) and when it was last used (modeled by the property `time`).

6 Personalized Access to User-Generated Content

In Section 4, we introduced two modes of operation that stand for different personalization features of MyTag. In the following, we explain how they are implemented.

6.1 Searching Personal Resources

A user's content is commonly distributed over multiple Web 2.0 applications, e.g. fotos are shared using Flickr, while bookmarks are organized using del.icio.us. To overview the personal content distributed over multiple applications, MyTag supports a cross-application search that considers personal content exclusively. The feature has been implemented reusing functionality already available in many Web 2.0 systems. For example, in Flickr it is possible to restrict a search to one's own photos. If a user wants to take advantage of this feature, he has to enter his account names for the different tagging sites he is member of and which are integrated into MyTag.

6.2 Cross-Application Self-adaptive Personalization

Next to searching personal content as described before, MyTag provides a mode that features *personalized* cross-application search. For instance, Tim and Tony benefit from that mode to retrieve personalized results for preparing their trip to a town. For this personalization approach two different techniques are needed: First, a representation of the user's interests is needed, e. g. in form of a user profile. Second, the user profile has to be taken into account during the ranking of the search results, i. e. the ranking algorithm has to be adapted in order to rank resources higher which are more interesting for the current user.

The approach implemented by MyTag is self-adaptive, i.e. implicit user feedback is used for building a user profile representing his or her interests in form of a personomy. The profile is automatically built based on the tags attached to resources the user picks from the result lists for his search queries. The personomy is modeled as a vector $\mathbf{p}$ of tag frequencies representing the previous search interests of the user. As it is based on implicit feedback no additional user effort is required to specify personal interests, however, implicit feedback may be less robust than explicitly stated interest and requires significant amounts of data to stabilize. The usage of implicit user feedback is a fundamental difference

to systems such as Flickr and Delicious, where personalization requires adding resources to the system.

With regard to personalized rankings, MyTag implements a ranking algorithm that combines information from the personomy and the tags assigned to resources of a result set. The tags of a resource are represented as a vector $\mathbf{v}$ of binary values indicating the presence of a tag. The rank r of a resource is computed by the scalar product of the two vectors: $\mathbf{r} = \mathbf{v} \cdot \mathbf{p}$. It is then used for ordering the resources based on their rank value.

While the first type of personalization is widespread, it is not the case for the second type of personalization. It is not available in the "main stream" tagging web sites like Flickr and Delicious but currently only implemented or planned for a few research oriented sites. For example, for the Bibsonomy application a personalized ranking using the FolkRank algorithm (see [5]) is currently considered.

6.3 Cross-Application Ranking

In its current version, MyTag integrates for each of the supported media-types only one tagging platform that is specialized on that type. Search results for different media-types are displayed in separate columns. This setting helps to avoid the fusion of result sets coming from different tagging systems into an overall result set.

To overcome this limitation of MyTag, we are at the moment implementing algorithms for the fusion of result sets. For this purpose, we are extending approaches for collection fusion (see [3] for an example) with capabilities for the personalized ranking of resources (see previous subsection). The new capabilities will directly be used for integrating further applications into MyTag like Bibsonomy, Connotea and Oneview.

Next to the ranking of results across applications, a representation of a user profile in order to implement personalized Internet utilization is limited to management of user identities so far. While the approaches discussed before are based on representations that are specific for the application type, namely tagging systems, other systems will require different representations. Both profile representations and personalization features that are applicable across applications will be needed to support personalized access to information served by a variety of Internet applications. The general user modeling ontology (GUMO) [4] is an example of initial work towards such a reusable profile representation.

7 Lessons Learned

Throughout the development of MyTag, we gathered insight into a variety of challenges that are to be handled in order to provide users with unified access to the information they contributed to the Internet. Challenges range from technological and conceptual constraints to policital, legal, and societal concerns and limitations of existing social platforms.

7.1 Political and Legal Constraints

Terms and conditions of Web 2.0 applications may restrict the implementation of applications that feature cross-application personalization. For instance, StudiVZ[3] forbids its users to access their own content by other means than the native user interface. Hence, applying MyTag on platforms like StudiVZ is legally forbidden while applying MyTag on platforms like Flickr, Delicious and YouTube does not have any legal basis — even if run from a user's PC.

7.2 Access Restrictions

While applications integrated into MyTag provide public APIs for accessing data they store, returned data may be heavily trimmed preventing the development of novel services. For instance, information about the ranking of results may be inaccessible.

In addition to such locking of data, we also experienced the blocking of API requests as another mechanism to restrict access. For instance, the use of some APIs is restricted to a certain number of requests per time unit.

7.3 Standardized APIs for Personalized Access

OpenSocial[4] is an API intended to define a standard interface to social networking platforms to ease the development of web-applications that exploit the data provided by such platforms. At the time of this writing, several platforms support OpenSocial and a JavaScript implementation is available for application development. The development of MyTag, however, could not profit from such a standardization effort: A common representation of user profiles is not foreseen by OpenSocial due to the different user representations on different platforms. Accordingly, we had to implement our own components for matching user profile information from different Web2.0 applications. Furthermore, OpenSocial does not provide interfaces for querying. However, such functionality is required to implement a cross-platform search as provided by MyTag. Moreover, interfaces for personalized search are missing. Accordingly, platforms that offer personalized access can only build upon *non-personalized* results that require further processing in order to adept them to users' preferences. Some applications, e.g. Facebook, provide programming interfaces for building add-ons that can access the data of the underlying application. While such approaches enable to exploit the infrastructure and available data, they lack support for cross-application functionalities and personalization.

7.4 Profile Management Support

Personalized web applications like MyTag require the interplay of different architecture components in order to implement personalization features. User interactions need to be tracked and filtered for generating and updating user profiles. At

[3] `http://www.studivz.net`
[4] `http://www.opensocial.org/`

the same time they may be accessed and refined by further user interface components as well as components that implement algorithms for personalized ranking and result-set merging. Current development frameworks provide limited support for the tracking, generation, exchange and provision of profile information for use by different architecture components and across Internet applications. While they commonly provide logging interfaces and session management components, they lack support for specific personalization tasks as mentioned before.

7.5 Decentralized Architectures

While MyTag enables transparent access to user-provided content spread over multiple web applications, it still does not change the fact that users have restricted access to the content they contributed. Approaches to distribute content in a decentralized fashion while enabling collaborative access are yet to be examined and constitute an open field of research. During the development of MyTag, we analyzed how to implement MyTag functionality by program code running in the web-browser of MyTag users. In such a decentralized architecture, profiling information could be gathered and stored on the users' clients giving them full-control over profile utilization and distribution. However, security policies implemented by standard web-browsers inhibit the development of cross-platform applications that run inside the user's web-browser. For instance, the *same-origin* policy ensures that program-code can only connect to the web server it was retrieved from. Thus, implementing JavaScript functions that access multiple web sites to implement a cross-platform search is not a viable solution.

8 Requirements for Future Internet Applications

We have presented MyTag, a running application that accomplishes cross-application usage of user-generated content, interactions and profiles in the domain of Web2.0 content sharing applications.

Technological Innovation. We have elucidated user requirements that led us to the development of MyTag capabilities. We have seen that these user requirements lead us to traditional as well as new aspects of needed research and development:

1. Traditional aspects of semantic data integration play a major role.
2. User profiles and user interactions need to be captured and securely managed as part of the user-generated data itself in order that they are applicable across applications. Research in this direction is virtually non-existent (see an exception [4,8]).
3. better support for user's tasks rather than solely the hosting of their data.

Application Innovation. However, we must also concede that the technological innovation achieved by MyTag is far from being sufficient. Though the focus of MyTag is on giving users access to their own content, profile and interaction,

MyTag itself is still a centralized application that locks such data. It is necessary to make such data directly available at the individual peers, e.g. by applications that build heavily on Ajax to establish communication between the web browser and Internet services while enabling corresponding local data hosting. This is currently technologically feasible, but very difficult, and hence aggravates the innovation of such applications. Assuming this next, will we be done and happy ever after?

New Legislation. The clear answer must be no. In spite of the fact that some applications provide Web Service access to one's own user data, the availability of these access mechanisms is far from guaranteed. Any successful application, even if run on the users' local PC, will need reliable and legal access to a user's own data.

To achieve this, technological and application innovation need to be accompanied by new laws and regulations. Currently, users of most Web2.0 applications concede the rights of their data, as well as the rights on their profile and their user interaction data to the application host leaving the possibility for hostile legal action by the application provider.

In order to allow for variety of applications in the future, users need to have a right on accessing their own content, their own profile and their own user interaction data in a machine processable manner — useful across applications and useful for the prospering of the future internet.

Acknowledgements. This work has been partially supported by the European projects "Semiotic Dynamics in Online Social Communities" (TAGora, EU IST FP6-2005-34721; `http://www.tagora-project.eu/`) and by "X-Media — Large Scale Knowledge Sharing and Reuse across Media" (EU IST FP6-26978; `http://www.x-media-project.org`). MyTag has been jointly developed by the authors of [2].

References

1. Anadiotis, G., Franz, T., Boll, S.: W3C Multimedia Semantics Incubator Group: Tagging Use Case (2007), `http://www.w3.org/2005/Incubator/mmsem/wiki/Tagging_Use_Case`
2. Braun, M., Dellschaft, K., Franz, T., Hering, D., Jungen, P., Metzler, H., Müller, E., Rostilov, A., Saathoff, C.: Personalized Search and Exploration with MyTag. In: Proceedings of the WWW 2008 Poster Session (2008)
3. Gauch, S., Wang, G., Gomez, M.: ProFusion: Intelligent Fusion from Multiple, Distributed Search Engines. Journal of Universal Computer Science 2(9), 637–649 (1996)
4. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: Gumo — the general user model ontology. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS, vol. 3538, pp. 428–432. Springer, Heidelberg (2005)
5. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information Retrieval in Folksonomies: Search and Ranking. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 411–426. Springer, Heidelberg (2006)

6. Maurer, H., Balke, T., Kappe, F., Kulathuramaiyer, N., Weber, S., Zaka, B.: Report on dangers and opportunities posed by large search engines, particularly google (September 2007),
http://www.iicm.tugraz.at/iicm_papers/dangers_google.pdf
7. Voorhees, E.M., Gupta, N.K., Johnson-Laird, B.: Learning collection fusion strategies. In: Proc. of SIGIR, pp. 172–179 (1995)
8. Wenning, R., Schunter, M.: The platform for privacy preferences 1.1. W3C Working Group Note 13 (November 2006)