Towards a Lightweight User-Centered Content Syndication Architecture

Vanda Lehel, Florian Matthes, Sheng Wei

Technische Universität München
Lehrstuhl Software Engineering betrieblicher Informationssysteme
Boltzmannstrasse 3
85748 Garching bei München
{lehel, matthes, weis}@in.tum.de

Abstract: This paper gives a structural overview of social software and its use for content aggregation and publication. We then extend this view to a user-centered perspective. We propose a lightweight XML-based content syndication architecture. It supports bidirectional information flow, as a proper extension and unification mechanism to facilitate the usage of several social software systems at once. Finally, we discuss benefits and limitations of this architecture.

1 Introduction and Motivation

Social software is a newly emerging class of community systems that allows individuals to form different communities to connect or to collaborate with each other based on their social context, and it effectively helps to facilitate interaction and collaboration among individuals [Burg05]. XML is widely used in social software due to its strengths, especially for networking, e.g. XML can be the major network payload and XML is the essential part of most related communication protocols [MD04].

In contrast to centralized systems like enterprise portals, the use of social software implies the membership of a user on different community platforms in several social contexts. This results in a distributed architecture of all personally published and available information. For accessing this information, many XML-based protocols and proprietary formats for content syndication and information exchange are used. Unfortunately there is a lack of a unified mechanism to interoperate with different community systems. Therefore it is difficult to achieve a user-centered integrated view on all available information. We address this problem by proposing in this paper an emerging form of information collecting, sharing and publishing that is also referred to as a user-centered social software system with an appropriate content syndication architecture.

In chapter 2, we present the fundamental concepts of the relevant kinds of social software systems and the XML-technologies they use. We then introduce the idea of user-centered social software in chapter 3. Chapter 4 explains our approach for an XML-based lightweight content syndication architecture and discusses it by means of current popular social software-related technologies. The paper concludes with a discussion of the benefits and limitations and an outlook on future research work.

2 Fundamental Concepts and XML-Technologies of Social Software

The use of XML-technologies has been established for several purposes in representing and exchanging resources managed by social software. In the succeeding sections, relevant concepts and functions of different kinds of social software, i.e. weblogs, social networking services, and object-centered social software are explained along with their related XML-technologies.

"*Weblogging*" is a common practice in social software to publish personal knowledge and information on the Web displayed in reverse chronological order as illustrated in [LMS03]. Its popularity results from the quality content and the immediate reflection of ownership to postings and comments in them in spite of the absence of centralized control [Paqu03]. Weblog-related technologies also foster selective syndication of published content and social networking among people.

RSS is an XML news syndication format [RSS05] for the exchange of different content, like postings and comments of a weblog. With RSS Enclosure, multimedia content can be embedded in the items as well. By subscribing to RSS feeds, users can remain up-to-date about the latest content by notification mechanisms. To link together related content from different systems, pingback [Ping05] and trackback provide mechanisms for the owner of a weblog to be notified about comments to his weblog in external systems.

Social networking services provide the opportunity to explicitly build up and manage contact relationships between individuals on a community platform to achieve viewing the path of indirect contacts. Well-established examples of social networking services are Linkedln [Link05] and OpenBC [Open05], both widely used for managing business contacts. However, information exchange and collaboration between different communities are difficult to achieve. An approach to address this problem is the XML-based FOAF [FOAF05] format to represent relationships between contacts and thereby describe the structure of the social network of an individual.

Object-centered social software is another class of online community platforms that focuses on allowing users to store, categorize and share a personal collection of objects. In contrast to taxonomic classification, freely chosen keywords (tags) are used as shared metadata and context information for collaborative categorization [HHLS05]. Widely adopted communities are Del.icio.us [Deli05] for bookmark sharing and CiteULike [Cite05] for bibliographic reference sharing, where people are grouped around shared objects.

3 User-Centered Social Software

Based on social software we propose an extension of personal information management (PIM) with an emphasis on social community platforms. Therefore, we define the new class of *user-centered social software* as a client software for *organizing information in* *a social way*, that is in the social context of a user. We call this kind of client software a *social organizer*.

In figure 1, a view on user-centered social software as an extension of PIM is shown. There, we depict a three-faceted view of content, context and contact that are managed using individual desktop tools, like text editing and image editing tools, feed readers or e-mail clients.

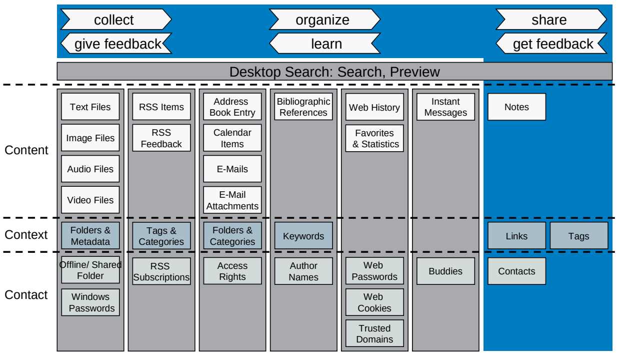

Figure 1: User-Centered Social Software as an Extension of Personal Information Management

Content is defined as the collection of different types of personal information objects like text and image files, address book entries and e-mails or instant messages. We distinguish between private content on the user's personal device (e.g. e-mails, personal documents) and content published on social software systems (e.g. postings, citations). By contact, we consider different individuals who are mainly peers in the social context of the user with whom he is sharing his personal information, e.g. RSS subscribers or buddies. The context links together content and contacts and thus defines how these objects are shared, most commonly either by shared folders or by shared categorization. When users publish and share their content, the context travels with the content in the information flow.

PIM tools like desktop search engines already support processes like searching or previewing by integrating content objects from different application "silos". Building on this successful approach, we propose to support processes for bidirectional information flow as follows (see fig. 1):

- *Collecting content* accessible through the social networks of a user.
- *Organizing content* based on the social networks and shared contexts of a user.
- *Sharing content and contexts* in a controlled way with selected contacts.
- *Learning from received feedback and providing feedback* to strengthen personal networks.

Further, we propose to consider semi-structured notes as a new kind of content objects, since in social software content is often represented by small fragments of information instead of complete digital documents like postings (see chapter 2) or annotations. The context in social software is built up from shared links and tags.

4 An XML-based User-Centered Content Syndication Architecture

Syndication represents a publish/subscribe mechanism, where the server (*publisher*) publishes the syndicated content (or feed) in standard formats for clients (*subscribers*). This syndicated content can contain both the content itself and related metadata. Syndication generally establishes an asynchronous multicast communication in which servers use push strategy to provide content, whereas RPC is based on synchronous and unicast communication in which clients use pull strategy to retrieve content. Therefore, a *syndication architecture* is a software architecture that models the information flows of such subscription services. In this chapter, we introduce our approach for a distributed content syndication architecture for user-centered social software.

4.1 Syndication Architecture for User-Centered Social Software

User-centered social software integrates available ad-hoc services and technologies to facilitate interaction and collaboration of existing social software systems. Figure 2 clarifies the related distributed content syndication architecture.

The "social organizer" in the middle is designed to let individuals manage all their information on local personal devices and social software systems uniformly. It acts as a universal information client that focuses on collecting personal information, effectively organizing information objects and their relationships, and publishing and sharing information in a controlled way. It can connect to several services at the same time for different purposes (e.g. content publication, information search). Therefore, the diversity of the underlying applications and services is transparent to the user, and barriers preventing individuals from using multiple resource locations are eliminated.

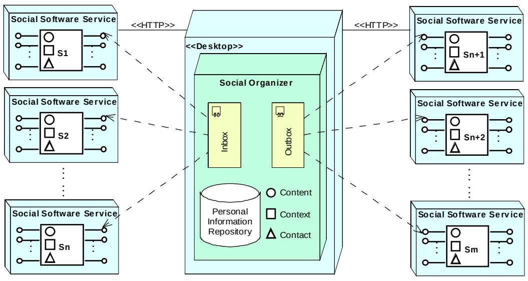

Figure 2: Distributed Content Syndication Architecture
for User-Centered Social Software

The social software services on the left represent external *content sources* of social organizers. By using different XML-based open formats and protocols, they offer different services such as subscription and feedback services. Therefore, users of a social organizer can retrieve the published content and provide feedback to content residing on that server. The social software services on the right act as *content sinks* of a social organizer which provide services like publication and administration. In this way, users of social organizers can publish and share their information in several communities. Users can create shared context information which is expressed using a shared vocabulary for the available content, e.g. interests, topics, goals. When the user shares and publishes content with his social organizer, the context travels with the content in the information flow to the related social software system. The inbox component of the social organizer (see fig. 2) is responsible for connecting subscription services to pull and filter out content and feedback information from community systems, while the outbox component supports publishing and sharing of information. Finally, HTTP is considered as the de facto communication standard.

4.2 Social Software Services

We have introduced social software and its related technologies in chapter 2. For a better understanding of the information flows between the social organizer and social software systems, we classify all currently available social software services into four categories in figure 3. In this diagram, content producers of social software systems are located on the left, while consumers can be found on the right. Producers and consumers can be either individuals or social organizers as their representatives.

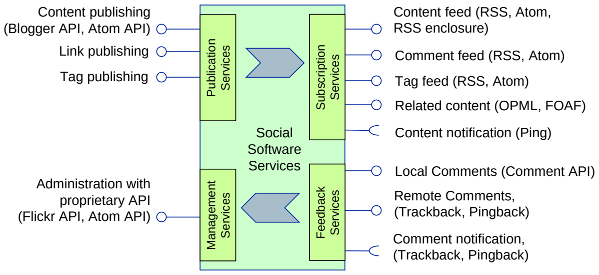
Content publishing (Blogger API, Atom API)
Link publishing
Tag publishing
Administration with proprietary API (Flickr API, Atom API)
Publication Services
Management Services
Social Software Services
Subscription Services
Feedback Services
Content feed (RSS, Atom, RSS enclosure)
Comment feed (RSS, Atom)
Tag feed (RSS, Atom)
Related content (OPML, FOAF)
Content notification (Ping)
Local Comments (Comment API)
Remote Comments, (Trackback, Pingback)
Comment notification, (Trackback, Pingback)

Figure 3: Classification of Social Software Services

Publication services: Social software systems act as centralized systems that collect information from individuals. Publication services with related APIs are used to publish content and the related context information, e.g. weblogs, links, tags.

Subscription services: Subscription services offer syndicated up-to-date or complete content in different standard feed formats, e.g. RSS or Atom, for clients to subscribe. These feeds represent a filtered view or organization of the content and can be distinguished as content feed, comment feed, tag feed, etc. Beyond that, systems can provide other related content, e.g. list of favorite weblogs of a user.

Feedback services: Comments to shared content can be given by visitors using different feedback services. For example, local comment services use comment APIs to attach the complete comment to the target content, while remote comments only provide references. Two kinds of remote comment-related services are available: the remote comment service and the comment notification service. Examples of such services are the weblog trackback and pingback functions.

Management services: Beyond the aforementioned services, each social software system provides different proprietary APIs for administration purposes (see fig. 3), since no standard exists.

In this section, we introduced a high level classification of the services provided by current social software systems. The challenges are that there is no standard for how to provide services, and there are different options for formats, protocols, communication mechanisms and APIs. These problems result in complexity in service implementation and difficulty in exchanging information between different social software systems. User-centered social software uses the bottom-up approach to support currently available services. The social organizer, as a unified information client, has built-in support for common related technologies, standards and protocols. It intelligently selects different wrapper components to interact with different social software services and it changes the adapters when the related service uses different formats and protocols. One social organizer can interact with many services at the same time, and different users have their own organizer to interact with a number of social software services. Thus a controlled, scalable, decentralized and dynamic network is established in the social context of each user.

5 Conclusion

In this paper, we propose user-centered social software as an extension of personal information management along with an appropriate XML-based content syndication architecture. In contrast to other information management solutions, user-centered social software creates immediate value to the individual by unifying the view on locally stored personal information and on content available through social networks with advantages of fast and offline information access, server-side services, ownership of information, and controlled sharing. On the other hand, user-centered social software still has some limitations. Unlike online services, the social organizer requires desktop software distribution and the underlying database systems and available GUI toolkits have to be taken into consideration. To support integration with several social software services, proprietary interfaces to desktop applications need to be added as well.

References

[Burg05] Burg, T.: Einführung zur Social Software, http://randgaenge.net/gems/Texte/html/SocialSoftware, Access: May 2005.

[Cite05] CiteULike, http://www.citeulike.org, Access: May 2005.

[Deli05] Del.icio.us, http://del.icio.us, Access: May 2005.

[FOAF05]FOAF Specification, http://xmlns.com/foaf/0.1/, Access: June 2005.

[HHLS05]Hammond, T.; Hannay, T.; Lund, B.; Scott, J.: Social Bookmarking Tools (I): A General Review, *D-Lib Magazine* 11(4), April 2005.

[Link05] Linkedin, http://linkedin.com, Access: May 2005.

[LMS03] Lehel, V.; Matthes, F.; Steinfatt, K. Weblogs als ein innovatives Instrument des betrieblichen Wissensmanagements. Tagungsband der Konferenz Mensch & Computer 2003, Teubner Verlag, Stuttgart, 2003.

[MD04] Megginson, D: Imperfect XML, Prentice Hall, 2004.

[Open05]Open Business Club, http://www.openbc.com, Access: May 2005.

[Ping05] Pingback Spec., http://www.hixie.ch/specs/pingback/pingback, Access: May 2005.

[Paqu03] Paquet, S: Socio-technological approaches to facilitating knowledge sharing across disciplines, PhD Dissertation, Université de Montreal, 2003.

[RSS05] RSS 2.0 Spec., http://blogs.law.harvard.edu/tech/rss, Access: May 2005.