The "Wisdom of the Crowd Pattern": A Person-centric data aggregation approach for Social Software

Peter Lachenmaier, Florian Ott
Cooperation Systems Center Munich, Bundeswehr University Munich
{Peter.Lachenmaier, Florian.Ott}@kooperationssysteme.de

Abstract

Driven by the success of Social Software in private and enterprise settings (Web 2.0 / Enterprise 2.0) a lot of specialized services have moved mainstream over the last few years. Thus, the number of services one single person uses in cooperative settings for different dedicated purposes, e.g. joint document editing, group calendars, microblogs or Social Networking Services (SNS) has highly increased. One problem of that development is that the distribution of digital activities over different platforms and services makes it hard to stay informed about the activities of others because of their technical separation. As exactly this kind of awareness information is one of the key features of Social Software, because it allows implicit coordination for cooperative work settings this leads to the demand of a flexible data integration solution for social services. In this paper we describe our person-centric data integration approach that is aligned to the special characteristics of data from Social Software. It is inspired by the way humans are collaborating in the social web and therefore called "Wisdom of the Crowd Pattern". This approach is technically implemented and integrated in the CommunityMashup.

Keywords: CommunityMashup, Aggregation, Person-centricity, Data Integration

1. Motivation

Different studies over the last years have shown that Social Networking Services like Facebook or Microblogging tools like Twitter are currently changing the way we are using the Web (e.g. [1]). Individual content sharing and social collaboration are becoming the most important activities [2]. In conjunction with the intensive use of activity streams the interconnectedness of people and content gains in importance. Web 2.0 platforms as well as their counterparts in organizational contexts (Enterprise 2.0) have emerged from prevailing content integration services to systems that support human relationships, and provide a technical foundation for inter-person-integration. Thus, one main difference in comparison to earlier research in the field of Computer Supported Cooperative Work (CSCW) where content was the central element (e.g. [3]) is the recent focus on individuals and their activities [4]. In contrast to earlier Groupware Social Software mainly relies on the individual visibility of people's activities around content instead of detached information objects. Wiki pages, blog posts and status updates in Social Networks or microblogging platforms are popular representatives of this new individual related information objects in Social Software. By reconnecting content to the corresponding authors, and thereby giving them a visible "natural" identity, Social Software increases awareness, and leads to better socio-technical integration [5].

Awareness as information about the activities of (potential) interaction partners plays an important role for efficient collaboration. In general, awareness can be defined as "an understanding of the activities of others, which provides a context for your own activities" [6]. As one of the key success factors of the Web 2.0 is to make the individual activities of people transparent for friends, followers or other groups of interest, Social Software can on the one hand help to increase awareness. On the other hand we are facing a specialization of different social services. Most of them have measurable benefit for their users with a broad reach and large amount of users in general. But they also have their own identity management and thereby their own user base, although the people using these platforms have only one "natural" identity. This leads to situations where people are using different social services depending on the desired task or the group of other people they want to share information with. Even if they do not want to use several services, they are forced to for staying informed about the activities of all persons they are interested in. The missing support for content sharing across different services results in redundant cross-posts distributed via different services or platforms to reach a desired group of people.

In order to benefit from the better awareness support of Social Software without having to post and / or consume information redundantly in different platforms we are facing the need for a flexible data integration solution that enables the aggregation of data from different social services. One of the most important characteristics of such an integration solution is that it

always retains a unified person-centric perspective in which each (natural) individual is only represented once in the overall dataset independent of how many different virtual identities for separate services and platforms he has. We have proposed the CommunityMashup as integration layer for data from different distributed social services that focuses this need [7]. Potential usage scenarios for our approach range from situations where **one** person is interested in the activities of **one** person (himself or another one) to situations where a **group** of people is interested in the activities of (another) **group** of people including combinations of these two extremes. From a more technical view this means single- and multi-user requirements at both ends of the integration stack. A technical solution therefore has to deal with accessing data of different sources depending on defined access rights and has to keep up that access rights while combining it with information of other services.

Based on the technical foundation of the CommunityMashup and its model-driven development approach [7] this paper presents a new person-centric data aggregation pattern that helps to successively combine and enrich individual profiles with additional pieces of information from different social services (Chapter 4). Before that we start with a short introduction to selected related work (Chapter 2) and a presentation of the overall CommunityMashup context in Chapter 3. We conclude this paper with a short outlook to future work in Chapter 5.

2. Related work

Most of the currently available aggregation platforms and mashup systems depend on the "pipes and filters" approach. This architectural pattern [8] sees data as streams that can be filtered and transported by pipes. For example [9] contains a good overview of systems using this approach. While pipes have a lot of advances like e.g. easy processing, good extensibility and easy configuration in small scenarios, bigger configurations are getting very complex, particularly when feedback loops are allowed. Advantages of the data federation approach in contrast to the pipes and filters approach are outlined also in [9]. A dedicated person-centric integration approach is presented in [10] which is implemented in a desktop application. The solution aims to combine the profiles of a person from different social networking sites and creates an aggregated stream of friend activities from a single-user perspective.

Enabling multi-user access to aggregated data requires taking care of data access rights. In settings that include numerous heterogeneous source systems an overall access model is needed. A secure version of the pipes and filters approach is presented in [11]. It adapts the well-known RBAC[1] [12] pattern for the filters. This enables the control during the aggregation process. To also enable the control of access after the aggregation process the aggregated data need to carry access information. An authorization and privacy model for semantic web services with semantic data annotations is proposed in [13]. Another more flexible annotation based access control model with focus on social and cooperative systems is proposed in [14].

For privacy enabled mashup applications authentication and authorization plays an important role. Different authentication models are discussed in [15] and a new user-centric authentication and privacy control mechanism is derived. Another approach, which lets users completely control access to their data, is provided in [16]. While these approaches promise better privacy they have the big disadvantage of requiring their implementation by all data providers. Most of the Web 2.0 services providing access to their data controlled by OAuth [17] that supports basic functionality for user managed data access. Due to its wide spread usage integration solutions are forced to use this mechanisms and if necessary extend it with additional privacy support.

The main problem beside the missing person-centricity is that none of the current approaches has dedicated support for multi-user scenarios, which is one of the main goals of the CommunityMashup.

3. The CommunityMashup

We detailed the whole system architecture and the development approach of the CommunityMashup in [18]. Whereas this paper shows the aggregation concept used inside the system. In this chapter we provide a short overview of the CommunityMashup as foundation for the aggregation concept presented in Chapter 4.

The CommunityMashup is a flexible integration solution for data from social services and provides features like application frameworks with offline data access for different platforms. In contrast to existing mashup solutions we aim to provide unified and aggregated information based on a person-centric data model. One main idea behind the person-centricity of this model is the wish to integrate social data that naturally belongs to a person or an organization, but is artificially distributed over different services in the web.

Figure 1 depicts the layered overall architecture of the CommunityMashup containing a few exemplary external services at the bottom and abstract mashup components in the middle, which are responsible for data unification, aggregation and filtering. The top layer shows mobile-, web- and desktop-clients as three different exemplary consumer applications representing

[1] RBAC: Role-Based Access Control

stereotypical usage scenarios for clients that need "mashed-up" data. Therefore the CommunityMashup provides specific high level interfaces.

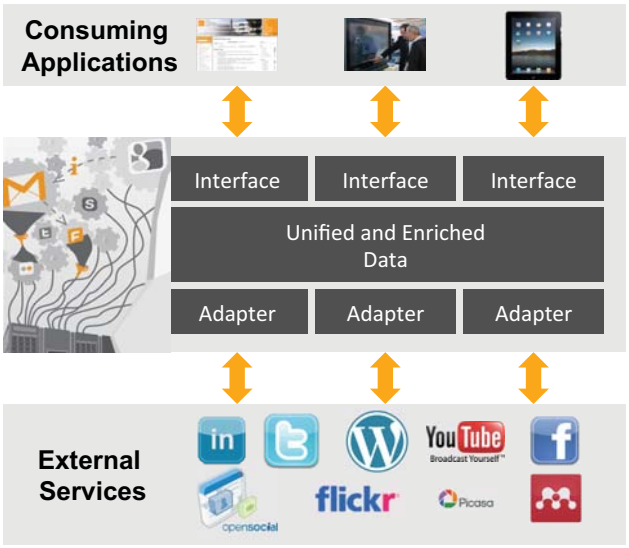

Figure 1: System Overview

Figure 2 shows the core elements of the mentioned internal person-centric data model. The key information objects person, content and organization are derived from existing models like SIOC [19], FOAF [20], the internal model of SocConect [21] and the examinations in [22]. We limited our core model to the most important entities of Social Software. The central element is the person, which can be grouped in organizations, and author or contribute to content. Organizations and content can be structured hierarchically (parent relation). Additional information can then be assigned to the information objects by rich attributes, tags, meta-tags and categories. Categories in comparison to tags can be modeled hierarchically. Tags, categories and rich attributes carry information that is directly gathered from external services whereas meta-tags are specific for concrete scenarios.

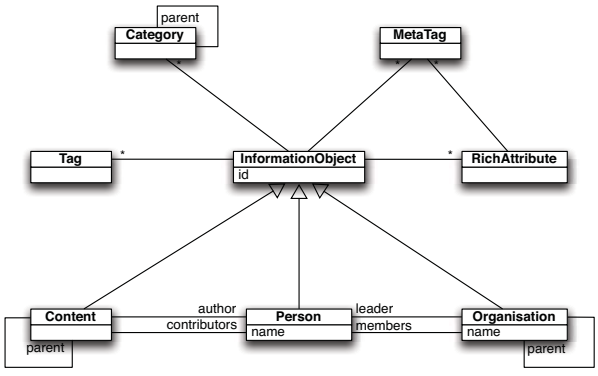

Figure 2: Core elements of the data model

The whole CommunityMashup is built on a service-oriented architecture and developed using a model-driven approach. The architecture especially contains specific source components for the connection to external services. Together with the data unification based on the presented data model this forms a flexible and powerful aggregation solution.

4. The Wisdom of the Crowd Pattern

It is a well-known practice in computer science to adopt things from the nature. With the "Wisdom of the Crowd Pattern" (WotC) we are adopting the way in which people generally work with data in the web for a new data aggregation mechanism. It is mainly based on the idea that people collaborate with a desired goal on a shared set of a data by fulfilling a number of more or less coordinated tasks. In the next sections we will give an overview over the mapping of this idea to a technical aggregation mechanism and outline how privacy is handled within it.

4.1. Definition

The "Wisdom of the Crowd Pattern" is an approach for the integration and combination of selected data from several different social services into one shared dataset. It especially addresses the specific characteristics of data from social services and how people are using these services.

4.2. Shared dataset

From an architectural point of view our approach is inspired by the shared repository pattern [8]. Figure 3 shows a schematic illustration how it is used in the CommunityMashup. First of all we use a central component (the shared dataset) for data exchange between system components. The shared dataset component completely relies on the person-centric data model presented in Figure 2 (Chapter 3). The technical solution is created by using model-driven techniques [18] to allow possible data model evolutions without manual changes. Furthermore the single instance of the shared dataset component provides the advantage to easily maintain data integrity and persistency.

Besides the shared dataset Figure 3 shows a few more components to describe the basics of the WotC pattern. At the top of the picture is an exemplary application component representing concrete applications that can consume aggregated information. Therefore the CommunityMashup provides high-level access, write and search methods. The more important components from the aggregation perspective are the source connection components. They are responsible for fetching data from external services and transforming it into the internal data schema as well as for the inverse direction. Figure 3

shows two source connection components that are connected to the same external system. This represents the fact, that an aggregated joint dataset is not created by only one single person. In multi-user settings different people providing individually pieces of information using several distributed services to create the complete dataset.

The use of the independent source connection components furthermore enables the aggregation process to be independent of single- or multi-user scenarios. For the aggregation process it makes no difference if a number of source connection components representing the different profiles of one single person in different services or if the source components representing a group of persons with their profiles to one ore more services. All of the system components from Figure 3 are implemented as small single services in the CommunityMashup. These services collaborate like humans do in order to build the joint data set.

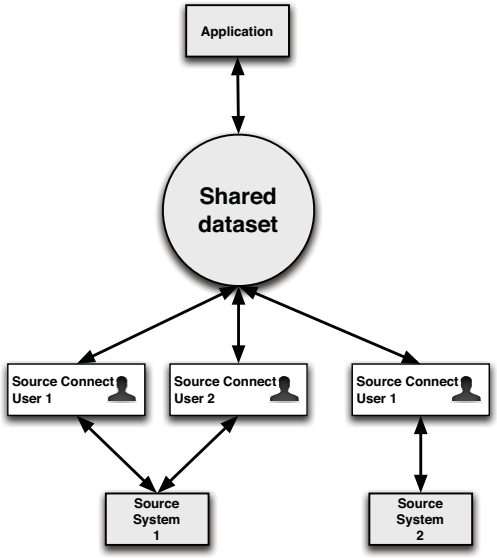

Figure 3: Shared dataset access

This first overview has only provided insight how data is logically gathered from the different system components but does not describe how single data entities are created, combined or accessed and how the control-flow in the aggregation process is directed.

4.3. Aggregation tasks

The source connection components introduced in the previous section fulfilling several tasks to perform the aggregation process. The overall process that defines which source components contribute and which dataset will be finally created is defined and controlled by a central configuration. This technical configuration in the WotC pattern is similar to the common goal people have when they are collaborating with each other on a specific topic. Of course this final goal may change during the collaboration process.

Figure 4 shows the concrete aggregation tasks that need to be performed by every source component. The first step in the aggregation process is the initial filling of the shared dataset **(Fill dataset)**. Like persons providing the information they want to share about a specific topic, a source component fetches data from the external service and adds it to the shared dataset. This can be handled e.g. by executing a search query via the API of the service and then transforming the data into the common data schema. After that the external service needs to be called periodically to retrieve new or changed elements to keep the shared dataset up to date **(Update dataset)**.

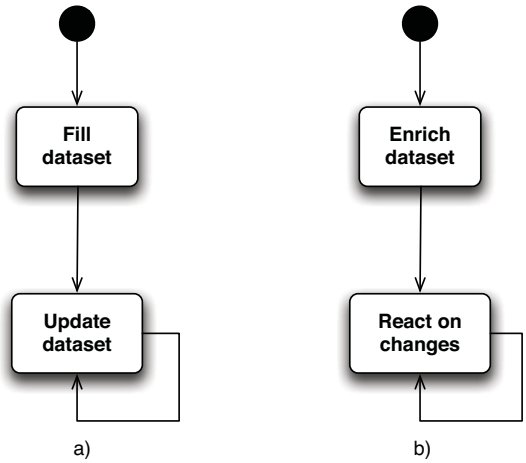

Figure 4: Tasks in the aggregation process

The second main task of source components is the enrichment of the data that is already contained in the shared dataset, because it has been provided by other source components that "started" their work before the new source was added to the mashup configuration. This fits to processes where new members are joining a collaboration team. First of all they can look at the already existing work and extend it with their additional knowledge. In the aggregation process this means iterating over the elements already contained in the dataset, identifying elements that can be enriched, calling the external service, transforming the query responses to the internal data format and adding them to the joint dataset **(Enrich dataset)**. After having done that for the whole dataset once, for the future only a reaction on changes to the joint dataset triggers an update mechanism **(React on changes)**. Possible changes are e.g. the creation, deletion or change of elements. This also contains the write back of changes to the external service if they support that. To enable the easy observation of the shared dataset and therefore the flexible execution of the aggregation tasks described above we integrated an event mechanism into the CommunityMashup.

If a mashup for example consist of two source components and the first component is providing data from a contact repository, it would fetch all names and email-addresses from that service, transforms it to person objects (cf. Figure 2) and adds them to the shared dataset. If a second source was added for e.g. Gravatar[2], it would look at all email-addresses in the enrichment task, fetch the corresponding profile images and add them to the related person objects. If a new contact is added to the repository in this example, the first source component would recognize this in the update loop and add a corresponding person object to the data set. This would also triggers a change event that would activate the second source component to make it add the matching profile image via the Gravatar API.

The aggregation tasks described above have the advantage that every single source component can be started, stopped, paused and reactivated independently during the whole mashup lifecycle while keeping a consistent dataset. Stopping a source triggers the deletion of all added data while pausing means that added data remains in the dataset. After reactivation of a source component it reacts in the update loop on the activities that happened in the external service. It also performs the enrichment tasks again to transfer the changes that happened to the aggregated dataset during the pause.

4.4. Privacy management

As outlined in the motivation privacy plays an important role for all social services and therefore also for systems aggregating data from those services. The management of privacy can be divided into three layers.

- The end user application has to provide a suitable user management to control the access to the data provided by the integration layer.
- The provided data must contain sufficient meta-data to describe the access rights.
- The authentication process with the external services needs to provide methods for users to allow and revoke access to personal information.

The first two layers of privacy are not necessarily required in public settings as well as in private settings where the aggregation process is directly performed on the end user's client device. Because thereby the first layer is highly dependent on the end user application, we will not discuss it in detail here. The second layer can be easily integrated in the CommunityMashup by providing special meta-tags with access role information for data objects. This can be done on the granularity level of (rich–) attributes (cf. Figure 2).

As stated in the previous sections the source connection components handling the data access between the CommunityMashup and the external services. Most of the current social services provide authentication mechanisms based on OAuth. These enable users to control access rights completely within the boundaries of the used services. Applications (in this case the source connection component of the CommunityMashup) can request access to personal data and the user can grant or deny this access. Once granted access lasts until the user revokes it.

The most important requirement for OAuth settings in multi-user mashup scenarios is that applications which have been granted access to personal data handle that access rights with care. Integration solutions need to guarantee that pieces of information are only made available to consumers that are qualified and immediately remove all pieces of information for which a user has revoked the access rights at the external service. In the CommunityMashup aggregation approach new data objects are created during the enrichment process (cf. Figure 4) depending on the existence of specific information in the source system. These new data objects need to inherit all access right information of the original source objects. As users can revoke access to their data it must be furthermore ensured that all data corresponding to that account is removed from the shared dataset. This also affects all pieces of information created during the enrichment process. If for example an image of a person was added based on an email-address that has been removed due to missing access rights, also the image needs to be removed. In the CommunityMashup all source connection components therefore are not only responsible for gathering data from external services, they also ensure the privacy during the whole lifecycle.

5. Conclusion and future work

In this paper we presented the Wisdom of the Crowd pattern as a novel approach for aggregating data from different distributed social services. The approach has been created for multi-user scenarios and is inspired by the way people collaborate in the web. This approach enables privacy aware handling of data from Social Software. It is prototypically implemented and integrated in the service environment of the CommunityMashup and allows a very flexible and easy configuration process.

As we see promising results in our first prototypical implementations we will evaluate our approach in bigger application scenarios with end user participation. This contains the research project elisa for elderly people were relatives are able to share awareness information with them across system boundaries [23]. Furthermore we are working on dedicated extended functionality that allows operations on aggregated objects that can be directly replicated to the external services. This contains e.g. operations like changing, commenting or rating.

[2] http://www.gravatar.com

6. References

[1] D. M. Boyd and N. B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication*, vol. 13, no. 1, 2007.

[2] J. Ganesh and S. Padmanabhuni, "Web 2.0: conceptual framework and research directions," in *Proceedings of the 13th Americas Conference on Information Systems (AMCIS 2007)*, 2007, pp. 1–9.

[3] T. Rodden, "A survey of CSCW systems," *Interacting with Computers*, vol. 3, no. 3, pp. 319-353, Dec. 1991.

[4] D. Gillet, S. E. Helou, C. M. Yu, and C. Salzmann, "Turning Web 2.0 Social Software into Versatile Collaborative Learning Solutions," in *First International Conference on Advances in Computer-Human Interaction*, 2008, pp. 170-176.

[5] C. Lampe, N. B. Ellison, and C. Steinfield, "A Face (book) in the crowd: Social searching vs. social browsing," in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 2006, pp. 167–170.

[6] P. Dourish and V. Bellotti, "Awareness and Coordination in Shared Workspaces," in *Proceedings of the 4th ACM Conference on Computer-Supported Cooperative Work (CSCW'92)*, 1992, pp. 107-114.

[7] P. Lachenmaier, F. Ott, A. Immerz, and A. Richter, "CommunityMashup. A flexible social mashup based on a Model-Driven-Approach," in *Proceedings of the 12th International Conference on Information Reuse and Integration (IRI)*, 2011, pp. 48-51.

[8] P. Avgeriou and U. Zdun, "Architectural Patterns Revisited – A Pattern Language," in *Proceedings of 10th European Conference on Pattern Languages of Programs (EuroPlop 2005)*, 2005, pp. 1-39.

[9] J. López, F. Bellas, A. Pan, and P. Montoto, "A component-based approach for engineering enterprise mashups," *Lecture Notes in Computer Science, Web Engineering*, vol. 5648, no. 2, pp. 30–44, 2009.

[10] Y. Wang, J. Zhang, and J. Vassileva, "A User-Centric Approach for Social Data Integration and Recommendation," in *3rd International Conference on Human-Centric Computing*, 2010, pp. 1-8.

[11] E. B. Fernandez and J. L. Ortega-Arjona, "The Secure Pipes and Filters Pattern," in *20th International Workshop on Database and Expert Systems Application, DEXA'09*, 2009, pp. 181-185.

[12] D. F. Ferraiolo, J. A. Cugini, and R. Kuhn, "Role-based access control (RBAC): Features and motivations," in *Proceedings of the Annual Computer Security Applications Conference*, 1995.

[13] L. Kagal, T. Finin, M. Paolucci, K. Sycara, and G. Denker, "Authorization and privacy for semantic Web services," *IEEE Intelligent Systems*, vol. 19, no. 4, pp. 50-56, Jul. 2004.

[14] P. Nasirifard and V. Peristeras, "Uncle-Share: Annotation-Based Access Control for Cooperative and Social Systems," in *On the Move to Meaningful Internet Systems: OTM 2008*, vol. 5332, R. Meersman and Z. Tari, Eds. Springer Berlin / Heidelberg, 2008, pp. 1122-1130.

[15] Y. Wang and J. Vassileva, "A User-Centric Authentication and Privacy Control Mechanism for User Model Interoperability in Social Networking Sites," in *Workshop on Adaptation and Personalization for Web 2.0, UMAP 09*, 2009.

[16] M. P. Machulak, E. L. Maler, D. Catalano, and A. van Moorsel, "User-managed access to web resources," in *Proceedings of the 6th ACM workshop on Digital identity management - DIM '10*, 2010, p. 35.

[17] E. Hammer, D. Recordon, and D. Hardt, "The oauth 2.0 authorization protocol," *draft-ietf-oauth-v2-18*, 2011. [Online]. Available: http://tools.ietf.org/html/draft-ietf-oauth-v2-25. [Accessed: 01-Apr-2012].

[18] P. Lachenmaier, F. Ott, and M. Koch, "Model-driven development of a person-centric mashup for social software," *Social Network Analysis and Mining*, online first, 2012.

[19] J. Padget et al., "The SIOC Project: Semantically-Interlinked Online Communities, from Humans to Machines," *Lecture Notes in Computer Science, Coordination, organizations, institutions and norms in agent systems V*, vol. 6069, pp. 179-194, 2010.

[20] D. Brickley and L. Miller, "FOAF Vocabulary Specification," 2010. [Online]. Available: http://xmlns.com/foaf/spec/. [Accessed: 13-Jan-2012].

[21] Y. Wang, "SocConnect : a social networking aggregator and recommender," 2010.

[22] B. Bazzanella, P. Bouquet, and H. Stoermer, "Top Level Categories and Attributes for Entity Representation," Trento, 2008.

[23] M. Burkhard, A. Richter, and M. Koch, "Ubiquitäre Benutzerschnittstellen für die Interaktion unter Senioren," in *Workshop-Proceedings der Tagung Mensch & Computer*, 2011, pp. 301-308.