Reactive Binders: A Framework and Prototype for Team-Oriented Web Work

Sven Rizzotti
Department of Computer Science
University of Basel
Basel, Switzerland
Sven.Rizzotti@unibas.ch

Helmar Burkhart
Department of Computer Science
University of Basel
Basel, Switzerland
Helmar.Burkhart@unibas.ch

***Abstract*—The fact that collaboration tools are often ignored to share information may lead to inefficiency and it is a challenge to change that situation. We believe that the entry costs into collaboration tool usage needs to be lowered. Users like a smooth shift from well-known metaphors and prefer models designed for simplicity. We introduce the Reactive Binder (RB) framework, which is based on the common known office model of shelves, binders, and registers. Our binders and shelves, however, are virtual and shared by web teams. RBs targets for reactiveness i.e., document binders are active entities that can react on events. With the RB framework and the PROBINDR prototype, we propose a collaboration solution with low entry costs and claim better user acceptance in distributed teams.**

***Index Terms*—Online collaboration, Data Cloud, Social Sharing, Interactive IR**

I. COLLABORATION TODAY

Today, professional collaboration is heavily based on information technology support. Manifold tools that address collaboration aspects exist and can be categorized as follows:

- *Communication tools* help in direct, personal information exchange and include simple instant-messengers as well as complex audio- and video-conferencing-Systems.
- *Coordination tools* help in management tasks and include project-management tools, work grouping tools, but also mind mapping systems for project preparation phases.
- *Data sharing tools* provide an infrastructure for collective storage. This can be either at a coarse-grain-level such as a file sharing-tool, or at a fine-grain level like bookmark managers for web links.
- *Document processing tools* support the preparation and discussion of shared documents such as collaborative writing and reviewing services, or all kinds of wikis.

A variety of commercial and free products exist [1], [2], but a majority of computer users are still ignoring their existence or do not know about them, which causes inefficiencies. The following list is typical for today's practice.

- *Email, email, email:* Email is still the #1 communication tool, because it is cheap, easy to use, works asynchronous and is universally usable [3]. On the other hand, email has a lot of drawbacks such as information overload, loss of context, content inconsistency, missing authenticity, and non-guaranteed delivery [4], [5]. Therefore, organizations that use email as teamwork and collaboration tool have to manage several obstacles. Email is not only inefficient, but must be considered harmful. Especially, email attachments would require that all receivers immediately delete pointers to older copies or even delete local copies when a new version of a document arrives. Moreover, if new members join a team, all information previously distributed must be resent. Both use cases are not only time and storage-inefficient, but also error-prone. Usage of a data-sharing tool can overcome such problems, but may face issues as mentioned for cloud space below.
  Many users still use email as a generic collaboration solution, even if it has been shown that it is not efficient [6], [7].
- *Shared drives / cloud space:* Shared drives within a single organization can be an effective way to get access to documents. However, the organization of documents and the notification of updates of these documents requires organization policies, which are hard to accomplish. If the filing of a document happens unnoticed, the document might getting stored in another location, with a different name and in a different version. This leads to inconsistent information.
  Another challenge is the freedom of file organization. It might look as an advantage that file systems can be organized individually, but the indefinitely expandable length and depth of hierarchies may become confusing and overcharged. It has been shown by [8] that nine is the maximum number of items that allows effective information processing.
  As soon as collaboration crosses organization borders, it becomes even more difficult to set up a working environment. The burden to create a virtual private network and firewall settings are other hurdles.
  The separation of files and comments cannot be solved with a shared drive alone. Users tend to comment on files or clarify discrepancies, but this process happens uncoupled. This leads to a detached and non-traceable evolution of a filesystem. After a while it becomes impossible to see how a filesystem grew and how the collaborators contributed.
- *Wikis:* Wikis are meant for collaboration and are a great

 ISBN: 978-1-61208-206-6

way to share and work together. Nevertheless, wikis need a certain syntax understanding and for people with little IT knowledge look like programming. Wikis can be used to work on text and with some restrictions also to share files. Most wikis have the possibility to figure out who participated and follow the development of a page. However, it needs additional effort to understand the collaborative parts.
Wikis need administrative work to setup and maintain, which limits their application. Wikis also face the same problems as filesystems regarding hierarchy depths and can even become more confusing due to cross linking of pages.

As explained, each category of tools faces certain difficulties, that can prevent their usage. Usually, a combination of tools is in place, which can lead to additional complications:

- *Universal notifications:* A central part in collaborative environments are asynchronously working participants. Independent working require update mechanisms to see changes over time. Such changes need to be reported across different tools. Information change is only one particular example for events that cause a system reaction. The reactive web demands for a more general rule-based event detection and trigger notification mechanism.
- *Multiple working sites:* A major problem is the existence of multiple, unrelated working sites, which users have to manage. Think of the administration of a lecture series titled “CS 101” within a campus environment. A Lecturer creates a structured folder in his local file system, sets up an e-learning site where teach-ware is placed, communicates by email with tutors who help in the course preparation, and browses for new material in the web. Without usage of a collaboration tool, there would be a file directory “CS 101” for document preparation, a mailbox “CS 101” to store related emails, a courseware entry “CS 101” to upload teach-ware, and a bookmark folder “CS 101” to store pointers. Having a strict naming system might help in keeping things right but the better solution would be an ”federated umbrella approach” in which a single entity “CS 101” represents the information base and interacts with involved collaboration tools.

The paper is structured as follows: In Section II, we describe the RBs framework for collaborative web work environments, which is intended to overcome the discussed problems. The users view of the PROBINDR prototype implementation is treated in Section III. Section IV reports on implementation aspects, Section V shows three practical applications and Section VI concludes with results and lessons learned.

II. THE REACTIVE BINDER FRAMEWORK

There might never be one single tool that fulfills all collaboration needs, but the fact that users tend to ignore tool support is a challenge. We believe that the entry costs into collaboration tool usage needs to be lowered. Users like a smooth transition from well-known environments and prefer models of type KISS (Keep It Short and Simple). Our approach is a contribution towards a more accepted and used collaboration environment.

We propose the concept of *Reactive Binders* (RBs) as a data organization and collaboration mechanism for team-oriented web work as seen in Figure 1. RBs extends the traditional office approach with social software mechanisms and is able to integrate web agent and ambient computing technology.

Shelves, Binders, and Tabs

The classic office organization is based on physical shelves and binders. People have been using binders for more than 100 years as document storage. Their usage has been so successful that specific names are popular in different countries: in Germany *Leitz-Ordner*, in Switzerland *Bundesordner*, and in US *3-ring binder* are terms known by a majority of people. Each binder has a register and labelled tabs, which represent the storage site where documents are placed. Shelves are placeholders for binders and are either private or shared.

One reason for its popularity even in the computer age is its simplicity. However, there are severe disadvantages:

a) Binders are physically bound to an office. While they can be moved and accessed within a certain distance, this is impossible from remote and and in a mobile environment.
b) The access scheme to binders is very restrictive: The physical key encoding defines whether you may enter offices and access shelves and binders. But key encodings define door access, which is much too coarse-granular in order to represent individual rights.
c) Binders are static: They “live” in a shelf, get updated from time to time, and only dust on cover sheets will indicate that content may be outdated or of archive type.
d) Binders have cross-relationships, which are typically represented in a narrow-sense only: e.g., common naming schemes, coloring, and placement of binders. The strict hierarchical single-place organization scheme is insufficient because documents are often relevant in different contexts.
e) The smallest shareable entity is a single sheet.

The RBs concept addresses these problems (see Figure 1):

a) RBs are virtual entities. If you have access right to a binder it is accessible from the internet.
b) Access rights to RBs are fine-granular. You can individually decide for each shelf, binder, tab and entry who gets administrator, editor, and viewer rights.
c) RBs are dynamic. Like objects in object-oriented programming, RBs represent data elements plus methods, that can be applied. For example, translation and data presentation can be universally defined.
d) RBs are reactive. A rule-based event-action mechanism results in live documents. Changes in one section of a binder may automatically result in changes in the same or other binders.
e) RBs are cooperative. All elements are addressable, which provides the functionality to interact with other tools such as web browsers, web services, wikis, and email systems.

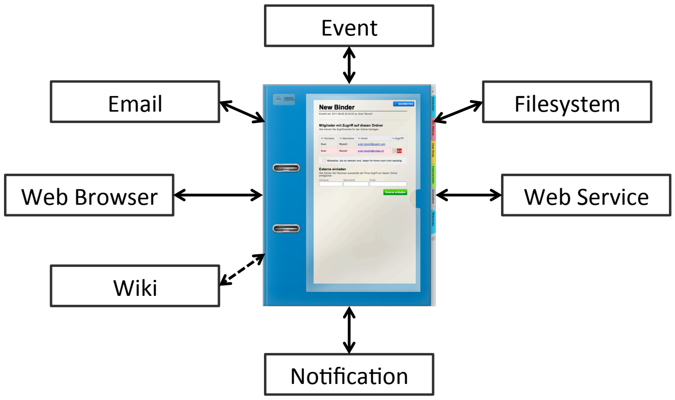

Fig. 1. RBs interfaces to various other systems

Related Work

We accentuate reactiveness strongly in our approach and RBs and its implementation should be viewed as a contribution to the Reactive Web [9]. In addition to collaboration possibilities of other tools, we add workflow functionality and a tighter integration of the real and virtual world to the RBs.

Other binder solutions include Assembla [10], LiveBinders [11], and TeamBinder [12]. Also DataSharing Tools such as DropBox [13] or Springnote [14] have certain relationships. However, these tools have their main focus on data sharing with collaboration aspects such as notifications upon changes, or access control mechanisms.

III. RB AS SEEN BY THE USER

ProBindr is our current prototype for experimenting with the RBs framework. There is a clear defined hierarchy in place for each user, which consists of shelves, binders, tabs, and content items. Each organization is represented as a shelf, which contains a maximum of 33 binders as shown in Figure 2. Whenever a user makes changes to one of the binders, a number on the shelf indicates the number of total changes among all participants.

Fig. 2. Fast overview about modifications in a structured way

A binder is a structuring element that contains a maximum of nine colored tabs. Each binder can be shared individually with members of the organization. Additionally external partners can have access to a particular binder as well. Fine-grained access rights define who can administrate, edit, or only view content items.

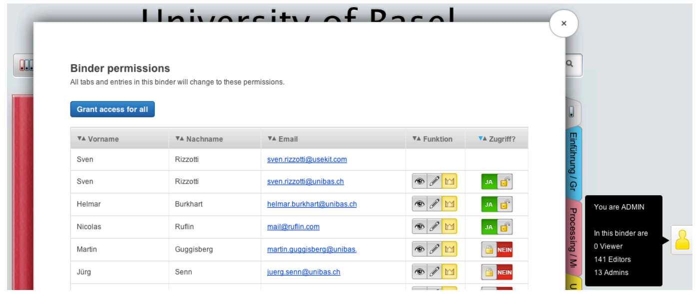

Fig. 3. A simple interface to grant complex permissions

A tab is an endless page that is a compilation of content items. Content items can be files, texts, links, section headings, and custom entries. Each item shows its author, modification date, and if it has already been read. Items can be commented, which replaces referring emails as shown in Figure 4.

Items can be edited or rearranged if a user has the appropriate permissions. The combination of files, texts, links, comments, and structuring elements such as section titles provide a simple and common understood way to collaborate.

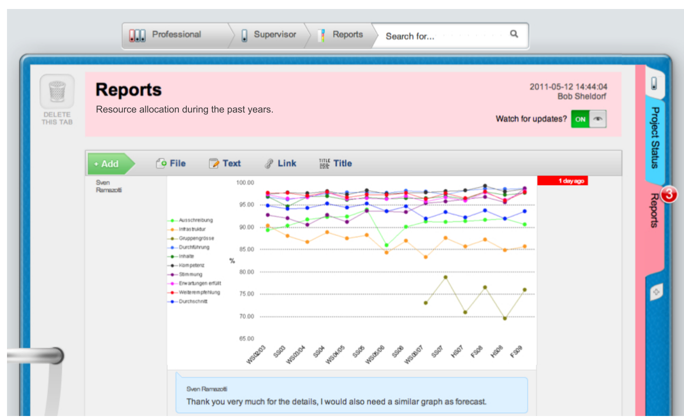

Fig. 4. Comments bound to content items as a way to avoid emails

Permission settings follow an inheritance strategy, known from object oriented development [15]. Settings at a certain level propagate from the shelf down to single entries. On the other hand, simple access is granted in a bottom up strategy. This mechanism allows to give access to a single entry, which makes the tab, the binder, and the shelf visible to the user, but prevents access to other binders. In the other direction, a permission change at the binder level for example inherits the access rights to all tabs and all entries in the binder.

The permission settings use sliders to switch settings on or off at the different hierarchy levels. Figure 3 shows the

 ISBN: 978-1-61208-206-6

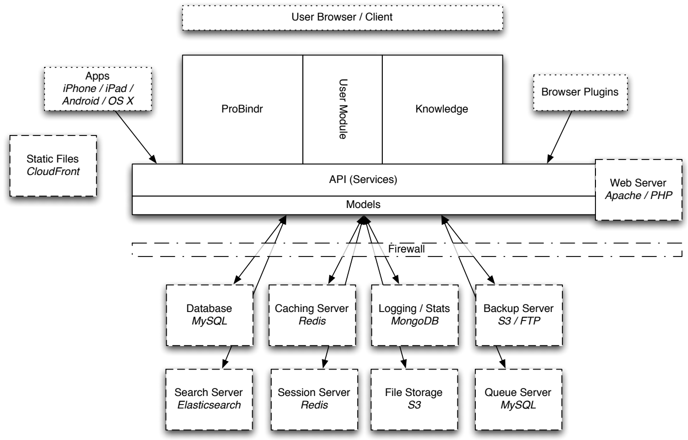

Fig. 5. PROBINDR Architecture

interface, which tries to follow the KISS principle, even if the inherited settings are complex.

IV. IMPLEMENTATION ASPECTS

The PROBINDR platform is completely cloud-based. Figure 5 shows data storage and processing, that are physically separated and distributed to several systems. This procedure offers strong caching possibilities and fast access. From the beginning, the platform has been designed to scale as access numbers increase. A key-value data storage such as used by Facebook allows efficient access. Individual access rights and the hierarchy implemented (user, organization, binder, tab, and content item) result in complex search queries. For instance, these queries are used to trace unread user items.

Beside data access, there are other tasks that are automatically being processed in the background. Search indexing and previews and image thumbnails are for example processed within a task queue. Notification actions are also processed and allow asynchronous completion. All actions happen through authorized APIs, which not only serve as basis for a strict separation of controlling and presentation, but also build the foundation for 3rd party applications. Mobile applications for example use exactly the same API as the platform internally.

V. APPLICATIONS

In three different application area, we currently have field tests for the PROBINDR prototype. All three have in common that different end users with specific interests need to collaborate.

A. Project Management in Architecture

Architects take the responsibility to coordinate among different parties such as craftsmen, administrative offices, building owners, and others. Communication and data exchange is a difficult task in this situation of scattered contact persons. Additionally, all people involved can change during a single project. Large data files, frequent version changes and the organization of all documents require a lot of effort. Thus, it is challenging to keep everyone updated.

The group of architects so far used email for coordination and communication purposes. But the list of their complaints was long:

- Document size often exceeds mailbox limits and cannot be sent.
- Several documents concerning one topic must be sent in multiple emails. Correlations get lost.
- Sending of modified documents ends up in a load of different versions and is not manageable.
- Additional phone calls or emails are necessary to clarify questions.
- No guarantee that information updates reach all recipients.

PROBINDR is currently tested whether it can better serve the architects needs. In addition to the shared information space that a binder provides, its reactiveness feature is used to set up required business rules. For instance, meetings become much more efficient if it becomes transparent who of the attendants has not reviewed recent documents. Logging of user activities in such an environment is not seen as privacy attack but more as a constructive element to reach final project goals.

So far, architects send plans as printouts, which is very expensive. While at later project stages of a construction process, printed plans are mandatory, in early planning phases electronic versions are sufficient. We currently explore the usage of QR codes for the co-existence of physical and virtual items. Together with a mobile application this would allow to quickly check if a printed document is still up-to-date and give access to online comments.

B. Informatics Support for Athlete Coaching

Together with a Center for Sports Medicine we have developed *FMS-Book*, which is an application for the administration of fitness tests of athletes. The sports medical background is an assessment procedure for the evaluation of an athlete's fundamental movement state [16]. Seven basic exercises that require a balance of stability and mobility have to be practiced and a therapist scores the exercises performed. So far, the FMS tests have been documented using paper and pencil, which is inefficient in many ways.

It is a requirement, which PROBINDR fulfills that the user groups involved (athlete, trainer, therapist, and medical consultant) have different access rights to the data collected. Reactiveness of binders is crucial for the final goal to improve athletes movement and stability state: As soon as marks are given in one tab of the athlete's binder, another tab will be automatically filled with exercises proposed by the medical center. These general recommendations prepared beforehand can individually be extended by therapist and trainer. As soon as any changes in the binder are registered, the athlete gets notified. The athlete's binder is a collection of relevant data regarding coaching and training support.

 ISBN: 978-1-61208-206-6

C. Collaborative Course Management

In the fall semester 2011, course preparations, teachware distribution, and student interaction for a computer science service course ”Software Tools in Informatics” used PROBINDR as organizational core component. In this course, a heterogenous community of chemistry, nano-science, computer science, and computational science students plus interested students from other fields attended. With over 100 students, efficiency in course management is important and because of their different background in using computers, a KISS-type support system was favored against tools already in place.

In phase 1, lecturer, chief assistant, secretary, and tutors collaborated by using binder ”CS102 Admin” for the creation of exercises, reservation of lab space, assignment of tutors to student groups, etc. In phase 2, a second binder ”CS102 Content” was in focus.

PROBINDR received positive feedback from the different end user groups. We have analyzed a survey from 123 students out of 12 different disciplines. 77% of the participants noticed the very intuitive usage and effective overview as one of the key strengths of PROBINDR. The majority of users prefers a further usage of PROBINDR to other tools.

VI. CONCLUSIONS

Currently PROBINDR is used by more than 300 companies with over 1000 binders.

With PROBINDR we could prove that it is possible to introduce a new generic collaboration environment that is being accepted. We could also show that the adoption of well-known structures and metaphors substantially contribute to the simplicity and acceptance and therefore also to the actual usage of the collaboration tool.

The RBs concept will now be tested for more complex scenarios: e.g., web spiders may signal within a binder whenever a referenced web site received an update.

REFERENCES

[1] J. Gube. (2008, 10) 15 free tools for web-based collaboration. [Retrieved: May, 2012]. Available: http://sixrevisions.com/tools/15-free-tools-for-web-based-collaboration

[2] A. Dreyssig, *Von Mindmapping bis Projektmangement Die besten Collaboration-Tools*, Lyonel-Feininger-Straße 26, 80807 München, Januar 2012.

[3] T. Yamasaki. (2010, April) Email statistics report, 2010-2014. [Retrieved: April, 2012]. Available: http://www.radicati.com/?p=5290

[4] MessageLabs Intelligence, *2010 Annual Security Report*. NY 10018 New York: Symantec.cloud, MessageLabs, 2010.

[5] Redicati Group, *Addressing Information Overload In Corporate EMail: The Economics of User Attention*. 1900 Embarcadero Road, Suite 206 Palo Alto, CA 94303: Redicati Group, Inc., 2007.

[6] J. Vogel, W. Geyer, L.-T. Cheng, and M. Muller, “Consistency control for synchronous and asynchronous collaboration based on shared objects and activities,” in *COMPUTER SUPPORTED COOPERATIVE WORK (CSCW)*, vol. 13, no. 5–6. Springer, 1992.

[7] S. Whittaker and C. Sidner, “Email overload: exploring personal information management of email,” in *CHI 96 CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS*. ACM Press, 1996, pp. 276–283.

[8] G. A. Miller, “The magical number seven, plus or minus two: some limits on our capacity for processing information.” *Psychological Review*, pp. 81–97, 1956.

[9] P.-L. Pătrânjan, “The language xchange: A declarative approach to reactivity on the web.” Ph.D. dissertation, University of Munich, Germany, July 2005.

[10] Assembla, Inc. Assembla workspaces to accelerate software teams. [Retrieved: May, 2012]. Available: http://www.assembla.com

[11] LiveBinders, Inc. Livebinders - your 3-ring binder for the web. [Retrieved: May, 2012]. Available: http://www.livebinders.com/

[12] QA Software Pty Ltd. Teambinder - advanced project collaboration. [Retrieved: May, 2012]. Available: http://www.teambinder.com/

[13] “Free online storage,” 760 Market Street, San Francisco. [Retrieved: May, 2012]. Available: https://www.dropbox.com/home

[14] openmaru studio. Springnote - a free online notebook based on wiki. Last checked: 14.4.2012. [Online]. Available: http://www.springnote.com/

[15] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed., ser. Addison-Wesley Professional Computing Series. Addison Wesley, 1994.

[16] G. Cook, L. Burton, and B. Hoogenboom, *Pre-Participation Screening: The Use of Fundamental Movements as an Assessment of Function - Part 1*. North American Journal of Sports Physical Therapy, May 2006, vol. 1(2), pp. 62–72.