University of Alberta

Collaborative Review and Analysis of Science Literature

by

Samaneh Bayat

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

To my beloved husband

and

to my dear family

Abstract

Scientific literature is constantly being reviewed and critically analyzed by researchers, individually or collaboratively. Collaborative reviewing enables receiving multiple perspectives on the content and extending ideas over each other. Existing collaborative-review tools do not effectively support collection and retrieving of reviews. We identify requirements for a system that enables the community of researchers reuse the reviews, receive comprehensive reports of existing reviews, perform incremental reviewing, and cross reference reviews through content or reviewers. We introduce CoRAL (Collaborative Review and Analysis of Literature) to support comprehensive collaborative reviewing and encourage publicly accessible reviews. CoRAL provides an API for collecting and analyzing a universal collection of reviews. Two other components of CoRAL, the Reporter and the Viewer, enable users to enter and collect context-aware reviews and to generate multi-dimensional reports about them. CoRAL's architecture is open for extension to various usage scenarios and for integration with other reviewing tools.

Acknowledgements

First of all, I hereby express my sincerest gratitude to my M.Sc. supervisor, Professor Eleni Stroulia, for her kind and great supervision, heartful encouragements, and constructive advice and feedbacks.

It is a pleasure to thank Fabio Corubolo, University of Trento, Italy, for his generous support towards my attempts in extending his work, the Fab4Browser, as part of this thesis.

In addition, I strongly appreciate and value the efforts of the undergraduate students who were involved in development of the CoRAL Reporter, a main component of the project presented in this thesis. Here comes their names in alphabetical order: Lei Dai (Simon Fraser University), Ilia Pak (University of British Columbia), Chong Su (University of Waterloo), Jia Sun (Simon Fraser University), Keen Sung (University of Alberta), and Adam Williams (University of British Columbia).

Last but not least, my special and deepest thanks goes to my dear family who have always been a great support and role model in every step of my life, especially my academic endeavors. Particularly, I would like to thank my beloved husband, Ramin, for all his immense love, support, and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Literature review or critical reading of scientific papers is an important research activity that can be seen in various forms: conference review process, personal literature review, research group meetings, survey writing, and critical observation of paper and poster presentations. Every scientific publication, mainly the papers and presentations, undergoes numerous forms of reviews around the world that are usually inaccessible to the general audience of the work. In this process, valuable reviews are produced, ideas are formed, and mental judgments are made, but these are generally lost by a change in time or location. In addition, the type of media used to read the content and make the reviews imposes another limitation to retrieving ideas at another time by the same or a different person.

Each comment, rating, tag, or reviewer interactions can be perceived as a small piece of review that carries information about how the content is analyzed or critiqued. We refer to these small pieces of reviews as *micro-reviews* in this thesis. Collecting, grouping, and analyzing these micro-reviews can produce a high value and comprehensive review of scientific literature. Losing them, on the other hand, means ignoring a valuable body of knowledge being produced every moment around the core research resource: papers. In addition to having reviewing value, the short comments that readers usually make on scientific papers may potentially carry valuable ideas that can be further improved or extended in a collaborative environment.

To further clarify the above problem statement, here we describe a common scenario in the literature reviewing process. Alice wants to read and annotate a

PDF copy of a paper titled ‘A solution for problem X’. She opens the paper in the Adobe Acrobat Reader, so that she can put comments on it to save her ideas. Sometime later, she wishes to take a look back at that particular paper to review or use her comments. So, she needs access to the exact same copy of the paper in order for the PDF to contain her comments. As a solution, she may use web-based library management systems such as Mendeley (refer to Section 2.4.4) to have access to her comments without carrying a copy of the paper. But, if she finds the same paper in the HTML format her comments cannot be retrieved on this new document format using such library systems. On the other side, Bob is another researcher interested in the same paper that Alice has been reviewing. Bob and Alice do not know each other, so they are unaware of the common interest and they cannot share their reviews using email or the group mechanism provided by library systems. Using a tool like Fab4Browser (refer to Section 2.5), Alice can retrieve her comments on the same content disregarding the document format, and Bob can see Alice’s comments on his copy of the document without prior sharing of the document. But, they will both need to use Fab4Browser and they both miss benefits of using library management systems. In addition, none of these tools help Bob and Alice to have an in-context discussion inside the document. On the other hand, Alice and Bob could benefit from a universal and comprehensive report about how this paper has been reviewed through time, who the other reviewers are, how it is rated by others, and what the related papers are, which are not provided by current tools. As another case, if they are reviewing a content that is not yet published, but it is being presented in a conference as a talk or a poster, it becomes very difficult to share and store ideas for future use or in a collaborative way. This scenario suggests that, current review-collection mechanisms are tied to either time, location, person, or media.

This research envisions to provide a collaborative environment for reviewing of scientific literature, targeting the problems faced in the above scenario. The main goal is to maintain a global and comprehensive collaborative-reviewing perspective around scientific papers and the reviewers. The main considerations in providing this environment includes removing the ties between reviewing activities and time, location, media, interface, and persons. We believe that this leads to producing and consuming novel ideas more rapidly and in a worldwide extent. Those novel ideas may be main topic or a part of a paper or can form by discussion on an initiative arisen from a piece of a paper.

In order to provide such an environment, we have implemented a tool named CoRAL (Collaborative Review and Analysis of Literature). CoRAL's distinguishing characteristics include universal collecting of collaborative reviews, document-centric review collecting with consideration to the fact that various copies of the same document may exist, attitude and context-awareness in commenting and discussions, open reviews and open collaboration opportunities, multi-criteria rating, and analysis of collected reviews from a contextual viewpoint.

Looking back at the Alice and Bob scenario, CoRAL supports public sharing of reviews without requiring the reviewers to know each other and form online groups. CoRAL provides an API for collecting reviews which makes it accessible through the potential library management systems or tools like Fab4Browser. Ideally, Alice and Bob can review their documents in their favorite clients while still being able to share comments. In CoRAL, inspired from Fab4Browser, the review collection can be chosen to be independent from the document format. In addition, CoRAL supports in-context discussions. CoRAL is also favorable for comprehensive report generation, where the reviews are summarized and analyzed from various perspectives. On the other hand, CoRAL is open for extension to the case that the document is being reviewed in a conference event.

This research considers two groups of users with specialized benefits for each group. The first group is envisioned to be the students or researchers new to a field. Those who look for new ideas or need professional guide through understanding a subject can benefit from having a collection of reviews, discussions, and highlights in the context of publications. Those who search for papers in a field, not knowing what they need to find or how to effectively search, can benefit from prior assigned tags and ratings by those who had read papers. Those who want to start with major publications in their field can benefit from finding papers based on the attention they gain in the community of readers and reviewers.

The second group is considered to be the professionals and active researchers in a field. This group can benefit from easily sharing and discussing their ideas where their comments can be found and critiqued by other active researchers. Authors of publications can find out how the community of researchers react to their publications. Professional networks can be extended around common interests, critical analytic abilities, and beyond location-specific interactions. Those who think similarly or perform worthwhile debates together, can match using the proposed tool.

In general, the community of researchers can benefit from the proposed system by having access to an open and universal collection of reviews.

1.1 CoRAL Features

Our study of the existing tools in the area of collaborative reading and reviewing (Section 2.4) shows that every tool collects its own collection of reviews rather than considering a universal collection. In addition, current tools miss some important characteristics that we find crucial for effective review collecting, if considered individually. In this research, we aim to encourage collecting public reviews on scientific publications independent of the tool used for making reviews. In this regard, we also identify some requirements for effective review creation and collecting.

Most of the collaborative-reviewing tools allow users to share reviews in a group basis, if not by explicit document sharing via email. This method bounds the reviews to the group in which they are shared, rather than the content of the reviewed publication. Groups remain unaware of existence of related reviews in other groups. Although this kind of review collection is beneficial for reviews that are meant to remain private, for publicly available reviews we should consider collecting reviews around documents rather than the groups. Document-centric collection of reviews is also in favor of giving access to publicly available reviews of people who are not in each other's prior collaboration zone, hence widening the concept of collaboration and facilitating reuse and extension of far or common ideas.

Reviewing scientific publications can be done in various forms: commenting about the whole publication, commenting on a particular idea in the specific context where it is discussed in the paper, highlighting important parts, discussion, tagging to identify categories and related publications, and rating. Talking about collecting reviews, we actually talk about these important forms. Most of the current collaborative-reviewing tools do not consider all these forms, specifically they are ignorant to in-context reviews including in-context comments and highlighting. The reason for this ignorance is generally separation of in-context reviewing tools from collaborative-reviewing tools, where the majority of collaborative tools concentrate on managing online libraries in the form of a collection of bibliographies without content. One major problem with having collaborative in-context reviewing tools is providing access to copyright content. The legitimate tools that combine collaboration with in-context reviewing, such as Mendeley (discussed in Section 2.4.4), require group-based collaboration to provide private sharing. In this research, similar to Fab4Browser (discussed in Section 2.5), we consider the following requirements for the proposed system in order to have legitimate in-context non-group-based review collecting:

- The tool should separate the reviewing interface from the review-collection mechanism.
- The system should generate reviews attached to the document content, but collect them unattached from the document body. This method eliminates the need to share the document in order to share reviews while enabling in-context reviewing.
- It should add a piece of content, or metadata about the content, to the reviews at the time of collecting or sharing. This adds value and information to the unattached reviews.
- It should allow the reviews to be able to reattach to the document body in a reading interface.

Beside the individual commenting and sharing, discussion is an important review activity which is inherently collaborative. Most of the existing tools support discussion outside the document content and in the form of forums or group discussions. Similar to the in-context commenting, the ability to have an in-context discussion is valuable in being directly connected to the publication content. When a small piece of content attracts some form of attention, it is easier to ask a question, raise an issue, praise an idea, and pursue a collaborative discussion with in-context commenting and discussing. In CoRAL, we encourage collaborative-reviewing tools to support this form of reviewing. We believe that the value of the comments increases with having some form of pointer to the content that it is specifically talking about.

Among the existing tools that consider in-context review sharing, most of them require a certain copy of the document to be shared as a review holder. This approach suffers from the same problem with the group-based review collecting: reviews are attached to something other than the "content" of the publication itself, in this case, the specific "file" in which the publication is represented. Similar to Fab4Browser, CoRAL collects metadata about the reviewed publication and the reviews can be retrieved from any copies of a scientific publication that are conceptually the same. This is also similar to the mechanism used in tools that only support global reviews, such as CiteULike (discussed in Section 2.4.2), collected over the bibliographic information rather than a file. In CoRAL, we combine the Fab4Browser's method and the bibliographic identification method to reach to a comprehensive copy identification mechanism.

Like most of the collaborative-reviewing tools, CoRAL considers social tagging as a form of review. Frequently assigned tags help identify categories or common aspects among reviewed publications. Effective use of tags leads to connecting related literature which is itself a form of reviewing literature from a broader perspective.

Another easy and routine form of reviewing is rating the literature. The existing collaborative-reviewing tools, designed for reviewing published literature as a research process, usually support rating by one total criteria. On the other hand, the same literature undergoes multi-criteria rating at the time of being reviewed before being selected for publishing. The conferences care for a comprehensive rating system in which reviewers rate scientific publications according to a number of criteria such as: novelty and clarity. CoRAL encourages the same rating system to continue to be used among readers of the literature. This way, the rating concerns are separated and a more precise rating is reached. In addition, the separation of rating concerns is in favor of the separation of searching concerns. Normally, ratings are useful for ranking search results. Multi-criteria rating contributes to specialized ranking by customizing the rating criteria that is more favorable for the goal of search.

An important consideration in the proposed system is the ability to analyze, summarize, or visualize the collected reviews. As pointed out earlier, the reviews are collected unattached from the document itself. Where we have a huge collection of floating reviews ready to be analyzed, having a piece of the related reviewed content and knowing the reviewer's attitude adds to understandability and value of the summary of reviews. In this research, we encourage context and attitude awareness in the collected reviews. Having a universal collection of review contents, various review forms, context-awareness, attitude-awareness, reviewer history, and metadata about the reviewed publication enables us to provide worthy reports and summaries about reviews on scientific publications.

CoRAL consists of three main components: the API, the Viewer, and the Reporter. The API is designed to separate the review collection from review making and review viewing interfaces. The existence of the API is the main step towards providing a universal collection of reviews. The Viewer is a review making and viewing application. According to the identified requirements for the proposed system, explained above, Fab4Browser is a close match to what we expect from the CoRAL review making client. The Viewer is an extension to Fab4Browser, which adds the missing requirements: context and attitude awareness, in-context discussion, bibliographic publication-identification, and tag and rate features. The Reporter is responsible for analyzing and visualizing the collected reviews to provide a better insight about reviews and reviewing activities.

Although the current CoRAL system is composed of the three mentioned components, the architecture is designed to be open to further expansions. As mentioned earlier, CoRAL aims to become a review-collection system independent of the used reviewing interface. So, it is designed with an API and easy connection mechanism (discussed in Chapter 3) to encourage related tools to contribute to the review collection. In addition, it is extensible to related usage scenarios. The basic supported scenario is reviewing published literature using a desktop application, as part of a routine research process. The focus of currently implemented CoRAL components is on this basic scenario. But, the system is extensible to more scenarios such as in the conference review process and reviewing presentations in a workshop or conference event.

1.2 Thesis Contributions

According to the discussion that we had in the previous section about the CoRAL features, the contributions of this thesis can be enumerated as follows.

- It identifies the requirements of an effective open and collaborative-review system, based on ideas from library-sciences research and from a comparative analysis of several current systems and their features and shortcomings.

- It develops a prototype tool for supporting collaboration among scientists and researchers with similar interests, by supporting discussions on publications, open document-centric review collection, and social-networking activities such as profile management, and community-member "following".

- The prototype tool implements the identified requirements with an open API, a review making tool (the Viewer), and an analyzer tool (the Reporter).

  - Providing an open API for the collection of reviews enables collection and analysis of reviews, independently of the particular interface and application used to produce the reviews. Through this API, the tool is designed to support future extensions in order to handle new usage scenarios.

  - The Viewer component supports in-depth reviewing with these features: in-context commenting and discussion for in-depth collaboration and understanding, identification of content and attitude related to comments to increase the context-awareness, free-vocabulary tagging, multi-criteria rating, and considering a document and its various copies in retrieving related reviews.

  - The Reporter enables users to obtain examples of insightful review reports and to visualize them from various perspectives.

- Through its design, we anticipate that, in the future, the tool will be used to conduct systematic studies on reviewing behaviors and practices by providing a collection of reviews which carries information needed for relevant studies.

1.3 Thesis Outline

This thesis is organized as follows. Chapter 2 studies related research and identifies the requirements for the proposed system to improve over the existing approaches. In Chapter 3, we discuss CoRAL's design, justify the decisions, and describe an overview of the system's architecture. In Section 3.1 to Section 3.3, the system components (the API, the Reporter, and the Viewer) are discussed separately.

Chapter 4 talks about the system usage and features as they are perceived by the user. In that chapter, snapshots of the interface of the actual system can be found.

After we talk about how the system is built and what it brings to the user in Chapters 3 and 4, we discuss the potential benefits of the proposed system for both collaborative review and review analysis in Chapter 5.

All over this writing, except Chapter 6, we concentrate on a basic scenario of usage that is envisioned for this system. However, in Chapter 6 we discuss how this system is open for extension to other related scenarios. In addition, we talk about integration of the proposed system with one of popular related tools that can be a great match to our goals: Mendeley.

Finally, Chatper 7 concludes the thesis and enumerates possible interesting future work.

Chapter 2

Background and Related Work

This research searches for potentials of improvement to existing approaches regarding collaborative literature reviewing. It aims to tackle these potentials with introducing a new system called CoRAL. Higher engagement of Web 2.0 principles into literature reviewing tools is an important aspect of such potentials which is discussed in Section 2.1. As a side effect of increasing the social aspects in our literature reviewing service, we should also consider the known trust and accountability issues that are studied in the context of existing social information systems (Section 2.2). As another rarely explored potential in literature reviewing tools, we are interested in having the proposed system in coordination with existing reading patterns which we study in Section 2.3. Other possible improvements are identified by a discussion on existing tools and extracting requirements for a new approach, found in Section 2.4. At the end of this chapter, we introduce the platform that is used for implementing the document reading and review making client. The chosen platform, Fab4Browser, highly conforms to the design requirements that we identify in Section 2.4.

2.1 Social, Open, and Crowdsourced

In the era of Web 2.0, more and more applications tend to take full advantage of the web as a network by employing the Web 2.0 principles. These principles, that emerged gradually from new social and open applications, were clarified by Tim O'Reilly [1] and then further explained in the form of 'big ideas' by Paul Anderson [2]. A number of existing collaborative scientific review tools, such as Mendeley (refer to Section 2.4.4), may fall into Web 2.0 concept by considering the tagging and social bookmarking tools as a category of Web 2.0 key services as mentioned in [2]. The social behavior of these systems is generally concentrated around tagging and bookmarking. The potential of Web 2.0 is not fully investigated by those systems in regard to the principles of openness, crowdsourcing, and user-generated content when it comes to textual and contextual reviews.

As we will discuss in Section 2.4, existing collaborative-review tools either do not support contextual reviews or they limit collection of reviews to groups. Closed groups, public or private, restrict social behavior of the system and limit the power of crowdsourcing and user content generation in high scale. For example, one may compare the popularity, benefits, and power of del.icio.us [1] (a popular social bookmarking service) in contrast with its community-based counterpart coRank.com [2]. In this thesis, we encourage openness of reviews and design a system to take full advantage of social creation of a rich collection of reviews.

Crowdsourcing, as defined in [3], is the concept of outsourcing a function that would traditionally be accomplished by a set of employees to a bigger network of people through an open call. In the context of scientific reviews, crowdsourcing suggests that instead of having a system like ACM Reviews (refer to Section 2.4.3) to publish expert reviews, let the readers of scientific publications to create comprehensive analytical reviews. This way, the consumers become the producers and the crowd benefits from the crowd. This is also an adaption of the 'wisdom of crowds' [1] [4] in replacing expert reviews by crowd reviews.

2.2 Trust and Accountability

Trust and accountability are important factors in continuous and effective usage of open collaborative systems. For example, Wikipedia is a huge collection of content published and edited by users with no formal peer review of the published material. P. Denning *et al.* [5] enumerate the risks and trust issues related to use of Wikipedia as an encyclopedia. The same openness and social editing can be seen in the tools such as CiteULike (refer to Section 2.4.2) which allow sharing reviews about pub-

[1] http://www.delicious.com/

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

lications. Among the risks that Denning *et al.* mention, 'motives' and 'uncertain expertise' of the editors can also be the risk factors in a tool that collects reviews.

Many studies are done to define reputation and trust metrics for editors and words in Wikipedia [6] [7]. In order to mitigate the risk factors that basically are dependent to the editor's behavior, studies suggest visualized trust values to be fed back to the reader of the article. This can be in the form of color-coding the words [7] or a separate page reporting about reputation of editors like in Wikidashboard [8].

In spite of existence of many studies on trust issues regarding collaborative information systems, there is rare attention to the similar issues regarding scientific review systems. For example, CiteULike and ResearchGate (refer to Section 2.4.1) allow users to submit reviews regarding scientific publications without going through formal peer review. However, these systems are not generally known as review-collecting tools due to limited user participation in actual reviewing. The reason can be related to the fact that they only allow creation of global notes and comprehensive reviews which is a barrier in easy creation of content in contrast with in-context small note-taking mechanism. In this thesis, the proposed system aims to take full advantage of social review creation with minimizing the barriers by letting users contribute to the content when they are reading the literature on their desktop application. In such a system, that we hope to attract more user participation, we should beware of the trust and accountability issues. For this reason, the proposed system contains a reporting mechanism in which user participation and attitudes are visualized. Every reviewed publication accompanies a comprehensive report containing information about its reviewers such as: how much active they are, what is their overall attitude with respect to other publications, and what are the other publications that they contribute to. According to relevant studies on Wikipedia [6], the more active users usually have the higher reputation in creating trusted content. So, we consider activity metrics as a promising reputation metric.

2.3 Change in Reading Patterns

C. Tenopir *et al.* [9] show that the increase in availability of digital articles has led to increase in the number of papers read and parallel decrease in the time spent reading each paper. Similarly, Z. Liu [10] shows that literature reading patterns have moved from in-depth reading to more keyword spotting and selective reading. This highlights a need towards applications that help researchers find what they need fast, while at the same time, including in the retrieved information "everything" related so that the reader can have a deeper understanding of the studied publication(s). L. Zhang *et al.* [11] develop a taxonomy of functional units, the smallest information unit, to guide readers to the highly relevant information in an article. Their study is done in one genre within psychology articles domain and employs core components of articles (Introduction, Methods, Results, and Discussion). They also consider associations between functional units to connect relevant pieces of information in the article. This is an example of analyzing a document and its components to help readers spot the information they most probably need. As another approach, crowdsourcing can help identifying interesting or controversy pieces of content within articles. Selective reading can be improved and guided through social identification of interesting pieces of content and collection of discussions around them, visualization of the crowd's attitudes, knowing about reviewing habits of others who read the piece of content, and a summary of relevant comments. These are all accomplished by the proposed system, CoRAL.

2.4 Related Tools

There are various tools that each tackles a dimension of collaborative literature reading and reviewing. Here, we briefly explain about goals and highlight some features of such tools, pointing to differences with the proposed system in reaching the goal that was explained as the basis of this research: supporting and analyzing universal collaborative reviewing.

Looking at research and development around collaborative literature reading and reviewing, several branches or themes can be identified such as:

- Academic social networks
- Peer to peer review and sharing
- Academic social bookmarking tools

- Reading libraries
- Review sharing and discussion boards
- Reference management tools

Current tools and services that target collaborative reading and reviewing, may embody one or more of the above themes. In the subsequent lines, we briefly introduce some of the related tools and the themes that they follow. Then, we choose a number of tools that have more common aspects with this research for deeper discussion and comparison.

Qiqqa [12] is a desktop application used as a library and reference management tool. Users can manage their reading list, read and annotate their documents, and retrieve citation information. Although it is a desktop application, the user's library can be synchronized with their account so that they can access to the library from any other machine. Qiqqa is very useful for managing large reading lists and generating summaries based on the annotations. However, it still does not provide means to share ideas and annotations.

JournalFire [13] is an online reading library, a discussion board, and also a light academic social network. Users of JournalFire can create reading lists in which they share comments about the articles. They can discuss articles on a discussion board, but outside of the publication's content. Users create their network by following each other's activities. The JournalFire does not indicate whether they are using the data gathered from users in any analysis. The discussion groups (called Journal Club) are open to all users, unless a premium service is purchased for private discussions.

In the category of peer to peer review and sharing tools, one can mention *Adobe Acrobat Reader*, *Microsoft Word*, and *Quilt* [14]. These are used for collaborative document producing, useful for reviewing drafts of documents and share annotations between peers. Other instances are *Crocodoc* [15] and *ReviewBasics* [16]. Crocodoc is used to view PDFs, images, and Office documents in the Internet browsers; add comment anywhere in the document and highlight; share documents and comments with others for real-time collaboration; or embedding the document in a web page for collaborative reviewing. ReviewBasics is for collaboratively annotating documents, images, and videos in the context of the document. ReviewBasics does not support discussions but Crocodoc does. Using this kind of tools, the documents can only be shared and reviewed among people who are invited to.

Academia.edu [17] and *LinkedIn* [18] are a number of interesting tools that have moved towards building a convenient environment for researchers to build their professional profile and network. Academia.edu is specialized for social networking in the academic community. It supports professional profiles, blogging, introducing people with similar interests and similar activities, and following people and papers. Although LinkedIn supports professional networking targeting both researchers and industry, but its main focus is on companies and careers. This category of tools have some common points with our research in the social aspects. In this research, we also consider professional profiling and following colleagues' work. However, these aspects are not the focus of this research and CoRAL provides these only in a limited functionality as basic needs of a collaborative system.

Fab4Browser [19] is a browser for reading and annotating documents in various formats. Using Fab4Browser, one can choose to publish her annotations to a public repository so that it can be retrieved by other users reading the same or similar content. The support for multiple types of annotations and document formats, easy annotation sharing, and the mechanism used to find annotations on the same content regardless of the physical location of the documents makes this tool a perfect base for our research. We have specified Section 2.5 to discuss this tool in more details.

CiteULike [20] and *Mendeley* [21] are online social bookmarking tools for the academic content and community. They also support review sharing and discussions. These tools are a couple of examples of numerous reference management tools that are listed and compared in Wikipedia [3]. In the following subsections we describe these tools and a couple of other services that have more common aspects with our research than the previously mentioned ones.

2.4.1 ResearchGATE

ResearchGATE [22] is an academic social network and professional profiling tool. Features include: following other users; posting comment, link, image, or status; adding publications of own; creating and joining groups; advertising and finding

[3] http://en.wikipedia.org/wiki/Comparison_of_reference_management_software

conferences and jobs; and finding, bookmarking, and commenting on publications. An interesting feature is abstract search based on semantic similarities which is useful for finding related literature when writing a paper. Another distinguishing feature is the ability to request for full text of publications which may be fulfilled by the authors. This requesting feature is also in favor of finding interesting titles that attract attention for being read. ResearchGATE generates statistics on the usage data which includes most read, most requested, most rated, and recently added papers. Discussions can occur in a custom group which will not be associated with any paper. ResearchGATE does not support social tagging.

ResearchGATE is similar to the JournalFire in the sense that reviewers can share their comments beside a publication and outside of the content. However, in JournalFire and tools with a group-based strategy the comments are attached to the group-document pair. So, comments are only available within the group that they were made, even for public access groups. This leads to having a different set of comments in each group on the same document. On the other hand, tools with document-centric view such as ResearchGATE collect all comments around the document. For a reviewing tool that aims to collect all reviews and provide a central idea-sharing service, the document-centric approach seems more favorable. However, lack of support for in-content annotations reduces the appropriateness of this tool for in-depth reviewing of literature, where it also lacks an intuitive discussion support around a publication.

2.4.2 CiteULike

CiteULike is a popular social bookmarking and reference management tool. It is a library for managing user's reading list, that can also be made available to public. The basic features include: adding articles from various sites and databases and automatically obtaining the citation information; specifying priorities for reading the articles; social tagging; commenting, reviewing, and rating; groups, blogs, and forums; following other users' activities; and a lot more. Some of the outstanding features are: recommending articles to read based on interest similarity; and introducing people who are reading the same papers. CiteULike also provides usage-based metrics by indicating the number of readers of each article and the most frequently and recently added articles.

Although users can attach a private copy of the article to their library, CiteULike does not support annotations and comments inside the article's context. As a consequence, it does not support in-context discussions and sharing of ideas. Speaking of the reviewing power, we believe that being able to point to smaller portions of a paper and share ideas around them is crucial. On the other hand, CiteULike gathers reviews and notes outside of the context. Accessing an article, one has access to links to other user's reviews and notes which is a good support for collaborative review. In addition to textual review, users can rate papers on a scale of 1 to 5 which gives a quick and easy review on the paper's total decency.

2.4.3 ACM Reviews

ACM's *Reviews.com* [23] is an online review service designated to reviewing the computing science literature. It is more perceived as an online journal rather than a tool. Similar to our research, the focus of this service is on review and critical analysis of literature. The big difference is in relying on field experts in Reviews.com in contrast with collaborative readers' perspective in CoRAL. This is extremely valuable to have specialized reviews, comprehensively critiquing the articles. In contrast, our research aims to collect collaborative reviews specifically among the bits and pieces of the article's content which is easily performed as part of the reading process.

2.4.4 Mendeley

Mendeley is a popular award winning service which is a reading library, reference manager, social bookmarking, collaborative reviewing, and professional networking tool. More importantly, Mendeley defines usage-based impact metrics to overcome with some problems of the citation-based reputation metrics. Henning *et al.* [24] talk about the problems with current citation-based impact metrics such as Journal Impact Factor (JIF), h-index, and g-index. Then, they introduce Mendeley as a framework to collect data for usage-based measurements. Usage-based impact metrics include real-time statistics around frequency of reading a paper in the community of researchers; skimming through, reading, or repeated reading of a paper;

demographic information of readers of a publication such as geographic region and affiliation; and etc. [25] [24].

Mendeley has two parts: Mendeley Desktop, and the Mendeley website. Mendeley Desktop is used for reviewing articles and managing user's library which can be synchronized with the Mendeley website to have a portable library. Being a reference manager software, articles added to the library are analyzed for their citation information that is obtained either from the file itself or from online citation indexes such as CrossRef [4] and PubMed [5]. Usage data is collected anonymously from the Mendeley Desktop, just as the user adds and reads the articles.

The Mendeley website is used for finding the aggregated statistics about papers, authors, and disciplines. It is also a place to access user's library online, manage profile, extend professional network, get recommendations, and share reading lists in the form of bookmarks.

A Mendeley user library can be a collection of citations, PDFs, images, Office documents, and generally URLs. However, only PDFs can be opened and annotated within Mendeley, while other type of documents would open externally using their specific software. Mendeley supports tagging, one star rating, attaching external notes to the whole paper, and attaching internal notes and highlights inside the content. Users can share their reading lists to public or in a group, in which only the citation information are shared. However, PDFs and their annotations can also be shared if a group is made private to specific users [26].

2.4.5 Summary and Discussion

In the previous sections, we introduced a few powerful and popular tools where each of them is expertized in one or more themes of collaborative reading and reviewing. In addition to identifying themes and discussing features, we are interested in comparing the aforementioned tools according to the following questions. Table 2.1 compares some of the related tools according to the below considerations. The colored cells in Table 2.1 highlight some of the missing beneficial considerations in development of existing reviewing tools which are also discussed in the following paragraphs. As

[4] http://www.crossref.org/

[5] http://www.ncbi.nlm.nih.gov/pubmed/

mentioned earlier and also discussed in the next paragraph, the document-centric review collection method is used to conceptually relate the existing thoughts and reviews to the content of the publication that they refer to. This suggests that this method should value contextual reviewing and analysis. However, as the colored cells in the table suggest, even the tools with document-centric approach do not effectively use the contextual reviews. More discussion can be found in the rest of this section.

Is the collection of reviews document-centric or people-centric?[6] Speaking of reviewing in the form of comments, JournalFire, Mendeley, and all the peer-to-peer review tools such as Adobe Acrobat Reader are people-centric. In the people-centric approach, whether the group or reviews are public or private, everything remains accessible within or via the group. ResearchGATE, CiteU-Like, Fab4Browser, and the research we present in this thesis are examples of the document-centric approach.

Groups are usually considered as a means to have closed and private discussions, although public groups are also usually supported. Mendeley supports both public and private groups, but the reviews (annotations) are only shared in private groups. JournalFire supports both private and public review sharing. In the private groups, collaboration is only done among those who explicitly know each other and intend to share ideas. Many people prefer closed discussions over publishing their ideas to the unknown public. In addition, for some purposes, specially for draft reviewing, existence of private peer-to-peer reviewing tools is crucial. Fortunately, there are various successful tools in this area.

Here, we consider the cases that ideas can be public and reviewers are open for unexplored collaborations. Access to public reviews is needed for collecting ideas in one place and supporting universal collaboration. In the people-centric approach, a document may be reviewed by many different groups and groups usually don't share reviews. The access point to each set of reviews is the group and the people who know each other or find a public group by chance or recommendation. So, having access to a publication does not guarantee access or even awareness of existence of previous thoughts and reviews. In the document-centric approach, the document itself is the

[6] Summarized in Table 2.1 as 'Document-centric review collection'

unique way to access relevant reviews. In comparison with the people-centric one, the document-centric approach is more appropriate for finding all previous discussions and valuable thoughts on a content; increasing the opportunity of learning and reusing of the reviews and review processes; understanding the evolution of reviews in the passage of time; wider collaboration possibilities with people outside of one's network; and much more.

Do they support creation and sharing of annotations?[7]

There can be three general types of textual review: comprehensive review or notes outside of the content (referred to as *global reviews*), and annotations attached inside the content (referred to as *in-context or contextual reviews*). ACM Reviews.com provides comprehensive reviews, critiquing the paper as a whole in every possible aspect. This kind of review is more suitable for experts, that is why it is used by this service. Writing a comprehensive review is much more difficult than expressing one single point that comes into mind, specifically for the starters. Notes can be written as a short note outside of the content, highlighting a couple of aspects of the paper that a reader finds worthy of mentioning. A collection of such notes, can form a nice summary of the paper from various perspectives. In-context annotations, can point to a specific piece of information in the paper, easily and briefly giving a comment on that part or highlighting an idea. Having in-context annotations makes it easier to have single-purpose discussions that become clearly understandable and shorter in length. Having the context accompanying the review adds to the value and effectiveness of the review. In addition, in-context reviews can briefly introduce new ideas or extract and highlight ideas out of the context.

ResearchGATE and JournalFire only support global notes. CiteULike supports both global notes and comprehensive reviews. Mendeley supports global notes and in-context annotations. Fab4Browser, Corocodoc, ReviewBasics, and many systems that can open a PDF, support in-context reviews. But, only a few of them such as Crocodoc support discussions using the in-context annotations (referred to as *in-context discussion*). In this thesis, we consider in-context annotations as an important way to produce valuable, small, and context-aware reviews. We also have

[7] Summarized in Table 2.1 as 'Global reviews', 'In-context annotating and sharing', and 'In-context discussion'

special attention to supporting in-context discussions as an effective collaboration method.

Among the mentioned tools that support in-context annotations, only Fab4Browser considers global sharing of annotations for open access. Mendeley and others require a private closed group in order to share annotations. In this research, we value open collaboration and global sharing of ideas to reach a universal perspective.

How do they analyze the collected usage data?[8]

CiteULike recommends articles to read next based on the user's reading list and its similarity with other users' reading lists. It also provides light usage statistics such as popular articles in last week and how many times an article has appeared in different reading lists. For every paper, ResearchGATE shows how many times it is read, full text is requested, and liked by users.

Mendeley gives more comprehensive reports about usage data which is necessary to reach its goal: introducing usage-based reputation metrics. The statistics that Mendeley provides are related to the reading activities and people profiles. Examples are: number of readers of articles; most read papers in all or each discipline; most read authors in all or each discipline; most frequently used tags in each discipline; and most read publication outlets.

Although most of the introduced tools support some form of commenting and discussion, none of them takes advantage of this type of usage data in their reports (referred to as *analysis of contextual reviews*). In this thesis, having focused on reviewing activities rather than reading, we envision to take the reviewing activities and content into consideration in the reports that the proposed system generates. So, two different viewpoints can be recognized in respect to collected usage data: a basically numerical viewpoint for Mendeley and others, and a contextual viewpoint for our research. For example, Mendeley looks for evaluating publications and readers mainly through collecting statistics, but CoRAL aims to generate a multi-dimensional report by collecting and sharing of ideas in a globally collaborative environment.

Do they provide open access to the collected data and/or analysis?[9]

[8] Summarized in Table 2.1 as 'Analysis of collected data' and 'Analysis of contextual reviews'

[9] Summarized in Table 2.1 as 'Open access to data/analysis'

CiteULike has made three datasets available to public for non-commercial access and research purposes which are updated daily. These datasets are: anonymous who-posted-what data, article linkout data, and group membership data [10].

Mendeley has an open API (application programming interface) so that new applications can be built by using Mendeley data [11]. There are two groups of methods in this API: public methods, and user specific methods. Public methods are used to get statistics about top papers, top authors, and etc. in addition to searching for articles, authors, and tags. User specific ones are used to let users of other applications access their Mendeley libraries after being authenticated.

The data collected in a collaborative reading or reviewing tool carries interesting information that can be used for research or development purposes. In this research, we respect this need by providing an open API that can be accessed to contribute to or retrieve the collection of reviews.

Do they consider context and attitudes in collecting reviews?[12]

The in-context annotations are the reviews that point or relate to pieces of content that are being reviewed. Since this form of review is created as an attachment to the content, the reviewers usually assume that the content always accompanies their reviews. So, collecting this form of review without the content that it refers to, eliminates a piece of knowledge that may clarify the body of reviews. On the other hand, collecting the reviewed pieces of content helps identifying the interesting parts of the publications that can be employed for selective reading. In addition, having this information facilitates sharing of reviews among different document formats where the physical location of reviews may no longer indicate the correct context that had been reviewed.

Although many of the reviewing tools support in-context annotations, they do not show interest in the pieces of content that are being reviewed. But in CoRAL, we are particularly interested in contextual reviews and have noticed their importance as a popular and intuitive form of reviewing. So, in CoRAL we increase context-awareness in our collection of reviews by identifying the pieces of content that the annotations are most probably referring to. Supporting context-aware reviews is in

[10] http://www.citeulike.org/faq/data.adp

[11] http://dev.mendeley.com/

[12] Summarized in Table 2.1 as 'Context and attitude awareness'

favor of better contextual analysis of literature, selective reading, and easier and more in-depth reviewing experience.

In order to increase the value of the collected reviews even more, CoRAL considers attitudes of the reviewers as another collectible knowledge. Ignoring the attitudes of the reviewers and just relying on the load of reviews may cause invalid conclusions towards interestingness of the reviewed content. A similar problem exists in current citation-based reputation metrics that count the number of citations without looking at the attitude of the authors in citing a publication. Note that a publication may be cited in a negative context as much as it can also be positively cited. On the other hand, knowing that a content is both liked and disliked by its reviewers helps us to identify the controversial publications that is another interesting form of contextual analysis. So, in CoRAL we consider the reviewers' attitudes as well as the number of reviews to come up with analysis about interesting or controversial pieces of content. We can also benefit from individual attitudes to identify each reviewer's global attitude in commenting. Knowing and reporting the global attitudes is a good way to evaluate each individual comment.

To the best of our knowledge, the context and attitude awareness are ignored by existing collaborative reviewing tools.

TABLE 2.1

Comparing a number of related tools according to the identified favorable design decisions

	CiteULike	ResearchGate	Fab4Browser	Mendeley	Adobe Acrobat	ACM Reviews	JournalFire	CoRAL
Global reviews	✓	✓		✓		✓	✓	✓*
In-context annotating and sharing			✓	✓	✓			✓
Analysis of collected data	✓	✓		✓		NA	Not Known	✓
Open-access to data/analysis	✓			✓		NA	NA	✓
Document-centric review collection	✓	✓	✓			NA		✓
In-context discussion	NA	NA			✓	NA	NA	✓
Context and attitude awareness								✓
Analysis of contextual reviews								✓

* CoRAL API supports global reviews, but neither the Reporter nor the Viewer distinguish a global note from a note without attached context.

2.5 Fab4Browser: Document Browser and Annotation Platform

Fab4Browser, proposed and implemented by F. Corubolo *et al.* [19], is a distributed annotation system designed for collaborative research. Fab4Browser, inspired by the idea of lexical signatures in creation of robust hyperlinks [27], creates location and format independent references to documents by employing the MD5 encoding algorithm [28]. It supports various types of annotations that are all available across various file formats [19]. The annotation types supported by Fab4Browser include: note inside the document content, anchored note pointing to document content, highlight, comment on highlight, citation to another publication, and etc.

Fab4Browser is built upon the Multivalent Browser [13] which is a digital document browser capable of viewing and annotating various document formats such as HTML, PDF, TeX DVI, and etc. Multivalent Browser is a realization of the Multivalent Document Model [29] and Multivalent Annotations [30]. Fab4Browser has added to the capabilities of the Multivalent browser. Examples are: enhancing the UI, improving the annotation save and retrieve mechanism, adding new media types and adapters, and more importantly, introducing the concept of location independence to the annotations.

Fab4Browser is open source with code license GNU GPL v3. The project home, containing the source code and releases, resides in the Google Code project hosting service [14].

As we discussed and compared related tools in Section 2.4, we also pointed to the properties of the Fab4Browser that are favorable to our research goals. In this thesis, the Fab4Browser is chosen as the platform for implementation of one of the components of CoRAL (CoRAL Viewer - Section 3.2).

Note that in this thesis when we refer to any internal part or functionality of the Fab4Browser, that part or functionality may come from the underlying Multivalent browser or from the Fab4Browser itself. However, for consistency and simplicity we do not distinguish them since we are using the Fab4Browser as the whole.

[13] http://multivalent.sourceforge.net/

[14] http://code.google.com/p/fab4browser/

Chapter 3

The CoRAL Software Architecture

According to the CoRAL's objective of being a center for collecting and analyzing reviews, a major part of the architectural design should consider having an application programming interface (API). Having an API, CoRAL provides open access to its data storage and retrieval services. This leads to widening the concept of CoRAL to a collection of tools, currently existing or coming in the future, that all have same goal but may approach it in different ways. The envisioned collection of tools, should serve at least in one of these ways: producer of reviews or consumer of reviews.

In this research, we introduce CoRAL not just with an API but with two other main components: the CoRAL Viewer (referred to as *the Viewer*) and the CoRAL Reporter (referred to as *the Reporter*). The Viewer can be considered in the category of producers of the reviews, where it also is a light consumer. The Reporter is a prototype of ideas for consuming the reviews in the form of visualized reports. In general, the Viewer and the Reporter are representatives of custom applications that perform these tasks: browsing and annotating scientific publications, and producing valuable reports about activities being formed around the publications. The CoRAL API is a service waiting for requests from the mentioned applications serving them with data storage and retrieval. An overview of the system architecture is illustrated in Figure 3.1.

As Figure 3.1 [1] suggests, CoRAL has a RESTful architecture. The components are connected via their REST connectors over the HTTP protocol. As defined in

[1] The notation used for the REST connectors is taken from Roy Fielding's dissertation who introduced RESTful architectures [31].

the REST basic description, the connectors encapsulate the methods for accessing resources and transferring resource representations [32]. Connectors are used to hide the connection details and implementation from the component. There are various connector types including server, client, cache, resolver, and tunnel [32]. The connectors used in the CoRAL's basic architecture are server (for the API) and clients (for the Reporter and the Viewer). This means that the Reporter and the Viewer are responsible for initiating the requests, while the API only accepts and responds to the requests.

Being implemented by the principles of REST, the scalability of the interactions is increased and the implementation of the components is simplified [31]. These are achieved by keeping the interactions stateless and hiding the connection details from the components. In the REST terminology, we should also define the *resources* in the context of CoRAL. Resource refers to a set of entities that are mapped to a *concept* that are to be transferred, accessed, or represented in the system. In CoRAL, resources include: any form of scientific publications (papers, presentations, posters, websites), persons (authors, reviewers, researchers, friends), and annotations (comments, tags, ratings). A conceptual model of the resources is illustrated in Figure 3.2 which is discussed in Section 3.1.

Data is transfered in JSON (JavaScript Object Notation) between the API and the components. JSON is a light-weight language-independent data interchange standard [33].

A typical flow of the events in CoRAL is as follows. Annotations made on a scientific publication are collected from the review producer client (e.g. the Viewer) and sent to the API. The API also expects information about the scientific publication and the reviewer. On a reverse direction, the producer can also fetch annotations from the API to show them to the user in the context of the document. The review consumer client (e.g. the Reporter) dynamically updates its reports based on the data collected by the API. The producer or the consumer can also provide other features such as social networking, search, and profile management.

The CoRAL architecture is open to support various usage scenarios. We have envisioned three basic scenarios. The first usage scenario involves a familiar desktop application, designed to be used by researchers reviewing published literature

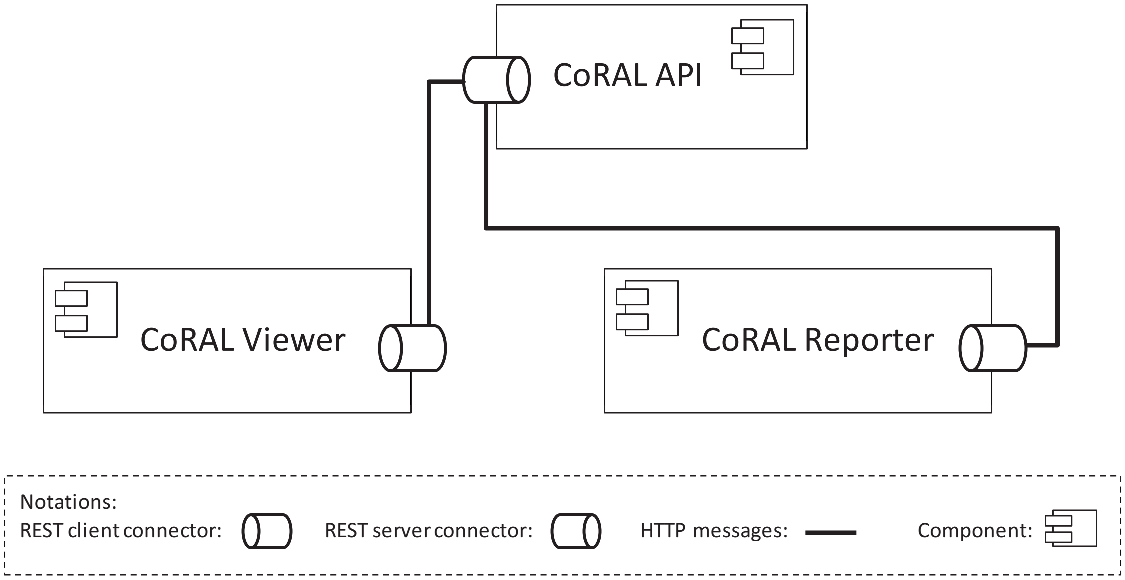


Fig. 3.1. CoRAL's component diagram illustrated with REST connectors

(referred to as *basic scenario*). The second usage scenario is inspired by the conference review process in which the system should deal with privacy management and conference settings. The third scenario deals with giving mobility to the first scenario and using it in the conference paper and poster presentation sessions. So, we can have a family of CoRAL applications that all support same goal and the data is located in a common place that all custom applications, useful for variety of usage scenarios, can access it. This leads to having a strong collection of reviews, independent of where, when, and how a review is made.

In the rest of this chapter, we describe the CoRAL components (the API, the Reporter, and the Viewer) based on the basic scenario of usage (Sections 3.1 through 3.3). In Chapter 6, we point to ideas for extending the architecture to the two other scenarios mentioned above.

3.1 The CoRAL API

The CoRAL API is a RESTful interface that provides both storing and retrieving of reviews and related data. Using simple HTTP requests and responses, a custom client can interact with CoRAL in order to use CoRAL data or add to the collection of reviews.

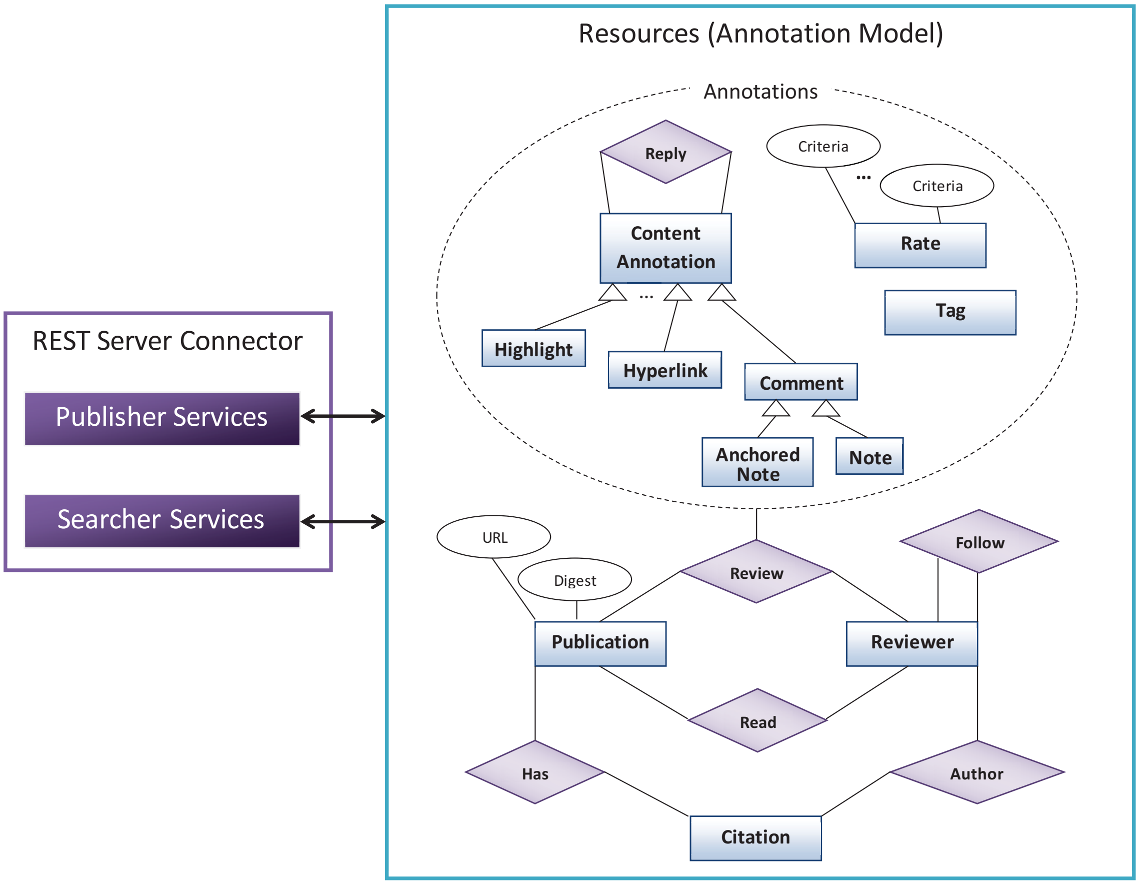


Fig. 3.2. The conceptual model of the API and the Entity-Relationship diagram of API's REST resources

Figure 3.2 shows API's inner modules by illustrating the REST resources and their relationships (the logical annotation model) in an entity-relationship diagram. The figure is a summarized version of the annotation model, eliminating the details and some of the entities for clarity. Figure 3.2 shows the semantics of the annotation model, not the implementation details. In the actual implementation, the entities and their relationships are persisted in a relational database (IBM DB2), separate from the logic of accessing the resources and/or their representations.

The REST Server Connector listens to the clients' requests through two main service containers: the Publishing Service, and the Searching Service. These two containers service the requested CRUD operations [2]. The Publishing Service consists of service handlers (bound to one base HTTP address) that manage requests and responses regarding creating, updating, and deleting CoRAL resources (the CUD

[2] Create, Retrieve, Update, Delete

operations). A client may add a scientific publication or post a comment, a tag, or a rating on a scientific publication using one of the relevant services in the Publishing Services category. This is done by sending an HTTP request to an address which its base URL is determined by the URL of the Publishing Service. On the other hand, the Searching Service container includes services responsible for searching and retrieving the resource representations (the R operation). Retrieving all comments on a scientific publication found by citation information is an example of services in this category.

A brief description of the elements of the annotation model comes in the following subsections. The API is implemented in Java programming language. The frameworks and technologies used in the implementation include: Jersey [3] for REST implementation, Grizzly [4] to build a scalable server, Apache Maven [5] for software project management and build automation, XML, JDOM, JAXB, and JSON.

3.1.1 Publication

Scientific publications include any form of published or unpublished scientific articles such as websites, papers, posters, and presentations. A scientific article may appear on a website and yet get cited several times, and hence, needs to be considered as a scientifically reviewed document. As an example, in Chapter 2 O'Reilly's article about Web 2.0 principles [1] was cited which is also cited by many other publications such as [34] and [2]. A publication can be distinguished from other publications by having one or all of the following: URI (can be local or on web), an encoding of the document content (binary digest or textual content digest), and citation information.

The document's *URI* can be used to access a document through its location that can refer to the exact same copy or a changed version of the document (specifically in case of websites). For local documents, residing in the user's machine, the URI should start with the user's unique identifier (username) followed by the local file address. This way, local addresses can be stored in CoRAL's web storage without making conflicts between different user's local copies. For web documents, the web address is stored as the document's URI and other document information such as

[3] http://jersey.java.net/

[4] http://grizzly.java.net/

[5] http://maven.apache.org/

digest can be used to further distinguish multiple versions of the publication.

In order to locate a publication independent of the varying addresses of its numerous copies, CoRAL stores a document digest in these two forms: *binary digest* and *textual-content digest*. This idea is taken from the Fab4Browser's mechanism to retrieve annotations independent of the location and format of the document, which itself is based on the idea of the lexical signatures [27]. Keeping CoRAL compatible with Fab4Browser, the MD5 algorithm [28] is used to create a unique identifier of the publication based on its content keeping or ignoring the format. The binary digest is an MD5 encoding of the document's binary content which encapsulates both document content and format. The textual-content digest is the MD5 encoding of the textual content of the document eliminating the white spaces and all formatting elements. CoRAL does not store the publication's full content upon a reviewing action. So, the digest data should be calculated by the client and sent to the API at the time of adding a publication, if the client wants to benefit from this feature.

As another identifying mechanism, CoRAL also takes advantage of *citation information* of scientific publications. Title and list of authors form the identifier of the publication which is used for collecting annotations over various copies of one publication. Abstract, keywords, DOI, ISSN, ISBN, conference or journal name, and other bibliographic information are also collected, if available. These extra bibliographic information can be used for categorizing publications, providing more information, and for citation purposes. More discussion on this can be found in Section 3.1.4.

Each of the methods above, together or individually, can be used to customize retrieving reviews on a publication. Choosing URI leads to retrieving annotations that were made on a known file location or website. Document digest should be used to find out what reviews have been made on any copy of a publication with the exact same content considering or ignoring the format, which is useful to find reviews of other reviewers without sharing copies. Similarly, citation information is useful to find reviews on the conceptually or bibliographically same documents. Availability of citation information also adds value to the reviews by providing meaningful and classifiable information about the publications.

Since the binary digest only helps to find format matchings, citation information can play an important role when the document is an image and the textual content digest cannot be computed. Also, use of document digest is vulnerable to document collision as the underlying encoding algorithm (MD5) is so [35]. This problem can be mitigated by choosing a more reliable encoding algorithm such as the family of SHA algorithms (refer to [36]). Although this can change, we have chosen the MD5 method to keep CoRAL compatible with Fab4Browser. Instead, having citation information can count as a backup method in case a collision happens. Although there are problems with use of digest, it is still an added value to the publication-identification method specially if a client does not support citation extraction (such as Fab4Browser).

The choice of retrieving a publication based on URI, digest, or citation information is given to the client or user to add flexibility to retrieving reviews based on client's dynamic needs and expectations. However, the URI is required by the API in order to enable finding reviews on the copy that they were initially made on, which is considered to be a basic functionality for a reviewing tool. In order to keep records consistent, whenever citation information for a publication with known document digest is recognized, all other publications with equal document digest are updated to refer to that citation information if they don't already. If an already existing citation record has less data than the new one, the new data is added to previous one. If a conflict arises, previously known citation information are kept without change and the new citation information is only recorded for the publication it referred to. In this way, a client may add value to citation information extracted by other clients.

3.1.2 Reviewer

Every micro-review is produced by a reviewer. A CoRAL profile is created for every reviewer that can be connected to her reviewing activities. For every action that leads to a change in the reviewing data, e.g. posting or deleting a comment, or in the personal information, e.g. profile settings, the user must be authenticated by CoRAL.

Profile settings can be managed by users to define their privacy rules. Privacy rules target revealing personal information and reading activities. Personal information includes name, affiliation, email, list of followed reviewers, and publications list. CoRAL does not reveal individual ratings to the public. But, comments are always public. We assume that users agree to publish their comments to public when they send them to the CoRAL API. CoRAL needs public reviewing activities in order to reach its universal reviewing goal and it is not supposed to be a storage space for private notes. However, CoRAL API receives its data from clients that support the reading and reviewing environment for reviewers. So, the reviewing client should enable the users to store private reviews and give the option to perform public posting to CoRAL.

Using CoRAL, reviewers gain the opportunity to find out about other researchers who are interested in the same content as they are. CoRAL also provides light mechanism for social networking in a basic way. Reviewers can indicate their interest in someone's reviewing activities by following them. In the current version, this is only a mechanism to express interest in someone's work. However, this mechanism can grow and support more features related to social networking. On the other hand, there are various academic social networking tools with advanced features that are widely used. Since it will be difficult for users to maintain relations in various tools, a better option may be integrating or migrating relationships from other related tools upon user's permission. This topic remains open for future improvements.

3.1.3 Annotation

Annotations include three main categories: content annotations, tags, and ratings. All types of annotations should participate in a reviewing relationship that must have three participants: annotation, publication, and reviewer. These three elements define a micro-review and give enough information to analyze reviews and reviewing activities. Although reviewing of the reviewers is also possible, in CoRAL the main concern is reviewing the literature.

3.1.3.1 Content Annotation

Content annotations include any type of annotation that refers to a part of the publication and has a reviewing value. Examples are notes, anchored notes, citations, and highlights. Notes and anchored notes are comments attached to the publication. Citations can be in the form of hyperlinks, linking a piece of the publication's content to another publication. Highlights span over the words inside the publication, showing a form of interest to that content. The reviewing value can be extracted from the textual body of the annotation itself (e.g. comment written in a note) or from its location in the publication (e.g. the highlighted area).

The CoRAL API expects clients to indicate the type of the annotation, the annotation body, and means to determine annotation's conceptual location. Annotation's conceptual location can be the related sequence of words or indicating the publication as a whole. Other information such as the physical location of the annotations can also be provided as an optional information.

Some types of content annotations can support discussions. For example, a note can be a reply to another note creating a thread of discussion in the context of the publication. This method supports effective and in-depth collaboration by use of basic annotating methods. So, CoRAL allows annotations to point to another annotation to indicate replying. It does not put restrictions on the type of content annotations that participate in this replying relation, so that every client can have its own replying technique. However, the standard type can be considered to be a note as a reply to another note.

Currently, an annotation can only point to one other annotation. This method can be improved by allowing replying to any number of annotations, supporting a broader and inter-reviews discussion.

Other supporting data can also be sent to the CoRAL API about the content annotations such as attitude and weights. Attitude refers to the qualitative message of the annotation, indicating whether it is positive, negative, informative, neutral, or none with respect to the content that it points to. Knowing the attitude of the reviewer in the comment that they make, is a valuable piece of knowledge and can itself be considered as a micro-review. Clients may ask users to explicitly indicate their attitude at the time of annotating, as an optional information which adds to the value of the review. The current version of API, only supports receiving attitudes from the clients at the time that the annotations are posted. However, another idea is to let the other reviewers vote on the attitude of each comment, at any time later. In this method, we can still receive the annotator's attitude as her initial vote.

Annotations can also be weighed by reviewers other than the original annotator. CoRAL expects binary values for weighing: like/dislike, agree/disagree. These votes can be posted to the API anytime after the original annotation is posted. They are stored in a cumulative way to show the number of likes/dislikes on a given review, restricting a user to weigh a comment only once by overriding their previous decision.

3.1.3.2 Tag

Tags are useful for categorizing publications and finding relevant literature without specifying any level of granularity. Tags can be considered as useful pieces of review in two ways. First, they help to find related literature which is useful for connecting concepts, publications, and reviews. Second, the collection of tags that a user attaches to publications with her own terminology and level of granularity gives an insight on that user's perspective with regard to the publications, in addition to indicating the user's interests.

In CoRAL, similar to most of annotating systems, users are free to use their own wording in the tags. Also, multiple tags can be assigned to every publication by each user, which allows putting a publication into every related conceptual category.

There have been many studies on use of free vocabulary in social tagging systems and the possibility of emerging a folksonomy as a result. "folksonomy is tagging that works", quoting Thomas Vander Wal who coined this word [37]. In other words, folksonomy is the vocabulary that emerges from social tagging and can be used as a content classification system. Halpin *et al.* [38] showed that given sufficient active users a limited set of stable tags emerges from social tagging over time. Although the number of stable tags seems to be a small portion of all the tags in the system, still they are found to be useful for classification in a high level of granularity.

The low-level detail-specific tags have much less chance of being used by many users than the high-level well-known categories. This happens for two reasons: more publications fall into a high-level category than a detail-specific subcategory, and high-level categories usually have standard and widely used names while subcategories are more open to user's idiosyncrasy. However, we are unaware of tendency of users towards using low-level tags in contrast with high-level tags that can affect the tags' frequencies. Anyhow, we find tagging useful for this reviewing system even if a low-level folksonomy does not emerge. The reason is that if one person discovers a relation between two publications in a way that can be expressed by a few words (tag), then another researcher can benefit from this relation if they are provided with those words. Based on this idea, CoRAL provides all tags made on a publication to reviewers who are interested in that publication so that they can find other related contents from various perspectives. CoRAL also takes advantage of tags assigned by particular reviewers to identify individual interests.

3.1.3.3 Rating

Rating provides a quantitative reviewing method that can be easily and quickly interpreted. Aggregated ratings, e.g. average rating over all received rates, shows the community's overall judgment with respect to the publication. Aggregated ratings become more reliable when more reviewers contribute to the rating, which leads to having a more universal outlook while making outliers less effective. Raters may rate a publication by only one rating touchstone that should be a total rating based on overall judgment of the reviewer about every aspect of the publication. This can be useful to bring a quick and somewhat easy ranking of the publications or judgment on one publication. However, a publication can be rated based on more than one criterion that is routine in the rating process in the conference reviews. Common rating criteria include rating based on novelty, clarity, structure, usefulness, and total decency of the publication (referred to as *rating criteria*). CoRAL chooses these rating criteria for its rating process, since this will help separating reviewing concerns when rating a publication. It also adds to the value of rating by containing more information than just a total rating. All ratings are made in the scale of 1 to 5, 5 being the best score. Having an odd number of possible choices (5) gives a chance of rating with the middle value to indicate neutral. Not having a large range of possible values for rating (1 to 5) increases the possibility of agreeing upon a value when reviewers agree upon the quality. Although having a wider range could help distinguishing more levels of quality, a smaller range is easier to decide and agree on and it reduces the effect of personal attitudes.

Rating is not only a valuable micro-review but it is also beneficial for deriving a good publication searching method. By injecting the rating data into the algorithm for ranking search results, one can obtain more convincing ordering in the search results. Having more than one criterion, each targeting one aspect of a scientific publication, is also in favor of goal-based search. By goal-based search we mean searching by being able to give a hint about goal of the search in addition to the keywords. For example, if a researcher aims to find publications in a topic with hope of coming with ideas for defining a new research project, it will help if results are sorted by having more novel publications at the top. In this particular example, use of aggregated rating for the novelty criterion becomes useful.

3.1.4 Citation Information and Authoring

According to the discussion we had on scientific publications (Section 3.1.1), citation information are used for identifying various copies of a publication and for providing metadata about publications.

There are two ways to access citation information for a given publication: provided by the client at the time of adding or reviewing a publication, or querying scholarly databases for relevant information. Either way, this information should be open to updates due to possible errors and existence of missing fields in both methods. However, the title and optionally the authors must be provided by the client which has access to the document content. Any querying method would need this information to further expand the collection of citation facts.

There are various tools, specifically the library management tools, that have incorporated ways to extract citation information or find citations by querying libraries. CoRAL's aim is to be integrated with or accessed via such tools as they are convenient ways to review literature and organize readings. For this reason, although citation extraction is a known procedure that can be seen in most of related tools, in this research we decided to seek for this information from the client. However, this feature can be considered as a potential future improvement in order to support clients that do not incorporate such a mechanism.

The API accepts citation information in two forms: list of attributes and their values, or the complete information in the form of one BIBTEX entry. The API is capable of parsing a healthy BIBTEX entry, extracting field-value pairs, and distinguishing the available identifying parts: title, author names, DOI, abstract, ISBN, and ISSN. The identifying attributes are stored for indexing and easier extraction at a later time. The BIBTEX entry is saved to be provided to the users who are seeking for it. If the information is provided in a list of attributes, not in the BIBTEX form, a similar action is taken: the identifying information are indexed, a BIBTEX entry is created filled with the given information and it is saved for future access. As described in Section 3.1.1, a citation information and consequently the BIBTEX entry is updated upon receiving more information.

Having citation information, CoRAL also can have an ordered list of authors' names. It would be interesting if publications could be categorized based on their authors, coming up with analysis about authors being reviewed. But there are many cases of inconsistencies in printing names on publications: inclusion of all parts of a long name or not, and identical names or abbreviations for different people. Affiliations and emails could help to distinguish these cases, but they have their own problems: it is less likely to have access to these information in a citation information, and people can change their affiliations and emails.

CoRAL saves the author's list, with no further analysis. The only usage could be showing information about the publication and searching based on author names. However, as every reviewer has a profile and every author can also be a reviewer, CoRAL provides means for the reviewers to list their publications and claim being an author. All co-authors of a paper can claim the paper. In this way, the publications are linked to real identities without bearing the risk of mistake. CoRAL may have less instances of authoring relation, but every one that it has is reliable and carries more value with linking to the author's profile and real activities.

3.2 The CoRAL Viewer

As previously discussed in the beginning of Chapter 3, the CoRAL Viewer that is described in this section is a prototype or an instance of possible applications that can produce and somewhat consume reviews in a CoRAL system. The CoRAL Viewer is an annotating tool built as an extension to the Fab4Browser (discussed in Section 2.5).

Figure 3.3 illustrates the extensions made to the Fab4Browser to make it a better candidate as a CoRAL client. Note that details of the Fab4Browser's architecture cannot be understood from this figure. In this figure, the plus signs show what

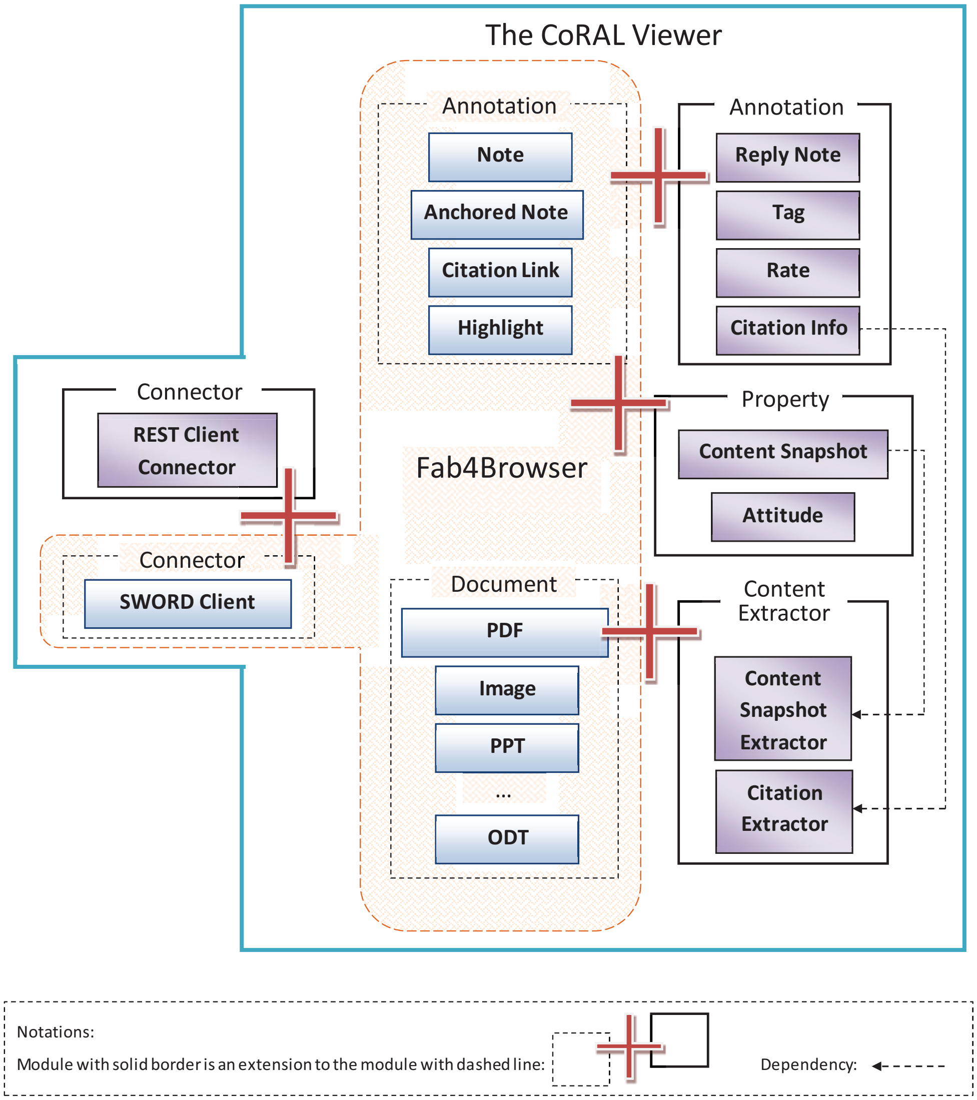


Fig. 3.3. The CoRAL Viewer: an overview of extensions to Fab4Browser

extension is added to which Fab4Browser module. The REST client connector is added to make the Viewer capable of being part of the REST architecture described in Figure 3.1. A set of new annotations (reply note, tag, rate, citation information) are added to the annotations originally supported by Fab4Browser. Fab4Browser handles browsing and annotating various document types including PDFs. For the PDFs, major format for scientific publications, the Viewer adds two analysis modules: content snapshot extractor and citation information extractor. The content snapshot is referred to a set of words or lines around a physical location in the PDF. It is useful for extracting most relevant part of the content that an annotation is made on it. The citation information extractor module extracts as many information as possible from the PDF's textual content, such as the title, list of authors, and abstract. Speaking of the extensions, two major properties added to the annotations include content snapshot and attitude. Content snapshot is the set of words that the annotation, mainly a note, refers to. When a comment is placed in a position on a PDF file, the content snapshot extractor module extracts the content starting from two lines before the position and ending at two lines after that position. The extracted content is then added as an attribute to the comment, giving it more meaning by carrying part of the context that it is referring to. As another attribute, user can indicate her attitude in making a comment (comment can be positive, negative, neutral, informative, or none) which will be added to the annotation properties.

3.2.1 Content Extraction

Fab4Browser parses the documents and creates a document tree for each page of the document. This document tree creation is based on the Multivalent Document Model introduced in [39]. In the document tree, leafs contain format and text of every word. All other nodes contain information about structure, formatting, and type of the document. For example, for a PDF document at some level down the tree, there is a node named 'text', the 'text' node has 'line's as its children, and each 'line' is parent of several words.

The extractor modules traverse the document tree of the PDF files and analyze the position of nodes, specific words and their position, and change of font sizes. Content snapshots are extracted by finding beginning and end of lines around a given position. However, extracting the citation information is more challenging. The CoRAL Viewer currently supports basic information extraction. It finds the title expecting it to appear in the first few lines with a larger font relative to the surrounding lines. Surrounding lines may include copyright notice or a headline indicating the journal or publication name. Title is normally followed by the list of authors and their information, appearing in a smaller font. Abstract is expected to be after the authors list, separated by the word 'abstract' (with optional punctuation marks such as a colon) and with a distinguishing font relative to the authors list. Hints for finding the end of the abstract is more variant than the previously mentioned facts, but it can also be managed by following the changes in the fonts when starting a new line. In a similar way to finding the abstract section, the Viewer also tries to locate and extract the keywords. This mechanism is tested on several ACM and IEEE formated papers and it can work on most of the other formats of scientific publications. However, since it is dependent on the wordings and ordering of basic parts (title, authors, abstract, keywords), we cannot guarantee its functionality on all the publications. In addition, the paper cannot be analyzed in this way if it is an image. Other forms of scientific publications such as presentations and reports are not supported by this content extraction mechanism, although the title can usually be extracted as expected.

To find the 'text' node, that is the parent of the textual content, the BFS (breadth-first search) tree traversal algorithm is employed. The reason is that it is expected to be in the first few higher levels of the tree. After finding the relevant sub-tree under the 'text' node, that contains lines and words, it is faster to continue traversing using DFS (depth-first search). Because, 'line's are the immediate children of 'text', and words (the leaves) are immediate children of every 'line'.

The content snapshot extractor may need to go through all the lines in order to find the nodes positioned near to the specified location. But, the page on which the annotation is made is known and every page has a separate document tree. So, the cost of content extraction is bounded with the number of lines in a page and it does not depend on the size of the document. This position should be found considering the zoom scale and with attention to positioning of the annotation with respect to columns, in case of two column documents.

3.2.2 Added Annotation Types

As displayed in Figure 3.3, the annotation types added to the Fab4Browser's set of annotations include: reply note, tag, rate, and citation information. Fab4Browser supports addition of new annotation types by subclassing a class named Behavior. As the name suggests, Behavior is the primary way to add user-level functionality to the system [29]. This includes annotation types, media adapters for supporting new media types, printing, and many more. The Behavior classes define protocol methods handling events after and before actions such as build, paint, save, and restore. Adhering to this implementation method, the system framework will call the appropriate method upon occurrence of an event. This way, new types of behaviors can be added to the system without interfering with the work of others. So, the new annotation types added to this system are implemented as described.

Reply note: In order to support discussions in the context of the publication, which is an important part of collaborative review, the Viewer adds 'reply note' to the collection of annotations of Fab4Browser. Fab4Browser has two types of commenting annotations: note and anchored note. Other types of annotations may also incorporate a way to add comment such as commenting on a highlighted area. The note annotation works like sticky notes, it can be placed anywhere on the publication and comments are typed inside it. The space surrounding that note can be considered as the part of the publication that the comment is referring to. Anchored note is a note with an anchor or an arrow pointing to some part of the content. Since replying is usually in the form of a comment pointing to another comment, the reply note is basically an anchored note. The difference is that, an anchored note points to the publication's text but the reply note points to another annotation. Having some notes pointing to each other can form a discussion thread, and the reply notes make the discussion thread distinguishable from other set of notes.

Currently, the reply note is designed to point to only one other note. However, as mentioned earlier, the system can be improved by having inter-note or even inter-publication commenting and discussions.

Tag: Tagging is a new behavior added to Fab4Browser in the form of annotating, similar to commenting. User can create a tag note and assign as many tags as she wishes in one tag note. Tags can have multiple words, separating tags by new line. Also, set of assigned tags by other reviewers is made available to the user so that they can read the paper and find related tags at the same time.

Rate: Being consistent with the CoRAL API's rating decisions, the Viewer lets user to rate publications based on the 5 optional criteria explained in Section 3.1.3.3. They may change their rating decisions and override the previous ratings by re-rating the publication.

Citation information: Although the Viewer attempts to extract the citation information from the publication, users also have the option to provide the system with this information or more. User can annotate the publication with adding or editing the citation information including title, authors, keywords, and DOI. If the title is extracted or provided by user, all others are optional. Otherwise, title must be provided in order to add more information.

User can do this by filling a form that contains major parts of a citation information, or she can provide the system with a BIBTEX entry containing the citation for the publication. In both ways, the information is sent to the API upon publishing the annotation to public.

Providing the citation information by user's will, mitigates the problems with automatic extraction specially for formats and forms that are not supported by the extractor module.

3.2.3 REST Client Connector

The Fab4Browser uses a common interface to save public and private annotations. This interface defines methods to post, search, and delete annotations using the publication information. For the private annotations, Fab4Browser serializes the annotations into XML using JDOM [6]. The XMLs are then stored in the user's local machine, indexed by Apache Lucene [7]. For the public annotations, the interface is implemented with the SWORD protocol [8] and the annotations are deposited into the remote Fab4Browser repository in the form of XML/RDF files. The SWORD protocol defines a standard way for depositing content into a server. The Fab4Browser server stores and indexes the annotations using the Cheshire3 XML search engine [9]. Then, the posted annotations are searched and retrieved by use of the SRU search protocol, which is an XML-focused Internet search protocol [10].

In the CoRAL system, the annotations are sent to an API for being further analyzed and used by many clients. In CoRAL, we are more interested in making the annotations ready for easy analysis rather than storing them in a repository. Considering the discussion we had about usefulness of the RESTful architecture at

[6] http://www.jdom.org/

[7] http://lucene.apache.org/java/docs/index.html

[8] http://swordapp.org/about/

[9] http://www.cheshire3.org/

[10] http://www.loc.gov/standards/sru/

the beginning of Chapter 3, we decided to have a REST client connector and do not use the SWORD implementation for the CoRAL Viewer. Meanwhile, in CoRAL we are interested in relations between elements of a review which is best analyzed with use of a relational model rather than analyzing large volume of XML entries. This decision also effected the use of SRU search and the Cheshire3 engine. In addition, reusing of the Fab4Browser's SRU search implementation and the Cheshire3 search was not in CoRAL's favor since it requires the clients to know how to create a query with specific format for every need they have. In contrast, using simple HTTP verbs (namely POST and GET) and the concept of resources in the REST architecture make it easier to use a REST client connector. By this decision, we hope to encourage more clients to connect to the CoRAL system.

However, we could reuse the Fab4Browser's mechanism for dealing with private annotations. In the CoRAL Viewer, the public annotations are serialized and parsed the same way that private annotations are: using JDOM to serialize/parse annotations into/from XML. The XML representation of the annotations, is then received by the API. The API parses the individual XMLs once and persists them in its relational annotation model.

3.3 The CoRAL Reporter

The CoRAL Reporter is a visualization of collaborative review. It is a prototype of ideas of how gathering reviews in the CoRAL system can be used and analyzed. In this work, the attempt is to take advantage of all three sides of a review - reviews, reviewers, and publications. The CoRAL Reporter aims to provide reports from various perspectives derived from the reviews: properties of the micro-reviews such as the content of comments and attitudes, interactions and collaborations, properties of publications and reviewers, different information obtainable from different types of micro-reviews, and timeline of events. The ultimate goal is to visualize reviewing activities, reviewers, and publications in order to give a better understanding of the collection of reviews, guide research, highlight ideas and interesting publications, support global collaboration, and improve quality of critical analysis of literature.

As illustrated in Figure 3.4, the Reporter has 4 main components: reviewer

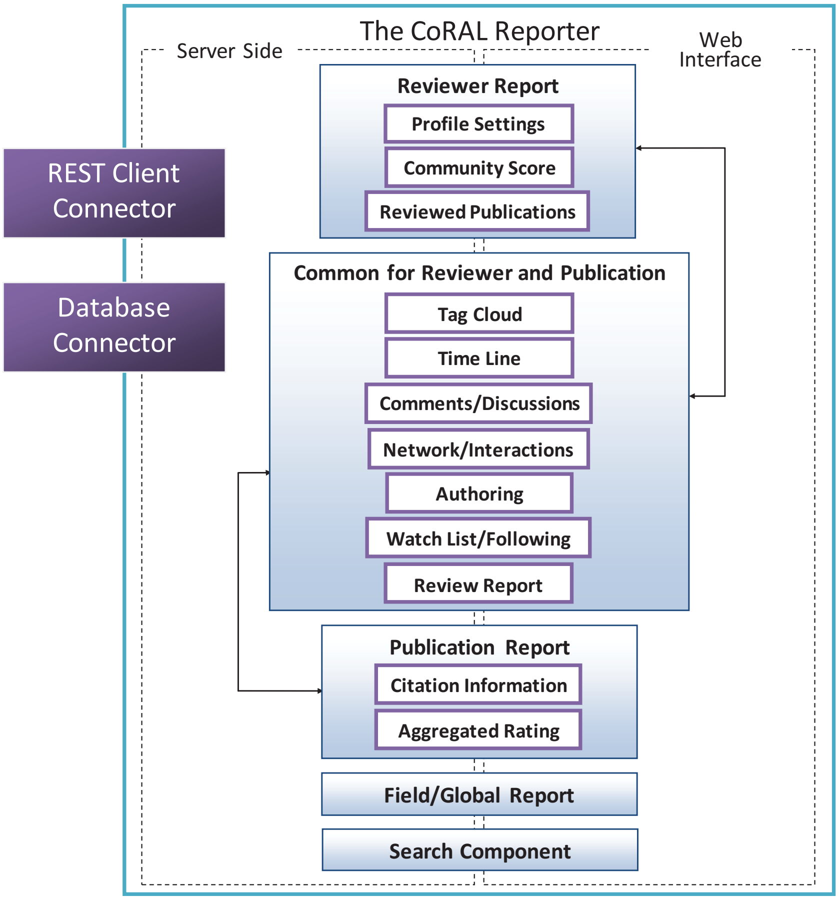


Fig. 3.4. The components of the CoRAL Reporter

report, publication report, field or global report, and the search component. These are responsible to generate visualized reports about reviews or help managing and finding information in the system. Each component is composed of two sides: a server side and a web interface.

In addition to the reporting and searching components, the Reporter needs connectors to retrieve the information it needs. As previously discussed in Figure 3.1, the Reporter is expected to have a REST client connector in order to contact the API and receive reviewing information. The Reporter can get all the information it needs from the API, but the data is given in a general form without being processed and filtered for special needs. In some cases, the general form of data is enough to create the answer that the Reporter needs in an efficient way. For example, the list of reviewers that a certain user is following can be retrieved directly from the API. But in some cases, given the current limited methods supported by the API, the Reporter should contact the API several times and parse and process on a higher volume of data in order to generate the data specialized for the Reporter's need. On the other hand, if the Reporter has access to the database it can process data much more efficiently. Since the Reporter and the API are both sitting on the same machine and both are implemented by the same team, it is decided to allow the Reporter to access the database for specialized data querying. Considering that the Reporter is just a prototype designed to visualize the potential benefits of the CoRAL system, this is an acceptable decision for a quick response. So, the Reporter also contains a database connector that is responsible for querying the review database. For future improvement, the API can be extended to be capable of processing more specialized queries so that detailed report questions can be answered directly by the API in one step. However, since every client may have their own needs and questions, we believe that the general form queries such as retrieving all annotations of a given publication will remain as the main queries asked from the API.

Each component has a server side and a web interface part. The server side has access to the data through either the REST connector or the database connector. The server side processes data and generates the reports and the web interface displays the reports. The server side is implemented by PHP. In the web interface side, some elements are visualized by use of HTML and JavaScript. For a few visualizations in the web interface, these existing libraries and APIs are used: Google Annotated Time Line [11], JavaScript Bluff API [12], and JavaScript InfoViz Toolkit (JIT) [13].

The details of the web interface and functionality of the reports is best described when explaining the system usage rather than the architecture. So, this chapter gives an overview of the components that build the Reporter and more details about each of these components comes in Chapter 4.

3.3.1 Publication Report and Actions

The most useful way to access a summary of reviews is having them accessible through the scientific publication itself. The Reporter collects the micro-reviews

[11] http://code.google.com/apis/chart/interactive/docs/gallery/annotatedtimeline.html

[12] http://bluff.jcoglan.com/api.html

[13] http://thejit.org/

around a publication in the publication-report component. The publication-report component contains a collection of tasks and visualizations related to the publications being reviewed in the CoRAL. As explained in Section 3.1.1, a publication is recognized by its URI, citation information, or digest. The Reporter only generates reports based on the citation information so that it can group the conceptually same articles in one publication report that its title as a scientific publication is known. Use of URI is not appropriate here, since a web-based URI may point to a dynamic content and a local URI is only useful for retrieving one's annotations and does not carry any information about the content of the publication. In addition, it is useful for retrieving annotations on only one copy of the publication accessible through a single URI. Use of digest limits the report to publications with same content (including the headers that can be different in various versions of a publication) and/or same format, while suffering from the document collision discussed in Section 3.1.1. Aside from these reasons, citation information is the best choice since it gives us a necessary piece of information about the publication: its title. For a valid scientific publication to have the minimum citation information (e.g. title and authors) it does not need to be a published paper. Although the Reporter does not depend on URI to identify a publication for report generation, a scientific web publication can also have a report, if its citation information is available.

The Reporter generates an interactive report about reviewing activities on a publication that contains: information about the publication taken from the citation information; related thread of discussions and individual comments including date and attitude of the comments, pointers to the reviewers, and utilities to sort or filter the comments; publication's tag cloud; time line of reviews; network of interactions between reviewers that visualizes the amount of contribution and general attitude of each reviewer; aggregated rating for all the rating critera; and facility to collaboratively edit and update the citation information. When viewing the report, users can take actions such as: adding the publication to their watch list or claiming the publication as their own work.

3.3.2 Reviewer Report and Actions

Having the collection of reviews, CoRAL also has access to the reviewing activities of the reviewers in a global or inter-publication extent. Also, every reviewer has a profile that should be managed by them in an easy web interface. The reviewer report component consists of actions that a reviewer can take in addition to a visualized summary of her reviewing activity.

This reviewer-centric component involves actions and visualizations including: list of publications that the user has reviewed; a summary of user's comments with links to the reviewed publication's report component; displaying the user information with support for changing this information and privacy settings (as introduced in Section 3.1.2); time line of reviewing activities; tags that the reviewer frequently assigns as an indicator of her interests; aggregated tag cloud of the publications that this user reviews; network of her interactions with other reviewers over any publication; the list of her publications with ability to claim and add existing or new publications; list of reviewers and publications that this user is following; and a report of the user's different types of activities summarized into a community score.

3.3.3 Global Report, Field Report, and Search

In order to give an insight about the most interesting publications being reviewed in the CoRAL, the Reporter generates a few global statistics. The top few publications that have been reviewed more than other publications; the top rated publications according to the 'total' rating criteria; and the publications being read more than others are the global statistics currently available in the CoRAL Reporter.

A field report can be achieved by finding a list of publications assigned with a specific tag, where the tag identifies the field. Considering the discussion we had in Section 3.1.3.2, based on converging to a folksonomy containing tags for major or high level categories, we can hope for having major fields automatically identifiable by frequent tags.

There can be a wide range of global or field-specific reports and statistics supported by the data gathered in CoRAL. The above mentioned statistics are a few samples provided in this prototype. Currently available services, specifically Mendeley, provide a broad range of such statistics over a large user-set. So, in this work we refrained from repeating same feature, hoping that services such as Mendeley will consider integrating with our service.

To ease finding reviewer profiles and publications, and hence the relevant reviews, the Reporter supports basic search for the publications and reviewers. The search component has basic functionality that supports searching keywords in the publication titles.

3.4 Summary

In this chapter, we illustrated an overview of the CoRAL architecture and mentioned that new components are expected to join the system through a REST connector. Then, we introduced the CoRAL API, the Reporter, and the Viewer in terms of an overview of their internal components. In addition, we discussed each design or development decision that we have made in building the components.

Talking about the API, we illustrated the underlying annotation model (Figure 3.2) that consists of scientific publications, reviewers, annotations, citation information, and the relations between these entities. Regarding the publications, we identified three methods for finding various copies of a document that can be used for selective retrieval of related reviews: document URI, document digest, and citation information. In a discussion related to the reviewers, we talked about reviewer profiles, activity recording, and managing privacy settings supported by CoRAL. We categorized annotations into three main categories: content annotations that are connected to the document content, tags that are used for social identification of themes and categories, and multi-criteria rating for quantitative review of publications. As a side note, a content annotation can serve as a reply to another annotation. We expressed that we care for context and attitude awareness in collecting content annotations. This can be achieved by expecting content annotations to carry information about the attitude of the reviewer and piece of content that the annotation refers to. We also discussed citation information management and authoring.

Section 3.2 in this chapter discussed the Viewer and its development. The Viewer is built as an extension to the Fab4Browser to make it more suitable to serve as a collaborative-reviewing client in the CoRAL system. Tagging, rating, reply notes, and user provided citation information are added to the Fab4Browser's basic set of annotations. In addition, user can optionally specify her attitude when adding a comment or reply. For the PDF file formats, the Viewer adds support for automatic extraction of citation information from scientific publications. A snapshot of the document content that a content annotation is more likely referring to, is extracted by content extractor module in the Viewer which increases the context-awareness of the annotation. A REST client connector is added to Fab4Browser to make it comply with the design decisions made for CoRAL.

The final section of this chapter talked about the Reporter and its components. These components are: publication report, reviewer report, tag and global report, and search. The publication report is responsible for generating summaries and visualized reports about reviews, reviewing activities, reviewers, and other data gathered around the reviewed publications. The reviewer report is a web page generated for each CoRAL member containing reports about their reviewing activities and profile management tools. Tag and global report show a list of reviewed publications that correspond to specific tags, or a list of publications that are ranked high in a reviewing metric such as in ratings or number of comments. The search component is used for searching for publications and finding reviewers.

Chapter 4

Usage and Storyboards

CoRAL consists of three components: the API, the Viewer, and the Reporter. The Viewer and the Reporter are the ones that embody and illustrate the actual usage of the CoRAL system. Users of CoRAL have direct interaction with these two components and other possible clients that may join the system. The user does not deal with the API. It is accessible through these two components and other future clients. In this chapter, we show how CoRAL is perceived by users from the Viewer and Reporter's perspective. We discuss the features and illustrate sample storyboards of the system use cases through these two components. Considering that we envision addition of other clients to the CoRAL system, other use cases may be supported by the system through the new interfaces.

4.1 Usage of the CoRAL Viewer

The CoRAL Viewer is an extension to the Fab4Browser application which is briefly described in Section 2.5. Fab4Browser is a desktop document browser capable of browsing and annotating local files and web sites. The CoRAL Viewer adds support for replying, tagging, rating, attitude declaration, citation extraction or annotation, and content snapshot extraction to the Fab4Browser, in addition to linking it to the CoRAL Reporter.

4.1.1 Browser Utilities

The CoRAL Viewer inherits all the browser utilities from the Fab4Browser. It supports basic features that every browsing application, specifically Internet browsers, are expected to support such as multi-tab browsing, bookmarking, searching in the content, and printing. Distinguishably, many document formats such as PDF, HTML, DVI, and JPG can be browsed and annotated in this browser all with same mechanism.

This browser supports two modes for storing and retrieving annotations: the public and private modes. In the public mode, authentication is required so that user can connect to the remote repository of choice: Fab4Browser repository or the CoRAL API. Fab4Browser and CoRAL API have some differences in the data they need and services they provide which is discussed in Section 3.1. Here, the CoRAL API is considered as the server.

Opening a document in the public mode sends the document information and reading data to the CoRAL API. Document information includes all or any of these: URI, binary digest, textual digest, and citation information as far as they are computable. Then, potential updates to existing information is performed, e.g. updating citation information for matching document digests or addition of a non-existing URI.

Content annotations such as notes are shown both in the document context (Figure 4.1(1)) and as a summary in a panel located in the left side of the document (Figure 4.1(2)). User can easily switch between public and private mode by selecting her choice on top of the annotation panel (Figure 4.1(3)). Using the private mode, the annotations are retrieved from the private repository which is in the user's local machine. In the public mode, annotations are retrieved from the remote server of the choice. User can save her annotations to either public or private repository, disregarding which source she chooses to retrieve the annotations from (Figure 4.1(4)).

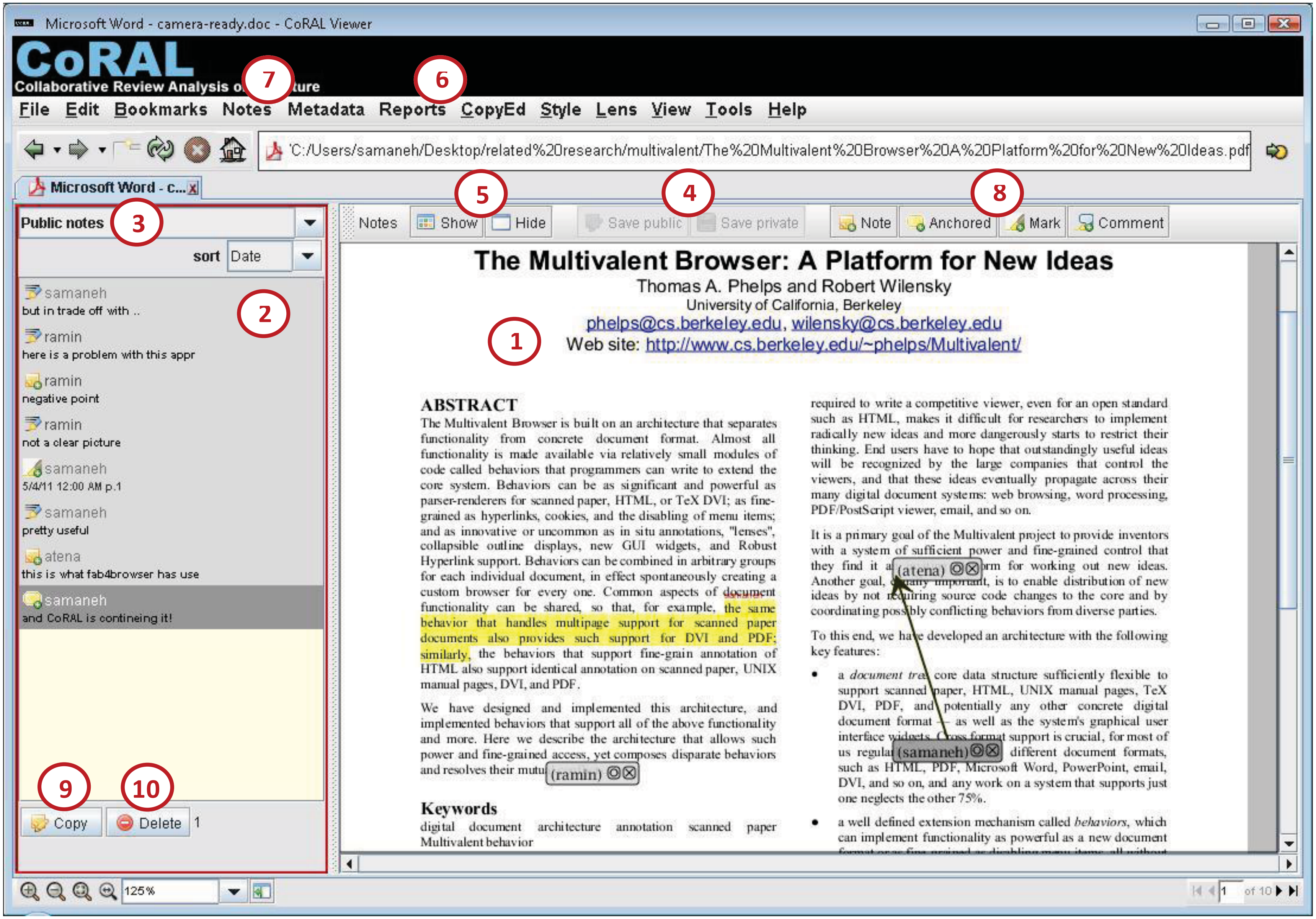
Upon opening a document in the public mode, the annotations should be retrieved from the remote server from the identified copies of the document. If the document is a PDF which is not a scanned image and its writing follows a standard paper format, then these are computable: the URI, the binary digest, the textual digest, and the citation information of the current document. User may decide to load annotations from the relevant document copies by choosing any combination of these options: same URI, same exact document (same binary digest), same content but possibly different format (same textual digest), and matching BIBTEX or citation information. Another given option is loading annotations from trusted parties only, that focuses on the author of the available annotations instead of the document copy. Choice of more than one document property leads to loading annotations from all of those choices with removing the duplicates. The 'trusted parties only' option would further filter the annotations against the annotators that the user specifies as trusted.

Loading the annotations may fill the document with content annotations such as notes and highlights. Tags and rates are not shown over the document content. The browser gives the option to hide or show all the content annotations on the document content (Figure 4.1(5)). When these are hidden, they can still be accessed via the annotation panel that contains a summary of every annotation and choosing any of them opens that annotation in its location on the document content. Figure 4.1 shows an article and its annotations browsed in the Viewer. The annotation panel can be found on the left and the document view is on the right. The notes in this figure are minimized to have a cleaner view. A minimized note contains the annotator's username as its subject.

When the user is viewing a document, she can find the report page relevant to that document in the Reporter by choice of menu options (Figure 4.1(6)). She can also navigate to her CoRAL profile that can be found on the Reporter accessible through the Viewer.

4.1.2 Content Annotations

Using both menu options (Figure 4.1(7)) and toolbar buttons (Figure 4.1(8)), user can create new content annotations including: note, anchored note, reply note, highlight or mark, comment on highlight, citation, and hyperlink. For the notes and anchored notes, user has the option to specify her attitude right below the content of her note. A reply note is created by choosing the annotation on which a reply is needed. Figure 4.2 illustrates various types of content annotations.


Microsoft Word - camera-ready.doc - CoRAL Viewer
CoRAL
Collaborative Review Analysis o ture
File Edit Bookmarks Notes Metadata Reports CopyEd Style Lens View Tools Help
'C:/Users/samaneh/Desktop/related%20research/multivalent/The%20Multivalent%20Browser%20A%20Platform%20for%20New%20Ideas.pdf
Microsoft Word - c...
Public notes
sort Date
Notes Show Hide Save public Save private Note Anchored Mark Comment
Copy Delete 1
125%
1 of 10
(atena)
(samaneh)
(ramin)
1
2
3
4
5
6
7
8
9
10

samaneh
but in trade off with ..

ramin
here is a problem with this appr

ramin
negative point

ramin
not a clear picture

samaneh
5/4/11 12:00 AM p.1

samaneh
pretty useful

atena
this is what fab4browser has use

samaneh
and CoRAL is contineing it!

The Multivalent Browser: A Platform for New Ideas

Thomas A. Phelps and Robert Wilensky
University of California, Berkeley
phelps@cs.berkeley.edu, wilensky@cs.berkeley.edu
Web site: http://www.cs.berkeley.edu/~phelps/Multivalent/

ABSTRACT

The Multivalent Browser is built on an architecture that separates functionality from concrete document format. Almost all functionality is made available via relatively small modules of code called behaviors that programmers can write to extend the core system. Behaviors can be as significant and powerful as parser-renderers for scanned paper, HTML, or TeX DVI; as fine-grained as hyperlinks, cookies, and the disabling of menu items; and as innovative or uncommon as in situ annotations, "lenses", collapsible outline displays, new GUI widgets, and Robust Hyperlink support. Behaviors can be combined in arbitrary groups for each individual document, in effect spontaneously creating a custom browser for every one. Common aspects of document functionality can be shared, so that, for example, the same behavior that handles multipage support for scanned paper documents also provides such support for DVI and PDF; similarly, the behaviors that support fine-grain annotation of HTML also support identical annotation on scanned paper, UNIX manual pages, DVI, and PDF.

We have designed and implemented this architecture, and implemented behaviors that support all of the above functionality and more. Here we describe the architecture that allows such power and fine-grained access, yet composes disparate behaviors and resolves their mutu

Keywords

digital document architecture annotation scanned paper Multivalent behavior

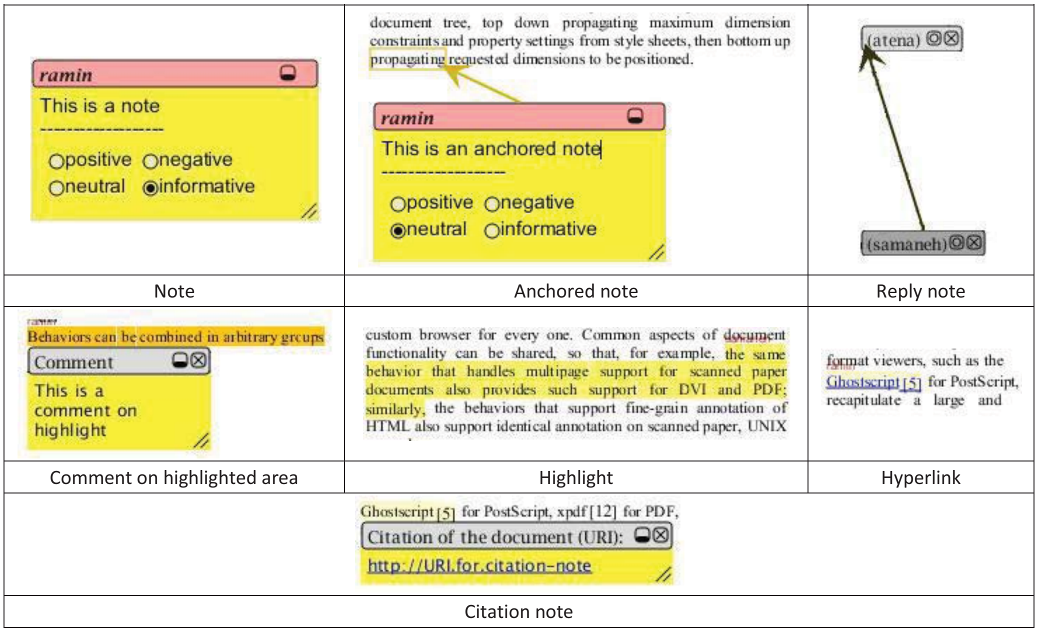
required to write a competitive viewer, even for an open standard such as HTML, makes it difficult for researchers to implement radically new ideas and more dangerously starts to restrict their thinking. End users have to hope that outstandingly useful ideas will be recognized by the large companies that control the viewers, and that these ideas eventually propagate across their many digital document systems: web browsing, word processing, PDF/PostScript viewer, email, and so on.

It is a primary goal of the Multivalent project to provide inventors with a system of sufficient power and fine-grained control that they find it a rm for working out new ideas. Another goal, ~~equally important~~, is to enable distribution of new ideas by not requiring source code changes to the core and by coordinating possibly conflicting behaviors from diverse parties.

To this end, we have developed an architecture with the following key features:

- a *document tre* core data structure sufficiently flexible to support scanned paper, HTML, UNIX manual pages, TeX DVI, PDF, and potentially any other concrete digital document format — as well as the system's graphical user interface widgets. Cross format support is crucial, for most of us regula different document formats, such as HTML, PDF, Microsoft Word, PowerPoint, email, DVI, and so on, and any work on a system that supports just one neglects the other 75%.
- a well defined extension mechanism called *behaviors*, which can implement functionality as powerful as a new document

Fig. 4.1. A document and its content annotations browsed in the Viewer



ramin

This is a note

○positive ○negative
○neutral ◉informative

document tree, top down propagating maximum dimension constraints and property settings from style sheets, then bottom up propagating requested dimensions to be positioned.

ramin

This is an anchored note

○positive ○negative
◉neutral ○informative

(atena)

(samaneh)

Behaviors can be combined in arbitrary groups

Comment

This is a comment on highlight

custom browser for every one. Common aspects of document functionality can be shared, so that, for example, the same behavior that handles multipage support for scanned paper documents also provides such support for DVI and PDF; similarly, the behaviors that support fine-grain annotation of HTML also support identical annotation on scanned paper, UNIX

format viewers, such as the Ghostscript [5] for PostScript, recapitulate a large and

Ghostscript [5] for PostScript, xpdf [12] for PDF,

Citation of the document (URI):

http://URI.for.citation-note

Fig. 4.2. Various types of content annotations in the Viewer

After one or more annotations are added to the document, user can save them to the private or public repository. Selected annotations can be copied from the public repository to private at anytime, and vice versa (Figure 4.1(9)). User can delete the annotations that she owns, if a reply is not yet posted on them (Figure 4.1(10)).

For those notes and anchored notes that are published to public, a content snapshot nearest to the physical location of the note or surrounding the anchor is extracted and accompanies the annotation to the server.

4.1.3 Tagging, Rating, and Citation Support

Tags can be attached to the document by creating a new tag note and writing as many tag phrases that the user wishes. Tag note is visually similar to a note, but differs in the color and subject. Tag phrases are separated by new line. User can see a list of all tags assigned to the publication which is affected by the type of relevant publication copies that she has chosen for retrieving annotations, e.g. based on document digest.

User can rate the publication in the scale of 1 to 5 for the five optional given criteria (refer to Section 3.1.3.3). If she had already rated the same publication, her previous rating is overwritten by the new one.

The content extractor module extracts citation information from the publication as much as possible, which is dependent to the properties of the publication: PDF file format with recognizable characters and sticking to standard paper presentation styles (discussed in Section 3.2.1). The citation information extracted from the publication can be viewed, corrected, and updated by the user.

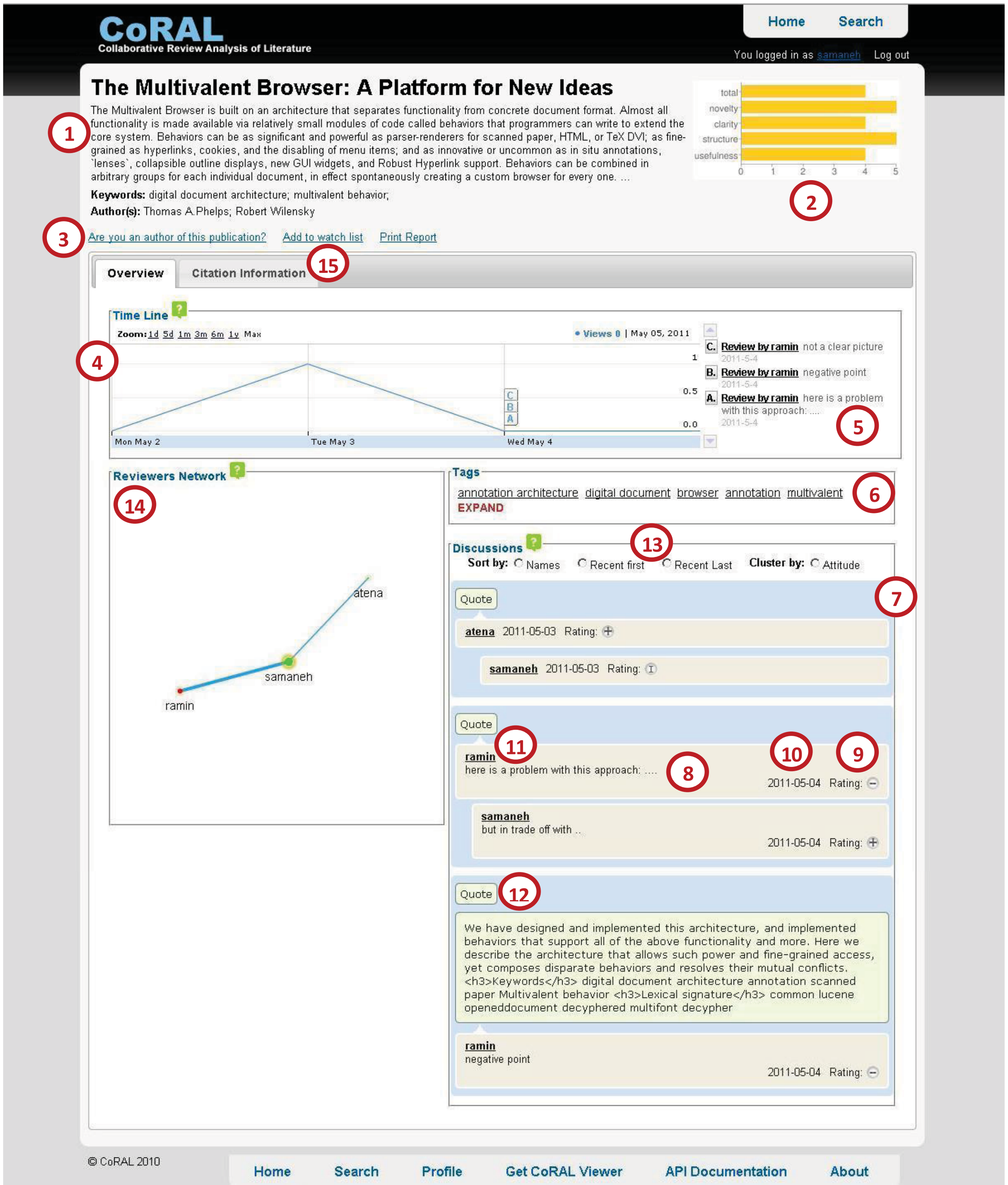
4.2 Usage of the CoRAL Reporter

The CoRAL Reporter is a web-based application that visualizes the reviews and reviewing activities in the CoRAL system. Since it also eases the access to various scientific publications and the reviewer profiles, it is also a good place to perform manipulating tasks such as changing profile settings and following people. The Reporter generates reports from two main perspectives: publication view and reviewer view. It also has a light view on field or tag reports. A basic search component is added to the system to ease access to these views.

4.2.1 Publication-centric View

In CoRAL, the reviews are collected in a publication-centric approach. So, accessing the reviews through the publication is the best anticipated way. For every reviewed scientific publication, identified by citation information, a publication-centric summary of reviews is generated in the Reporter. This summary consists of various small components, each concentrating on one or a few aspects of the activities. Figure 4.3 illustrates the publication-centric view in which its components are identifiable. Those components are described in the following. This figure is showing the main section which is the 'overview' tab in more details, where the hidden sections are simply understood by the descriptions that comes in the following.

Publication introduction: The publication view starts with a summary of the scientific publication (Figure 4.3(1)). This section includes the title, abstract, authors, and keywords of the publication as they are provided in the citation information.



The Multivalent Browser: A Platform for New Ideas

The Multivalent Browser is built on an architecture that separates functionality from concrete document format. Almost all functionality is made available via relatively small modules of code called behaviors that programmers can write to extend the core system. Behaviors can be as significant and powerful as parser-renderers for scanned paper, HTML, or TeX DVI; as fine-grained as hyperlinks, cookies, and the disabling of menu items; and as innovative or uncommon as in situ annotations, `lenses`, collapsible outline displays, new GUI widgets, and Robust Hyperlink support. Behaviors can be combined in arbitrary groups for each individual document, in effect spontaneously creating a custom browser for every one. ...

Keywords: digital document architecture; multivalent behavior;

Author(s): Thomas A.Phelps; Robert Wilensky

C. **Review by ramin** not a clear picture
2011-5-4

B. **Review by ramin** negative point
2011-5-4

A. **Review by ramin** here is a problem with this approach: ....
2011-5-4

atena 2011-05-03 Rating: ⊕

samaneh 2011-05-03 Rating: Ⓘ

ramin
here is a problem with this approach: ....
2011-05-04 Rating: ⊖

samaneh
but in trade off with ..
2011-05-04 Rating: ⊕

We have designed and implemented this architecture, and implemented behaviors that support all of the above functionality and more. Here we describe the architecture that allows such power and fine-grained access, yet composes disparate behaviors and resolves their mutual conflicts. <h3>Keywords</h3> digital document architecture annotation scanned paper Multivalent behavior <h3>Lexical signature</h3> common lucene openeddocument decyphered multifont decypher

ramin
negative point
2011-05-04 Rating: ⊖

Fig. 4.3. Publication-centric view and its components

Rating chart: A rating chart can be found beside the summary of publication (Figure 4.3(2)). This chart shows the aggregated reviewer ratings for the five rating criteria. Being the simplest piece of reviews, ratings come at the top with a quick to grasp visualization.

Tool bar: The tool bar is a line of hyperlinks consisting of a set of actions that user may wish to perform regarding the publication (Figure 4.3(3)). Using this tool bar, the author of this publication can claim authorship and link this publication to her own profile. Another option is adding the publication to the user's watch list to easily find the interesting publications later. If an author has provided the full text for the publication, a download link appears in the tool bar for interested readers. The last option is to print a report of the reviews. This option is currently designed as a personal report, which includes the logged-in user's comments and discussions that she has started on this publication. The tool bar is open for future extensions to add more functionality.

Time line: Every scientific publication attracts different levels of attention during years. A publication may become popular for some time, but lose its popularity after a while. Some papers though, remain reasonably popular for a very long time. Looking at the time line of reviewing activities is a great way to visually evaluate the effectiveness or popularity of a publication over time. The time line widget in the Reporter aims to provide this interesting view (Figure 4.3(4)). The time line is built by tracking the date of reviews and showing at what times the highest amounts of reviewing activities have occurred or when the reviewers have been less attracted to this publication. Being built based on the reviews, the time line itself becomes a new review from a higher level perspective.

Over the line of the time line, a flag is placed for each review which eases finding the reviews made at a specific time. A column containing the review summaries is located at the right side of the time line (Figure 4.3(5)). This column includes the reviewer username and the few first words of each comment.

Tag cloud: The tag cloud shows the tags assigned by reviewers along with visualizing the relative frequencies of the tags (Figure 4.3(6)). The default tag cloud only shows a few top frequent tags, but an interested user may access all the assigned tags by expanding the default tag cloud view. Each tag is linked to a related tag report (refer to Section 4.2.3).

Discussion threads: This component contains the information carried by the notes, anchored notes, and reply notes annotated on the publication (Figure 4.3(7)).

Notes and anchored notes are used to start commenting on the publication, and reply notes point to those comments and form a conversation. The conversations are visualized in this component by having the relevant note or anchored note at the beginning of a thread of comments. The conversation may have been continued by use of reply notes. This is illustrated by showing the replies coming after and indented relative to the comment to which they point. Each type of note in a conversation contains a body or the comment (Figure 4.3(8)), the reviewer's attitude (Figure 4.3(9)), date (Figure 4.3(10)), and the reviewer's username (Figure 4.3(11)). A conversation also has a snapshot of the content about which it talks, which is carried by the first comment in the thread. This snapshot can be viewed at top of every conversation by expanding the 'quote' option (Figure 4.3(12)).

The conversations can be sorted by the date or name of the reviewers according to their root comment. Grouping conversations is possible based on the attitude of the first commenter of each thread (Figure 4.3(13)). The reviewer's profile is easily accessible by clicking on the username in the comment.

Interactions network: A network of interactions is provided at the left side of the discussions component (Figure 4.3(14)). This network visualizes various information about the reviewers of this publication and their interactions. The interactions network consists of nodes and edges. Each node represents a reviewer involved in reviewing this publication. The relative size of the nodes indicate the relative amount of activity of the reviewers. The color of each node talks about the overall attitude of this person, green being all positive, red being all negative, and the colors in between talking about average attitudes which is a mix of green and red attitudes. A node may have a yellow halo. Existence of a halo shows that this reviewer has been recently active in reviewing. The bigger halos indicate more activities in recent days. The node properties are taken from the global activities of the reviewers, giving the report a hint of how these reviewers act in the overall scale. The edges between nodes indicate that these two reviewers have participated in the same conversation in the discussions component.

Interactions between report components: The interactions network contains common parts with the discussion component, while they differ in how this information is represented. In the interactions network, users can see who has interactions with whom, which is also found by looking at the conversation threads. Having this common element, the interactions network can be used for an interactive experience in conjunction with the discussions component. Choosing a reviewer node in the network leads to highlighting her interaction in both the network and the discussions component. This action causes the discussions involving this person to expand, while the unrelated discussions are minimized.

There is also a common element between the time line and the discussions component: the comments. As described earlier, the time line comes with flags and summaries of comments. Clicking on a flag or a comment summary leads to highlighting the related comment in the discussions component, so that the user can access the full information about that comment. In addition to highlighting, the page scrolls to the location where the comment is displayed for easy discovery.

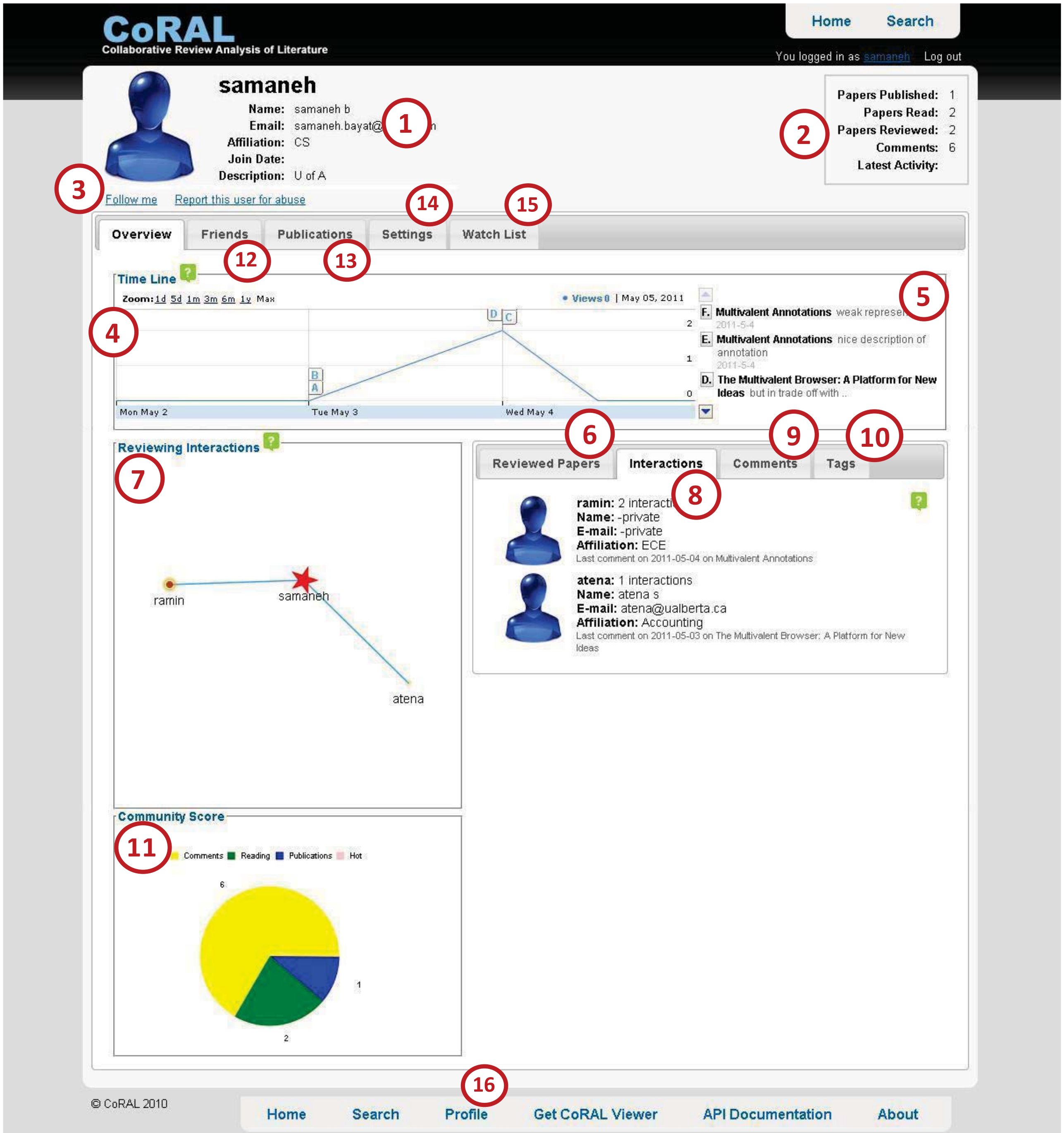
Citation information: In the citation information tab (Figure 4.3(15)), user can view or update the citation information of this publication.

4.2.2 Reviewer-centric View

Every reviewer in the CoRAL has a profile which includes her reviewing history. This profile can be viewed as a reviewer-centric report. Similar to the publication-centric view, the reviewer-centric view is composed of several components each concentrating on a few aspects of the reviewing activities. In addition to the reviewing activities, the reviewer profile is a good place for changing settings and keeping track of their own interests such as managing watch list. Figure 4.4 illustrates the reviewer view and its components. A description for each component comes in the following.

Reviewer introduction: Similar to the publication view, the reviewer view starts with a summary of the reviewer's available information (Figure 4.4(1)). This section includes username, name, affiliation, email, join date, and a brief description provided by the reviewer. The availability of any of these information, except the username, is dependent to the reviewer's privacy settings. In the future, adding a profile picture can be added to the profile features.

Statistics chart: A statistics chart can be found at the top right corner of this view (Figure 4.4(2)). This chart is a quick numerical view on the reviewer's activities including the number of publications she has claimed, number of papers



samaneh

Name: samaneh b
Email: samaneh.bayat@ n
Affiliation: CS
Join Date:
Description: U of A

Papers Published: 1
Papers Read: 2
Papers Reviewed: 2
Comments: 6
Latest Activity:

| May 05, 2011

F. Multivalent Annotations weak represen
2011-5-4

E. Multivalent Annotations nice description of annotation
2011-5-4

D. The Multivalent Browser: A Platform for New Ideas but in trade off with ..

ramin: 2 interacti
Name: -private
E-mail: -private
Affiliation: ECE
Last comment on 2011-05-04 on Multivalent Annotations

atena: 1 interactions
Name: atena s
E-mail: atena@ualberta.ca
Affiliation: Accounting
Last comment on 2011-05-03 on The Multivalent Browser: A Platform for New Ideas

Fig. 4.4. Reviewer-centric view and its components

read or reviewed, number of comments, and latest activity date.

Tool bar: This tool bar contains links to relevant actions that user can take with respect to the viewed profile (Figure 4.4(3)). The tool bar currently contains these two options: follow me, and report for abuse. The user can indicate interest in this reviewer's work by choosing the 'follow me' option in the tool bar. This way, the current viewer will be added to the user's list of followed people which can be found in the friends component. Clicking on the report for abuse option will take the user to a form to fill. Using this form, users can report that they have observed an abusive behavior or comment from the current reviewer's side. They should also indicate a reason for this claim. An email containing the report data will be sent to the CoRAL's administrative email address that will be reviewed by the administrator. The administrator then can remove users or delete comments. The functionality of the administrator is implemented in the Reporter with limited features.

Time line: The reviewer view contains a time line that shows the reviewer's activity over time (Figure 4.4(4)). This component can be reviewed to find the activeness of this user and how it changes by time. A column containing the reviewed publications and the reviewer's comments comes to the right side of the time line to give a quick view of what comment were made at what time (Figure 4.4(5)).

Reviewed papers: A list of the papers that have been reviewed by this reviewer comes in this component, ordered by date (Figure 4.4(6)). The related publication views are accessible each by a click.

Interactions network and interactions tab: Similar to the publication-centric view, the interactions network visualizes the interactions where nodes are reviewers and edges indicate common conversations (Figure 4.4(7)). But, the network in the reviewer view illustrates the interactions that current reviewer has with other reviewers among all the publications. A new type of node can be found in this network which represents the reviewer for which the network is drawn: a red star node. For clarity and concentration, only the interactions between the star node and other nodes are shown. The properties of the regular nodes are exactly the same as the network in the publication view.

A list of collaborators that this reviewer has interacted with, comes in the interactions tab (Figure 4.4(8)). In this component, a summary of the collaborators such as their name and affiliation is listed considering their privacy settings. A link to each person's profile is also provided.

Comments: A list of the most recent comments of this reviewer can be found in the comments component (Figure 4.4(9)). Each comment points to the publication view on which the comment is made. The attitude and date of comments are also shown in the comments list.

Tag clouds: Every reviewer has two tag clouds: one representing the frequent tags attached to the papers reviewed by this person, and the other representing the frequency of tags that this reviewer has assigned to any publication (Figure 4.4(10)). Both of these tag clouds are shown in the tags component showing the first few top ones by default. Each of them can be expanded to see all tags in the category. Every tag links to the tag view explained later in Section 4.2.3.

Community score: The community score is an alternative representation of the statistics chart that is illustrated as a pie chart (Figure 4.4(11)). The pie charts are easier to quickly understand than numerical representations. Using the pie chart, users can find out which activity is more interesting to this reviewer: commenting, reading, or adding publications. The concept of community score is introduced to encourage researchers to do more reviews by visualizing their contributions.

Friends: If the visibility of the friends list is set to public, the friends tab (Figure 4.4(12)) shows two lists: the users who are followed by this reviewer, and the users who follow this reviewer. A brief summary of every user with a link to her profile is provided in each list.

Publications: In this component (Figure 4.4(13)), the list of publications that this reviewer has claimed to be an author of them is shown. For every added publication, a publication-centric view is available by a click. If the user is viewing her own profile, she can also add new publications to her list. When adding a publication, user can include citation information and optionally upload the full text of the publication which will become accessible to all CoRAL members.

Settings: If the user is viewing her own profile, the settings tab becomes available (Figure 4.4(14)). In this tab, user can change her information or privacy settings. This includes updating: password, email address and its visibility, affiliation and its visibility, description and its visibility, visibility of friends list, visibility of publications list, visibility of the watch list, and visibility of name.

Watch list: The watch list contains links to the publication views of the publications that this reviewer is interested in (Figure 4.4(15)). This list acts as a convenient way for the reviewer to access the reviews on publications that she likes. Users can find out about each other's interests or find interesting publications by looking at the profile of the reviewers they like. The watch list can become a private list by changing the privacy settings.

4.2.3 Tags View, Global View, and Search

The tags view is a light component used to connect literature based on the common concepts that the reviewers identify. The current tag view is a list of papers that are tagged with a given phrase. The publication-centric report of each paper in the list is accessible by a click.

The global view can be found in the CoRAL's home page. The global view consists of summary of usage in the CoRAL system. This summary comes in the form of reporting the top publications added to CoRAL based on these aspects: most read papers, most annotated papers, and highest rated papers. Many other kinds of global reports can be added to the system, such as reports that involve reviewers or attitudes.

In order to ease finding the publications and the reviewers, a light search component is provided. The current version of the search component supports basic search by keywords. Two search categories are specified: one searches for publications, the other one searches for reviewers. The search component can be improved by adding advanced search options. For the publication search, advanced search should include searching based on citation information, tags, and the attention that the publication has received in the CoRAL community. In addition, a goal-based search can be used to take advantage of the five rating criteria used in rating the publications (discussed in Section 5.4). The possible advanced search options for the reviewer search include searching in their information such as name and affiliation, in addition to considering CoRAL activities such as the assigned tags or amount of commenting activity. In the case of searching based on the user information, the privacy settings should also be considered.

4.3 Storyboards

In the previous sections, we talked about the features of the CoRAL components and how they are used individually. In this section, we describe use-case scenarios and illustrate storyboards about how users can take advantage of these tools with respect to the CoRAL goals that was discussed in Chapter 1.

Figures 4.5, 4.6, and 4.7 illustrate storyboards for a basic flow of usage in the Viewer. In describing the usages, we go back to the scenario of two researchers, Alice and Bob, reviewing the same content (refer to Chapter 1). Alice reads a paper in the Viewer and places a comment on it (Figure 4.5a). At this time, she may also specify the general attitude of her comment to let the CoRAL system take advantage of this data. She may decide to keep the comment in her private repository or publish it to the public (Figure 4.5b). In this scenario, she publishes her comment to the CoRAL public repository. She can do the same thing using other forms of annotations such as highlighting, anchored notes, and tags.

Bob reads the same paper using another physical file location. He can decide how he wishes to receive annotations from other copies of the same publication by changing his preferences (Figure 4.6a). In this case, he has chosen to view his own annotations (same URL), annotations from documents with same content and not necessarily same format, and same bibliographic information. Since both Alice and Bob are viewing a PDF version of this publication, both later options lead to Alice's comment to be shared with Bob. Bob notices content annotations published by other reviewers in both the document content and in the annotation panel (Figure 4.6b). Alice's comment contains a point that Bob has an answer for it, so he replies to her comment and posts his reply to public (Figure 4.6c). This way, the researchers communicate and take advantage of each other's reviews without sharing a document.

Users can tag, rate, and provide citation information for the reviewed publication using menu options (Figure 4.7a). Similar to sharing comments, users can attach tags to the publication using a note-like interface in which multiple tags are separated by new lines (Figure 4.7b). In addition, the publication and the reviewer are both linked to their respective pages in the CoRAL Reporter (Figure 4.7c).

The annotations can be retrieved from a copy of the publication with a different format, as long as the textual content or the citation information match. Figure 4.8 shows the same paper that Alice and Bob are reviewing but in the text format. Since the formatting data is lost, the user needs to provide the tool with primary citation information (title and authors) in order to be able to take advantage of bibliographic identification method (refer to Section 3.1.4). The annotations and their contents are successfully retrieved. However, the placement of annotations in the physical locations may not match with the PDF version because the layout of the content has changed, e.g. pages and columns are lost. As a future improvement for format-independent placement of annotations, we can benefit from the context-awareness of the annotations by recognizing the textual content on which the annotations were made.

When Alice reviews the paper publicly, her reviewing activity becomes available in the Reporter. As shown in Figure 4.7c, she may go to her profile or the publication-report view in the Reporter directly from the Viewer. Another researcher, John, who is interested in the papers talking about 'multivalent documents' can find this paper or other related publications using the Reporter's search option. Figure 4.9a shows the search view in the Reporter. Note that in the current version, only the title search works. From the search results, he can browse the publication report for every publication in the search results (Figure 4.3). He may also reach to these papers (Figure 4.9b) by clicking on a relevant tag (e.g. 'multivalent') in a tag cloud found in publication or reviewer views. In addition, he can find about the most reviewed and read papers in a brief global report brought in the CoRAL's homepage (Figure 4.9c).

In the publication view, the reviews and the activities of the users who have reviewed this publication are visualized. As Figure 4.10a shows, Alice and Bob's conversation can be found along with a piece of content that they are talking about. Not having the full text of the publication, this piece of content helps the observer to find the conversation more meaningful. The reviewing interactions is shown as in Figure 4.10b where information about interactions on this publication, reviewers' amount of global activity, recency of their activities, and their general attitudes can be inferred (as discussed in Section 4.2.1). The network shows overall attitude of the reviewers throughout all the publications that they have reviewed. The discussion thread mentions the reviewer's attitude related to every specific comment on the current publication. The bigger the nodes means the respective reviewer is more active in CoRAL. For example, Alice has made only one comment which is negative and her node is shown as a small red circle. But, Ramin has a negative comment on this publication (Figure 4.10c) along with a big green node shown in the interactions network (Figure 4.10b). This indicates that although he has made a negative comment on this paper, his overall attitude in other places is positive. The conversation between Alice and Bob is illustrated by a line between their respective nodes. This information helps other researchers, namely John, to get to know other reviewers and benefit from their comments.

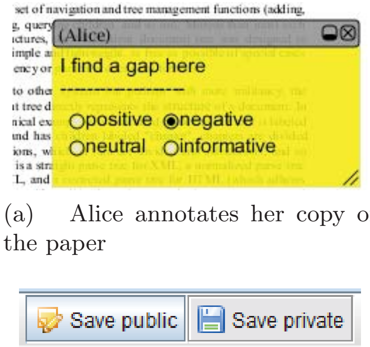
A tag cloud (Figure 4.11a) shows the top frequent tag phrases assigned to this publication, each linking to a tag view (Figure 4.9b). This tag cloud can be expanded to view all assigned tags for a full report. John may use this tag cloud to access reports about related publications tagged by 'multivalent' as an indicator of a common aspect. Rating chart is also helpful to give the observer a quick glance on reviewers average satisfaction (Figure 4.11b).

Looking at the time line of reviewing activities (Figure 4.11c), John finds out when and by whom this paper has attracted more attention. The peek of reviewing activities may be in the last month or in a few years ago, giving him an overview of when this topic has been popular. Alice and Bob's comments are the last ones seen in this time line. They can find out about most recent interests on the same paper that they are currently reviewing which may lead to discussions and collaborations with those users.

The user's own profile is accessible through both Viewer links (Figure 4.7c) and Reporter's 'profile' link (Figure 4.4(16)). A time line similar to the one found in every publication report (Figure 4.11c) is also created for every reviewer such as Alice. This time line shows Alice's reviewing activities through time pointing to the publications that she has reviewed. Alice's summary of interactions can be found in her profile (Figure 4.12a). As the profile shows, bob is the only reviewer that Alice has had discussion with. Bob's personal information, e.g. his email address, are not shown in Alice's profile due to his privacy settings. Bob has managed his privacy settings by an interface shown in Figure 4.13a that he can reach through his profile.

Alice notices that she and Bob may have common interests by receiving a comment from Bob on a paper of interest and/or by reviewing Bob's profile. She can take advantage of Bob's tag clouds (Figure 4.12b) as a way to find links to papers that can be of her interest too.

Alice can add her publications to the CoRAL system by entering bibliographic information and optionally uploading the full text (Figure 4.13b). If the publication is already existing due to being reviewed in the community, it will be linked to the corresponding publication report. If Bob views Alice's publications tab, he can see the list of her publications if the privacy rules allow. The form used to add publications is only accessible through the user's own profile.



(Alice)

I find a gap here

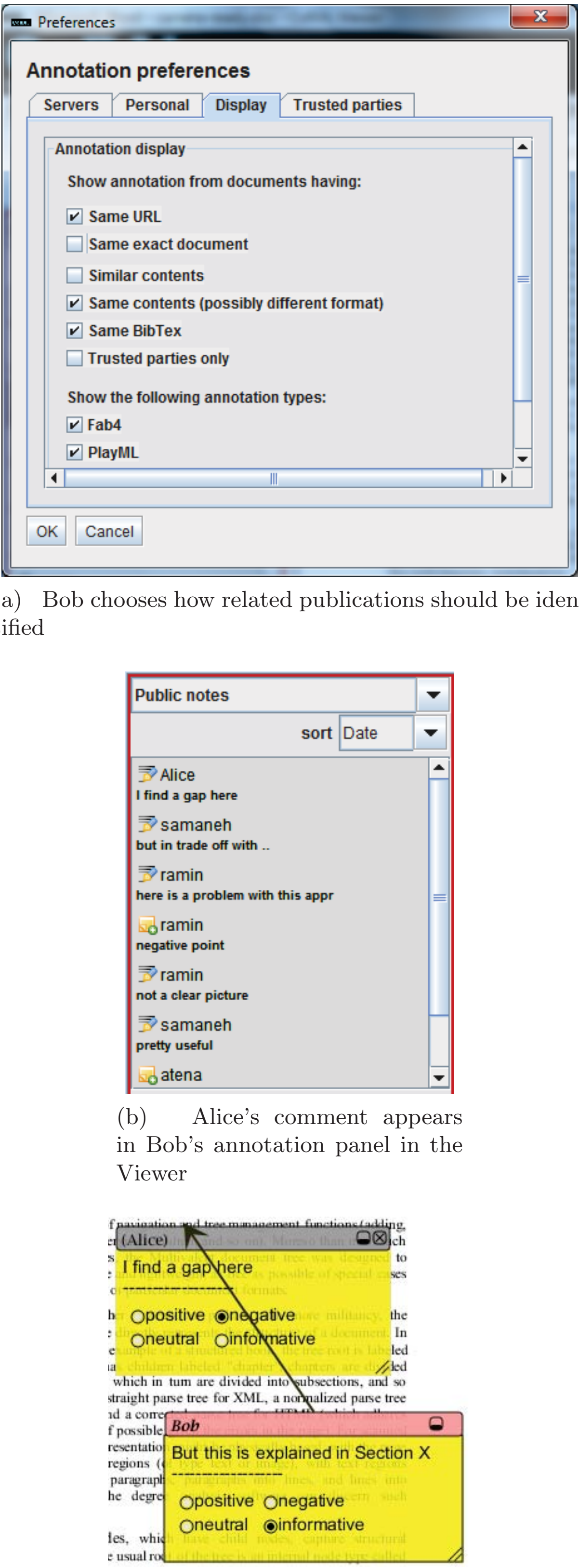
○positive ●negative

○neutral ○informative

(a) Alice annotates her copy of the paper

(b) Alice posts her comments to public

Fig. 4.5. Viewer storyboard: commenting and sharing



Annotation preferences

Annotation display

Show annotation from documents having:

☑ Same URL
□ Same exact document
□ Similar contents
☑ Same contents (possibly different format)
☑ Same BibTex
□ Trusted parties only

Show the following annotation types:

☑ Fab4
☑ PlayML

(a) Bob chooses how related publications should be identified

Public notes

sort Date

Alice
I find a gap here

samaneh
but in trade off with ..

ramin
here is a problem with this appr

ramin
negative point

ramin
not a clear picture

samaneh
pretty useful

atena

(b) Alice's comment appears in Bob's annotation panel in the Viewer

(Alice)
I find a gap here

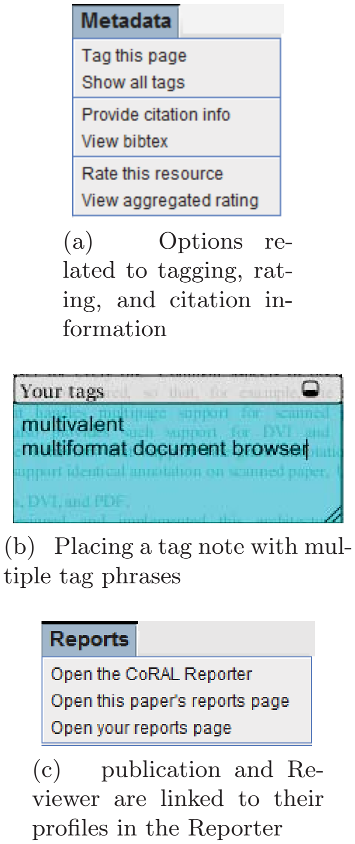
○positive ●negative
○neutral ○informative

Bob
But this is explained in Section X

○positive ○negative
○neutral ●informative

(c) Bob replies to Alice's comment

Fig. 4.6. Viewer storyboard: retrieving related content annotations



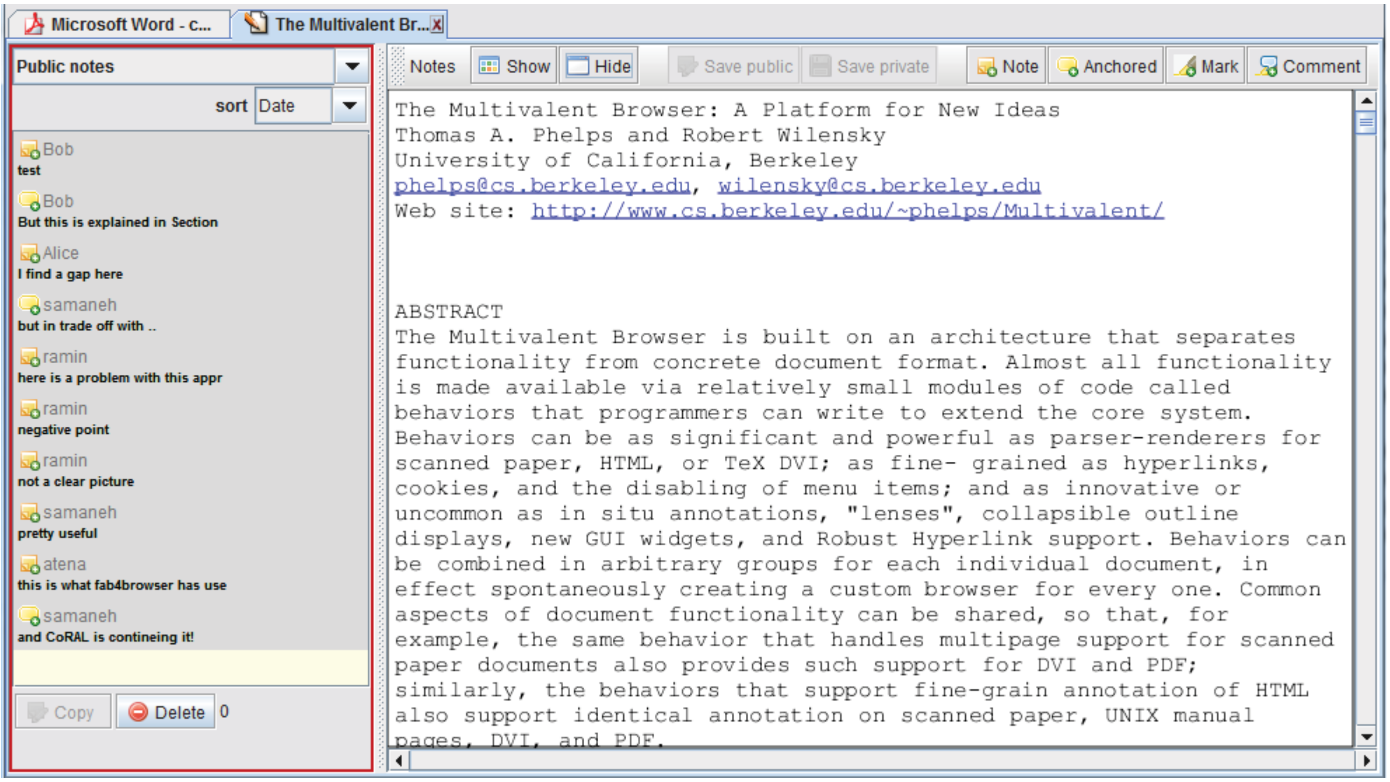
multivalent
multiformat document browser

(a) Options related to tagging, rating, and citation information

(b) Placing a tag note with multiple tag phrases

(c) publication and Reviewer are linked to their profiles in the Reporter

Fig. 4.7. Viewer storyboard: examples of other basic features



Bob
test

Bob
But this is explained in Section

Alice
I find a gap here

samaneh
but in trade off with ..

ramin
here is a problem with this appr

ramin
negative point

ramin
not a clear picture

samaneh
pretty useful

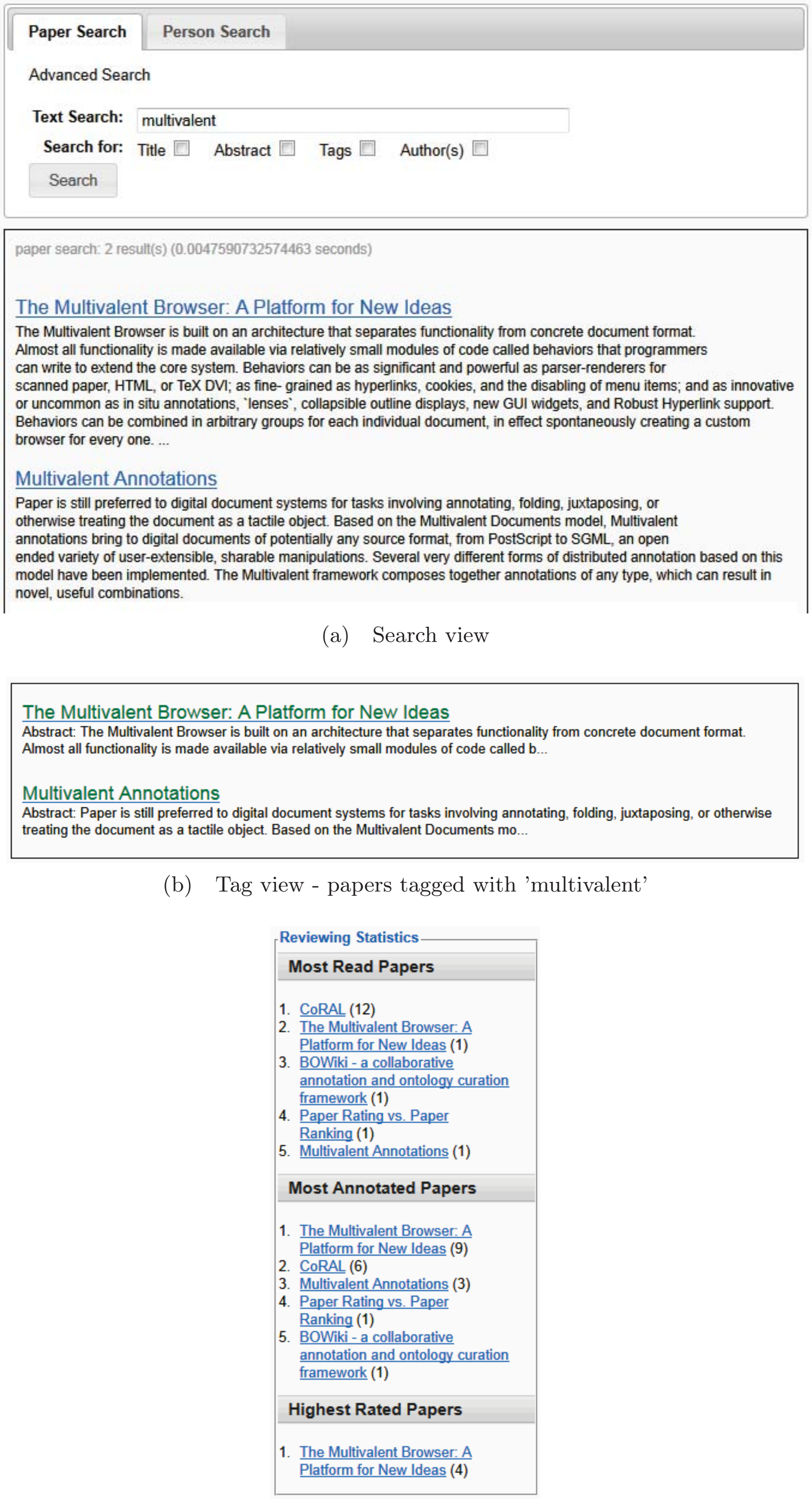
atena
this is what fab4browser has use

samaneh
and CoRAL is contineing it!

The Multivalent Browser: A Platform for New Ideas
Thomas A. Phelps and Robert Wilensky
University of California, Berkeley
phelps@cs.berkeley.edu, wilensky@cs.berkeley.edu
Web site: http://www.cs.berkeley.edu/~phelps/Multivalent/

ABSTRACT
The Multivalent Browser is built on an architecture that separates functionality from concrete document format. Almost all functionality is made available via relatively small modules of code called behaviors that programmers can write to extend the core system. Behaviors can be as significant and powerful as parser-renderers for scanned paper, HTML, or TeX DVI; as fine- grained as hyperlinks, cookies, and the disabling of menu items; and as innovative or uncommon as in situ annotations, "lenses", collapsible outline displays, new GUI widgets, and Robust Hyperlink support. Behaviors can be combined in arbitrary groups for each individual document, in effect spontaneously creating a custom browser for every one. Common aspects of document functionality can be shared, so that, for example, the same behavior that handles multipage support for scanned paper documents also provides such support for DVI and PDF; similarly, the behaviors that support fine-grain annotation of HTML also support identical annotation on scanned paper, UNIX manual pages, DVI, and PDF.

Fig. 4.8. Viewer storyboard: retrieving annotations in a differently formatted paper



Paper Search | Person Search

Advanced Search

Text Search: multivalent

Search for: Title □ Abstract □ Tags □ Author(s) □

Search

paper search: 2 result(s) (0.0047590732574463 seconds)

The Multivalent Browser: A Platform for New Ideas

The Multivalent Browser is built on an architecture that separates functionality from concrete document format. Almost all functionality is made available via relatively small modules of code called behaviors that programmers can write to extend the core system. Behaviors can be as significant and powerful as parser-renderers for scanned paper, HTML, or TeX DVI; as fine- grained as hyperlinks, cookies, and the disabling of menu items; and as innovative or uncommon as in situ annotations, `lenses`, collapsible outline displays, new GUI widgets, and Robust Hyperlink support. Behaviors can be combined in arbitrary groups for each individual document, in effect spontaneously creating a custom browser for every one. ...

Multivalent Annotations

Paper is still preferred to digital document systems for tasks involving annotating, folding, juxtaposing, or otherwise treating the document as a tactile object. Based on the Multivalent Documents model, Multivalent annotations bring to digital documents of potentially any source format, from PostScript to SGML, an open ended variety of user-extensible, sharable manipulations. Several very different forms of distributed annotation based on this model have been implemented. The Multivalent framework composes together annotations of any type, which can result in novel, useful combinations.

(a) Search view

The Multivalent Browser: A Platform for New Ideas

Abstract: The Multivalent Browser is built on an architecture that separates functionality from concrete document format. Almost all functionality is made available via relatively small modules of code called b...

Multivalent Annotations

Abstract: Paper is still preferred to digital document systems for tasks involving annotating, folding, juxtaposing, or otherwise treating the document as a tactile object. Based on the Multivalent Documents mo...

(b) Tag view - papers tagged with 'multivalent'

Reviewing Statistics

Most Read Papers

1. CoRAL (12)
2. The Multivalent Browser: A Platform for New Ideas (1)
3. BOWiki - a collaborative annotation and ontology curation framework (1)
4. Paper Rating vs. Paper Ranking (1)
5. Multivalent Annotations (1)

Most Annotated Papers

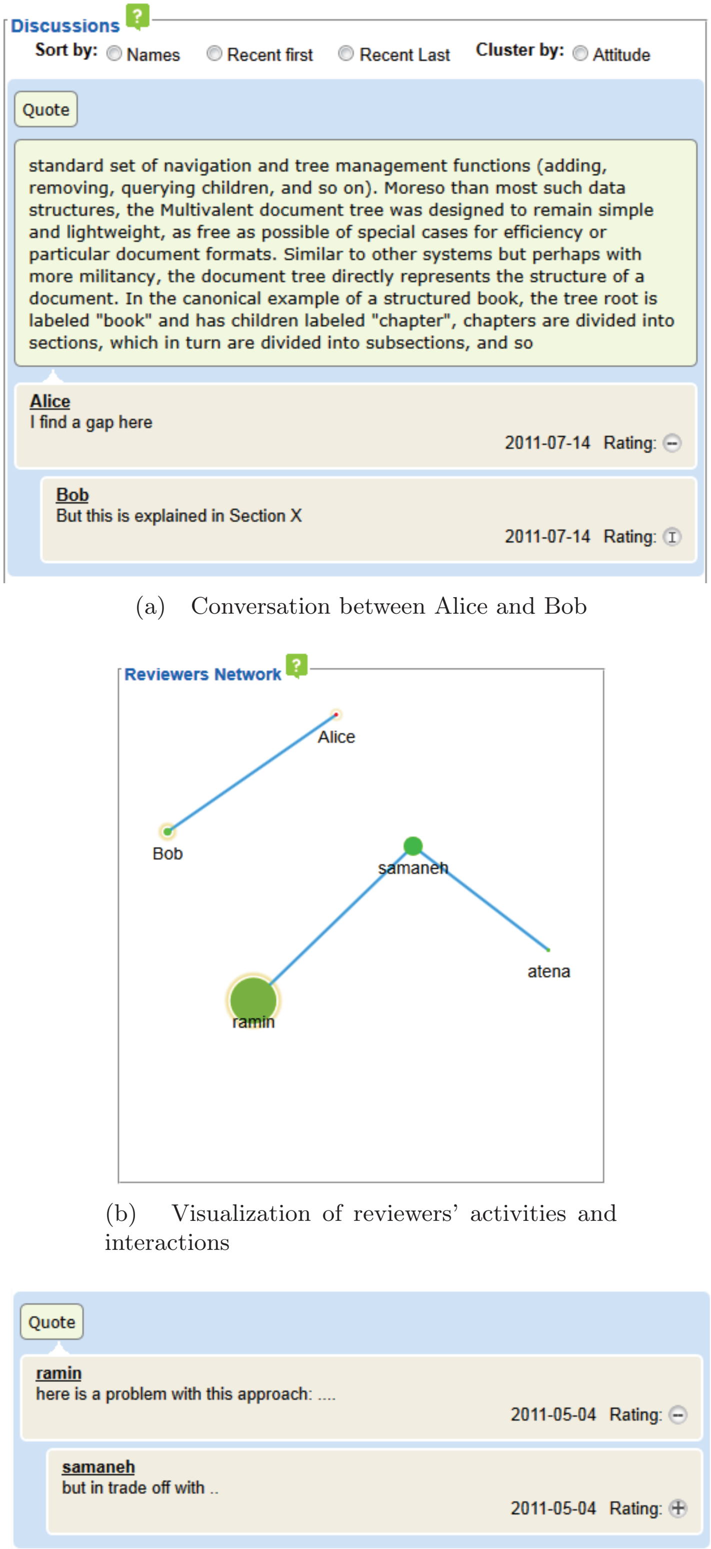
1. The Multivalent Browser: A Platform for New Ideas (9)
2. CoRAL (6)
3. Multivalent Annotations (3)
4. Paper Rating vs. Paper Ranking (1)
5. BOWiki - a collaborative annotation and ontology curation framework (1)

Highest Rated Papers

1. The Multivalent Browser: A Platform for New Ideas (4)

(c) Global reports

Fig. 4.9. Reporter storyboard: search, tag view, and global reports



Discussions

Sort by: ○ Names ○ Recent first ○ Recent Last Cluster by: ○ Attitude

Quote

standard set of navigation and tree management functions (adding, removing, querying children, and so on). Moreso than most such data structures, the Multivalent document tree was designed to remain simple and lightweight, as free as possible of special cases for efficiency or particular document formats. Similar to other systems but perhaps with more militancy, the document tree directly represents the structure of a document. In the canonical example of a structured book, the tree root is labeled "book" and has children labeled "chapter", chapters are divided into sections, which in turn are divided into subsections, and so

Alice
I find a gap here
2011-07-14 Rating: ⊖

Bob
But this is explained in Section X
2011-07-14 Rating: Ⓘ

(a) Conversation between Alice and Bob

(b) Visualization of reviewers' activities and interactions

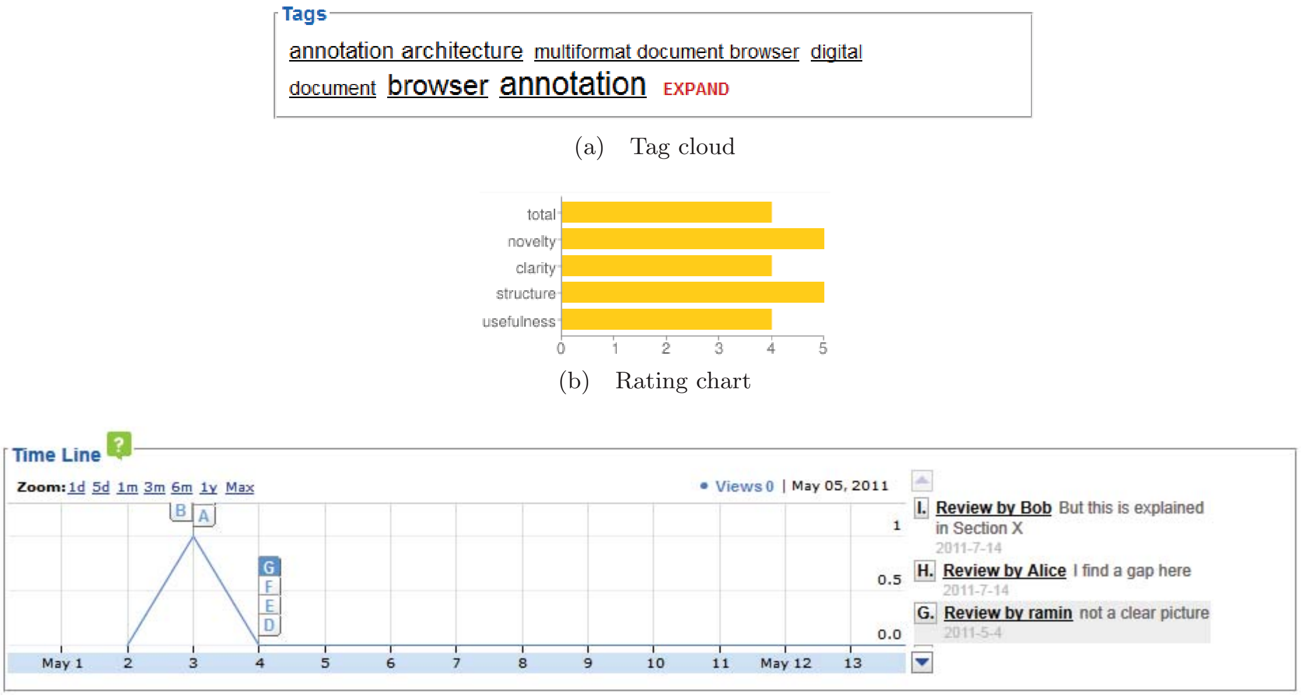
Quote

ramin
here is a problem with this approach: ....
2011-05-04 Rating: ⊖

samaneh
but in trade off with ..
2011-05-04 Rating: ⊕

(c) A negative comment by an overall positive commenter

Fig. 4.10. Reporter storyboard: discussions, interactions, and attitudes in the publication-centric report



I. Review by Bob But this is explained in Section X
2011-7-14

H. Review by Alice I find a gap here
2011-7-14

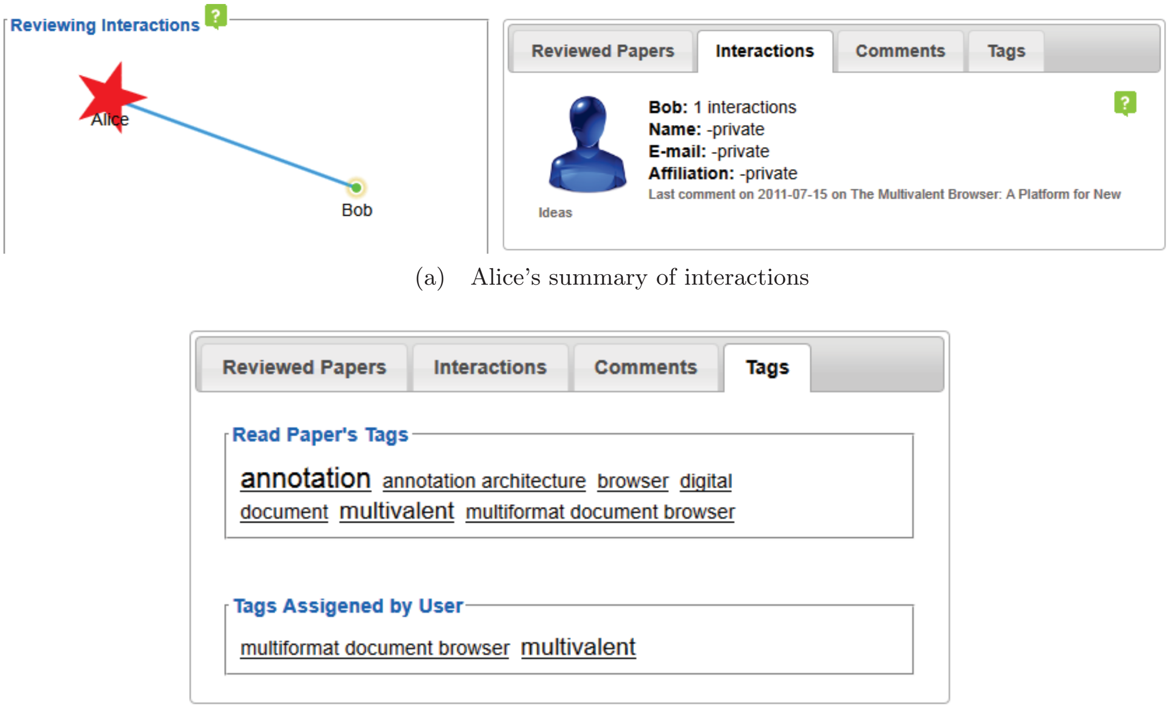
G. Review by ramin not a clear picture
2011-5-4

(a) Tag cloud

(b) Rating chart

(c) Time line of reviewing activities on this publication

Fig. 4.11. Reporter storyboard: tag cloud, rating, and time line in the publication report

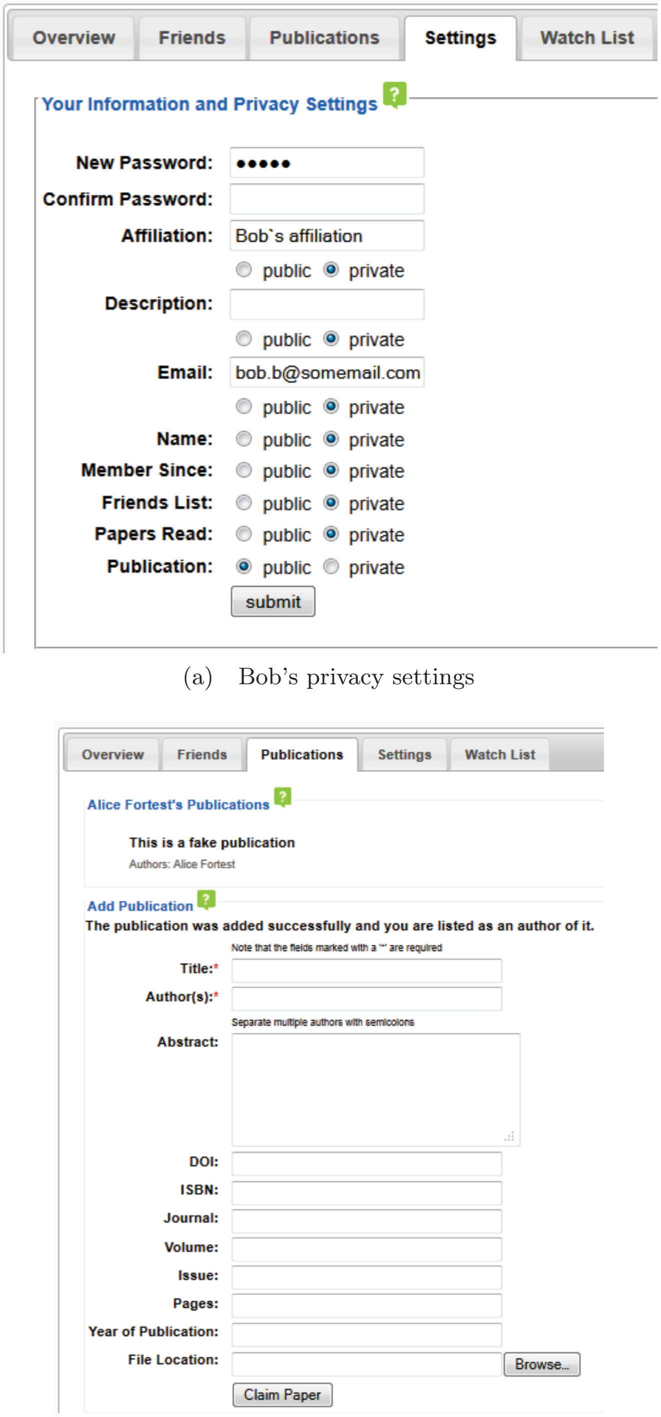


Bob: 1 interactions
Name: -private
E-mail: -private
Affiliation: -private
Last comment on 2011-07-15 on The Multivalent Browser: A Platform for New Ideas

(a) Alice's summary of interactions

(b) Bob's tag clouds

Fig. 4.12. Reporter storyboard: interactions and tag clouds in the reviewer reports



Your Information and Privacy Settings

New Password: ●●●●●

Confirm Password:

Affiliation: Bob`s affiliation

○ public ● private

Description:

○ public ● private

Email: bob.b@somemail.com

○ public ● private

Name: ○ public ● private

Member Since: ○ public ● private

Friends List: ○ public ● private

Papers Read: ○ public ● private

Publication: ● public ○ private

submit

(a) Bob’s privacy settings

Alice Fortest's Publications

This is a fake publication

Authors: Alice Fortest

Add Publication

The publication was added successfully and you are listed as an author of it.

Note that the fields marked with a "*" are required

Title:*

Author(s):*

Separate multiple authors with semicolons

Abstract:

DOI:

ISBN:

Journal:

Volume:

Issue:

Pages:

Year of Publication:

File Location: Browse...

Claim Paper

(b) Alice’s publication list management

Fig. 4.13. Reporter storyboard: privacy and publications management in reviewer profile

4.4 Summary

This chapter introduced the features of the two CoRAL components that involve user experience: the Reporter and the Viewer. The Viewer is a document browser and annotator that gives the ability to save annotations in the private or public repositories. The features of the Viewer were discussed in Section 4.1 in three categories: browser utilities, content annotations, and other annotation types. Browser utilities include: private or public annotation saving, choosing the document copy identification method, browsing various document formats, hiding or showing all or selected annotations, and automatic citation extraction. Content annotations are various types of annotations shown in Figure 4.2 that are related to the document content. Other annotation types are tags, ratings, and support for citation information.

The Reporter, discussed in Section 4.2, mainly consists of a publication-centric view and a reviewer-centric view. Both of these views consist of smaller components that each of them concentrates on visualizing or managing a few aspects related to a publication or reviewer. These views successfully managed to generate interesting visualized reports including time line of activities, tag clouds, interaction networks, discussions and comments, attitudes, and frequency of activities. In addition, there are features related to managing profiles and publication-related actions including following one's work, adding publications to watch list, printing personal report, reporting users for abuse, downloading full text, managing citation information, claiming authorship and adding publications, and changing privacy settings.

Tag view, global view, and basic search component are provided in addition to the aforementioned main views. The tag view lists summaries and links to the publications that correspond to a given tag phrase. The global view refers to the global statistics that are derived from the system usage in regard to the resouces.

Chapter 5

Features and Aspects

In the last two chapters, we discussed how users interact with CoRAL and how CoRAL is built in the direction of the requirements that was identified in Chapter 2.4. As we have already discussed, CoRAL's main objective is to support collaborative reviewing of the scientific literature. In addition to targeting its main goal, CoRAL supports other interesting aspects related to or resulting from collaborative reviewing. In this chapter, we briefly point to the benefits of CoRAL in context of its main goal and the related aspects.

5.1 Supporting Broader Literature Review

Supporting field review is an important objective of CoRAL. This objective is reached by several features that come with CoRAL. In-context commenting helps understanding and critiquing various aspects of a paper. Reviewers can also categorize their comments by indicating their attitude: whether it is a positive, negative, neutral, or informative comment or none. A report of comments on a paper includes content, attitude, date of review, and the reviewer of the comments. Another report can contain timeline of reviewing activities on a paper along with a social network showing researchers discussing the paper. One can also view aggregated ratings on a paper based on five different criteria. Such reports would give a comprehensive review of the paper from various perspectives. In addition, CoRAL shows small snapshots of the document that comments are made on them. Collection of such snapshots would give an overview of interesting or controversial pieces of the text without revealing the full context.

Tags can effectively group related literature considering various aspects of papers. Exploration of the literature review through the tags leads to broader field review with meaningful links between pieces of literature.

Collaboration is the primary way of collective thinking and networking is an effective way of gaining collaboration. CoRAL supports both collaborative thinking through discussion and explicit networking through following one's work. This is in direct favor of producing valid field reviews, since collective criticism is more trustworthy than individual review.

5.2 Introducing New Collaboration Links

Comments collected on papers are accessible by all reviewers irrespective of where the various copies of the paper, on which the comments were made, are located. Researchers can discuss about a paper and find a chance to become acquainted with other researchers with similar interests. Also, reviewers can see reports about the reviewing activities showing who said what in response to whom and with what attitude. In addition, CoRAL can provide general reports pointing to active people in a field or authors of papers that attract more attention. These links are usually outside the zone in which people know each other based on their workplaces or previous collaborations. Showing a community score for users of the system based on any reviewing or networking activity that they perform, is also another way of introducing possible interesting collaborators.

Along with the indirect networking activities that happen through commenting and tagging, CoRAL also provides explicit connections in the form of following each other's reviewing activities.

5.3 Statistics that Guide

The data collected in CoRAL is suitable for extraction of various kinds of interesting statistical reports about reviewing activities in the community of researchers. A possible statistical report can be achieved by listing the papers that attract more reviews, further organized in terms of the reviews' attitudes (i.e., positive or negative).

Another interesting report can be a list of papers that seem controversial with long discussions and contradictory number of positive and negative comments. Activities of the reviewers can also be reviewed: the most active reviewers, the most positive or negative reviewers.

Looking at the tags assigned by users can be another source for interesting statistics. Tags that are repeatedly used by a specific reviewer can be an identifier of her interests. Tags that are frequently assigned to many papers may be a result of interestingness of a topic in the community.

In addition, some statistics can be extracted to guide the review process or introduce review trends. For example, papers that are read frequently but have few comments could be shown to users with a request to review them.

Reaching a quick glance of interesting publications, authors, reviewers, and tags gives an insight about current research trends and important people in a field. As discussed in Section 2.4, Mendeley provides similar statistics over a huge user-set which practically introduces reputation metrics and may influence literature reading trends. These two kinds of statistics differ in the data they are referring to: reading data for Mendeley and reviewing data for CoRAL. Although reviewing data carries more information such as attitude and rating, the reading statistics can predict some of the reviewing statistics such as attraction of readers. Since Mendeley is providing such statistics in an extensive way, we have refrained from implementation of these statistics and we only have introduced them as the CoRAL data supports them.

5.4 Goal-based Search

CoRAL asks users to optionally rate publications based on five criteria that is generally important from a review perspective: clarity, novelty, structure, usefulness, and total quality. The rating statistics derived from these rating criteria is a constructive feedback for the authors and the reviewers. Moreover, gathering these ratings from various readers would greatly help in ranking publications that is useful to sort search results. Unlike rankings that are based on a general score, CoRAL has the potential to rank publications based on what a specific searcher is willing to gain. Suppose a researcher is looking for new research topics, ranking based on novelty could be much more important than clarity. Suppose someone is looking to define an industrial project, searching for more useful papers would give him better results. Suppose a student wants to learn about a topic, clarity would benefit her more. Based on these observations, the rating data collected in CoRAL API can be used to give users the option to give more weight to some of these criteria according to their goals. Tags can also accompany these criteria. The ranked list of papers under a tag (category) improves the quality of search for people (especially students) who are willing to find the significant papers in an area as their first search results.

In addition, degree of interestingness or controversy of papers to the CoRAL community can also be chosen to effect the ordering of search results. The rating, tagging, and statistical criteria can be added to basic search fields such as keywords and authors.

The data needed to support such a goal-based search is collected and accessible by the CoRAL API. Although the current implementation of the consumer application (the CoRAL Reporter) does not support the goal-based search, this is something that is encouraged and introduced by the CoRAL API. The goal-based search is a neat idea in further employment of collaborative reviews in a practical and useful way to help researchers.

5.5 A Platform for Analysis of Reviewing Behaviors

Assuming that CoRAL is broadly adopted and used as envisioned, it will end up collecting a rich behavioral dataset that can be further analyzed to find interesting patterns or correlations in reviewing behaviors or research trends. Here we point to a few possible research studies that can be accomplished using the CoRAL data, considering extensive usage of the tool in the future. Many more studies can be designed to use the valuable data stored in CoRAL.

CoRAL users perform a series of activities through time including paper reading, commenting, discussing, rating, and tagging. Each of these activities have a content. For example, content of the rating activity is derived from its numerical value and content of a comment contains the comment body and the reviewer's attitude. Based on the similarity of the researchers' activities and their activity contents, we can extract activity clusters. We envision at least two ways in which to explore these clusters:

Mutual Influence of CoRAL Activity Clusters and Other Sources of Collaboration: Researchers have various kinds of collaboration relations, such as co-authorship and friendship networks. There are many tools that target those relations and contain useful data that can be studied along with the CoRAL's data. As an example, ReaSoN [40] inspects and analyzes the research activities around producing and consuming publications including the co-authorship relations. The co-authorship networks may get influenced while researchers are involved in a global collaborative-reviewing system.

Considering researchers in the same activity cluster and analyzing the evolution of their other networks and relations may lead to conclusions about the social impact of having collaborating review activities. Being in the same activity cluster indicates that the researchers have common viewpoints, have shared experiences, have exchanged information, or simply had the chance to get to know each other. On the other hand, being in a previously formed collaboration relation may be a source of performing similar or close activities in CoRAL. While studying the effects of CoRAL on formation or evolution of collaboration networks and relations, one should also consider the reverse influence.

Previously Formed Relations and Interest to Follow Reviews: Another possible study can target the correlation between being two members of the same activity cluster and having an explicit following relation. In this study, we also need to collect information about friendship relations outside and before using CoRAL. To perform this study, one may seek for data in other social networking tools such as Facebook, or may ask users to explicitly mention whether they knew each other before using CoRAL when they intend to follow one's work. This would not be a surprising question knowing that other systems such as LinkedIn also ask such a question, although for different purposes. Having this information, enables the study of these cases: do people with previous connections perform similar reviewing activities? do people with similar reviewing intentions tend to follow each other's work through finding each other in collaborative reviewing? which people form majority of the researchers that a user follows: previously known friends, or those found through reviewing?

5.6 Summary

In this chapter, we talked about the tasks that can be accomplished as a result of using the proposed CoRAL system. Although the focus for development of this system is on supporting a convenient and effective environment for collaborative review of literature, this system brings various other benefits to the society of researchers. These benefits were identified and discussed in this chapter and a summary of those benefits comes in this section.

The main objective and hence the most important benefit of the system is supporting literature review. The features chosen for the developed system are in favor of deeper and broader review of literature. As an effect of choosing document-centric collection of reviews and openness of idea sharing without being bound to explicit sharing of documents, CoRAL introduces new collaboration links to the researchers. Also, collection of reviews in one place helps gathering interesting statistics about how literature is reviewed and by whom. Studying the review trends and attitudes can guide the researchers in knowing the literature and their community. In addition, collecting multi-criteria ratings helps development of an effective search system in which users are able to rank literature according to their search goals.

Another important aspect of this system is its ability to become a platform for analysis of reviews and reviewing activities. A few possible studies were proposed in this chapter, but many more possibilities can be anticipated.

Chapter 6

Possible Extensions and Integration

The CoRAL system as described in the previous chapters is ready to be used in the basic scenario of usage: collaborative reviewing of published literature on a desktop machine. As mentioned in Chapter 1, other usage scenarios are foreseeable for which CoRAL can still be a suitable solution. In this chapter, we discuss how the system can be extended to two other usage scenarios and how this is beneficial. In addition, this chapter talks about possible integration of Mendeley with CoRAL as an example of adding a new reviewing client to the system. Mendeley is a successful and popular application that has common aspects with CoRAL Viewer, making it a good candidate for integration. The benefit of this integration is adding a huge collection of valuable reviewing data to the CoRAL's collection of reviews, which is a big step towards reaching a universal collection.

6.1 Extension: The Conference Review Committee

In the previous sections of the thesis, we focused on the basic scenario of usage in which CoRAL is used to review published literature. Although the primary focus of this research is review of published or online literature, the implemented tool is also a good candidate for the conference review processes. As a branch to this research, we investigated this possibility by prototyping a sample limited version of such a system. Since it was only for investigation, the prototype is not ready for use but the idea is presented here.

The main difference between reviewing published literature and conference review process is in the privacy of the reviewers, reviewees, content of the review, and content of the document. Another difference is in consideration of the sequence of actions in a conference scenario. Every conference defines its own timing and sequence of actions that always accompanies privacy rules. Customizing the CoRAL components to understand sequence of actions and privacy rules and installation of a restricted-access API, produces a conference-specific CoRAL. This is mainly achieved by adding a custom proxy module to the API that will control the outgoing and incoming data according to the conference's sequencing and privacy rules.

In such settings, the CoRAL Viewer can introduce a new approach to conference reviewing by use of in-context annotations. Reviewers may also discuss their points in the content of the document in order to come to a collaborative decision. If the conference's settings allows, authors may at some point read and reply to the reviewers' comments inside the content. Using the CoRAL Reporter, the committee reviews the reviewer's activities and attitudes throughout the whole collection of papers, giving them quick insight on how everybody has accomplished their tasks. A conference may retain the reviewing data for future conferences in order to achieve a report of its reviewing trends. Using the architecturally same system that is used for reviewing published literature, a conference may decide to later publish some of its reviewing data to the CoRAL's public API. Ratings and tags are specifically useful and publishable in this case.

6.2 Extension: The Mobile CoRAL

In this thesis, we introduced the CoRAL Viewer as an example of an application that is used to read and review the literature. The CoRAL Viewer is specifically useful for the basic scenario of usage, which is designed for reviewing published literature from the convenience of a desktop application. However, every paper is not immediately accessible in a PDF form and the first time a paper is reviewed is not usually on a desktop. A huge amount of reviews takes place in conference paper and poster presentations, when people meet, discuss, and ideas form from the first time exposure to a content. Considering this fact, we investigated the ability of this system in dealing with such situations. Here, we propose a solution that is also a suggestion for new applications to be built around the API.

In a conference event, the easiest form of digital interaction with online services is use of smart phones. Considering that smart phones are growing in popularity and abilities, they can be convenient choices for reviewing activities. A trivial application may ask the user to manually enter the paper or poster's citation information prior to any reviewing activity. But, a better approach can be employment of QR codes [41] to digitally identify what publication is being reviewed and receive already known citation information. QR codes are gaining an increasing popularity in various applications, such as in libraries [1]. In order to use the QR codes, the authors or the conference should have already decided to support the system. After the publication is identified, the mobile interface can be used to add a note, tag, or rate. Using this method, we take advantage of the active reviewing going on in the conferences to enrich the collection of reviews. As a trade off, the reviews collected by this method lack the context-awareness and discussion ability. However, there are secondary solutions for these problems. For example, to pursue a discussion in CoRAL the mobile application may bring the list of comments made on the publication and support replying. For the lost context-awareness, user may be able to add a picture or snapshot of the piece of content she is reviewing which adds complexity to the user's actions.

6.3 Integration with Mendeley

Mendeley is a powerful library and reference management tool that also produces valuable statistics about reading trends in scientific literature. Mendeley Desktop supports notes, in-context annotations, citation extraction, tag, and one-star rating. These properties reminds us of a potential review producer application for the CoRAL system. In addition, CoRAL lacks support for personal libraries and sharing reading lists which is quite popular and useful. On the other hand, as discussed in Section 2.4, Mendeley lacks the ability for document-centric review access and open-

[1] http://www.libsuccess.org/index.php?title=QR_Codes

ness in reviewing. Lacking these two properties, it also cannot produce reports about how literature is being reviewed considering attitudes, interactions, timelines, and the actual review contents. According to these observations, Mendeley and CoRAL should be investigated for being integrated.

This type of integration is already simplified by both systems, since they both provide an open API. Mendeley's API provides user's library information and also gives public statistics. CoRAL API gives and accepts annotations on publications in addition to citations, ratings, interactions and etc.

Mendeley is a pioneer with many active users and the Mendeley Desktop is already supporting library management and creation and visualization of annotations. So, calling the CoRAL API from the Mendeley Desktop would be a better choice than the reverse. But in this case, the Mendeley Desktop needs to give the users the option to publicly publish their annotations upon their will which is not currently supported by Mendeley. Replacing the CoRAL Viewer with the Mendeley Desktop, the CoRAL Reporter can still be used and probably accessed via the Mendeley website. Making changes to Mendeley is not in our hands, this discussion is only examining possible improvements to the collaborative-reviewing activities considering an existing popular service.

6.4 Summary

In the previous chapters, we mainly concentrated on the basic scenario of usage for defining features and development decisions. The basic scenario is about reviewing published literature available on desktop. Two other scenarios were discussed in this chapter and the current CoRAL features and architecture was examined for its ability to support these scenarios. In this regard, we have examined a prototype of a limited access version of CoRAL for the conference review scenario. In this scenario, we consider CoRAL as an effective tool for encouraging in-context discussions between reviewers and with authors. In addition, it can be a help for conference chairs to investigate the attitudes and quality of reviews throughout the revisions. They can also benefit from having a collection of reviews and a record of reviewers for a longer period than one conference instance.

The other discussed scenario is about critical review of the literature as soon as they are presented in conference events. For this usage, a mobile application seems more favorable. The noticeable shortcomings of this approach are the inconvenient way to identify publications and lack of context-awareness in the produced reviews which are the effect of inaccessibility to the documents for making a review. The benefit of this approach is gathering reviews in conference events where literature is highly discussed and people can actually meet.

In addition to these usage scenarios, the possibility of integration of Mendeley with CoRAL was discussed in this chapter. This is a way to reach to reviews produced by many users of Mendeley, in addition to taking advantage of facilities that Mendeley provides for a more convenient review production.

Chapter 7

Conclusions and Future Work

This thesis noted that existing collaborative-review tools do not take full advantage of social potential of reviewing (Chapter 2). In this thesis, higher engagement of the principles of Web 2.0 are taken into consideration for an effective exploitation of social reviewing to benefit and improve scientific reviewing. In Section 2.4, the current approaches regarding development of collaborative literature-review tools are reviewed. According to this review and based on Web 2.0 principles, requirements for design and development of an open and social reviewing environment are identified and discussed (Section 1.1 and 2.4.5). These requirements lead to these considerations: publication-centric collection of reviews, providing an open API, sharing of ideas and open reviewing, knowing reviewers' attitudes, annotating and discussions in the document's context, context-aware reviews, multi-criteria rating of literature, and contextual analysis or reporting of the collected reviews. Based on these, we presented an open source system named CoRAL (Collaborative Review and Analysis of Literature) which consists of three main components: an open API, a review making tool (the CoRAL Reporter), and an analyzer (the CoRAL Viewer).

The CoRAL API is the core of the system where reviews are stored in and retrieved from. It enables collecting and analyzing reviews independent of the applications used for making reviews. The design and functionality of the API is described in Section 3.1. The Viewer is a desktop application used to browse and review the literature according to the standards that we have identified for effective collaborative review. It supports in-depth collaborative reviewing by allowing in-context discussions and respecting the context- and attitude-awareness requirements in collecting reviews. The Viewer provides open collaboration opportunities by its document-centric review approach. Free-vocabulary tagging and multi-criteria rating are other features of the Viewer. The details of development of the Viewer are discussed in Section 3.2. The Reporter is a web-based application designed to analyze reviews and generate visualized summaries of the collected reviews. In addition to its role in visualizing and analyzing the collected reviews, the Reporter can be used as a trust or reputation evaluator. This is achieved by visualizing and reporting users activities as an indicator of their reviewing reputation which is based on our discussion in Section 2.2.

The Viewer and the Reporter are example applications from an envisioned category of collaborative-review tools that can fit into the described CoRAL system. The description of the architecture and development decisions of the CoRAL components can be found in Chapter 3. A description of other potential tools that fit into the CoRAL definitions was discussed in Chapter 6.

In Chapter 4, we have discussed the features of the Reporter and the Viewer and illustrated usage storyboards. The Viewer, built upon the Fab4Browser, is a document and annotation viewer. Using the Viewer, one can browse documents in multiple formats, comment and reply to comments with specification of attitude and accompanying a piece of content, highlight, tag, rate, add or edit citation information, and choose between private annotating and publishing to the public. The Reporter generates reports about the review activities from various perspectives. Users can also manage their profiles using the Reporter. Publication-centric view and reviewer-centric view are the most interesting parts of the Reporter. In the former, reviews are collected around the publication and various types of reports are generated including time line of reviews, rating chart, interactions networks, summary of discussions, tag cloud, and managing citation information. In the latter, a summary of every reviewer's activities can be found in addition to means to specify interest in people and papers, and managing privacy settings.

Being a complex system which is in relation to scientific literature and human collaboration, CoRAL brings beneficial side effects that were discussed in Chapter 5. One of such side effects is providing a platform for future study and analysis of reviews and collaborative reviewing. In that chapter, we have also introduced the goal-based search that can benefit from multi-criteria rating that CoRAL provides.

7.1 Future Research Directions

CoRAL is a big project with multiple dimensions ready for more work. In this thesis, we have built the foundation for a framework to support collecting collaborative reviews that should be considered for further growth. Throughout this thesis, we have pointed out to the improvements and extensions that can add to the value and effectiveness of the proposed system from the development perspective. Examples of such improvements are: giving more flexibility to the API methods and functionality without endangering the system, and supporting discussions by enabling reply notes to point to more than one annotation in one or more publications. In addition to considering improvements to note-taking form of review, we should design ways to take advantage of highlights as an important indicator of interestingness of pieces of content.

Other envisioned enhancements in favor of collaborative literature review can focus on broader field review. We suggest developing more strong links between publications that goes beyond common tagging. This can be done as another collaborative task, such as the ability to connect publications from within the contextual comments based on common concepts, comparisons, or information linking. A summary of related reviews can be another interesting feature.

In this thesis, we introduced goal-based searching and how it is a benefit directly accomplished from the reviews and targeting the community of readers. The implementation and more research on the effective development of this idea remains for future work.

Another important future work concentrates on analysis of the collected reviews and reviewing behaviors. This work can be accomplished after the CoRAL community grows and enough data is collected in the real universal usage scenarios.

In addition, more work is needed for development or integration of related tools that can add to the flexibility and the collecting power of the total system.

References

[1] T. O'Reilly, "What is web 2.0: Design patterns and business models for the next generation of software," 2005. [Online]. Available: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

[2] P. Anderson, "What is web 2.0? ideas, technologies and implications for education," JISC, Tech. Rep., 2007.

[3] J. Howe, "Crowdsourcing: A definition," 2006, wired Blog Network: Crowdsourcing. [Online]. Available: http://crowdsourcing.typepad.com/cs/ 2006/06/crowdsourcing_a.html

[4] J. Surowiecki, *The Wisdom of Crowds*. Anchor, 2005, vol. 75, no. 2. [Online]. Available: http://www.amazon.com/dp/0385721706

[5] P. Denning, J. Horning, D. Parnas, and L. Weinstein, "Wikipedia risks," *Communications of the ACM*, vol. 48, no. 12, p. 152, 2005.

[6] S. Javanmardi, Y. Ganjisaffar, C. Lopes, and P. Baldi, "User contribution and trust in wikipedia," in *Collaborative Computing: Networking, Applications and Worksharing, 2009. CollaborateCom 2009. 5th International Conference on*, nov. 2009, pp. 1–6.

[7] B. T. Adler, K. Chatterjee, L. de Alfaro, M. Faella, I. Pye, and V. Raman, "Assigning trust to wikipedia content," in *Proceedings of the 4th International Symposium on Wikis*, ser. WikiSym '08. New York, NY, USA: ACM, 2008, pp. 26:1–26:12. [Online]. Available: http://doi.acm.org.login.ezproxy.library.ualberta.ca/10.1145/1822258.1822293

[8] E. H. Chi, "Augmented social cognition: using social web technology to enhance the ability of groups to remember, think, and reason," in *Proceedings of the 35th SIGMOD international conference on Management of data*, ser. SIGMOD '09. New York, NY, USA: ACM, 2009, pp. 973–984. [Online]. Available: http://doi.acm.org.login.ezproxy.library.ualberta.ca/10.1145/1559845.1559959

[9] C. Tenopir, D. W. King, S. Edwards, and L. Wu, "Electronic journals and changes in scholarly article seeking and reading patterns,"

ASLIB Proceedings, vol. 61, no. 1, pp. 5–32, 2009. [Online]. Available: http://www.emeraldinsight.com/10.1108/00012530910932267

[10] Z. Liu, “Reading behavior in the digital environment: Changes in reading behavior over the past ten years,” *Journal of Documentation*, vol. 61, no. 6, pp. 700–712, 2005. [Online]. Available: http://www.emeraldinsight.com/10.1108/00220410510632040

[11] L. Zhang, R. Kopak, L. Freund, and E. Rasmussen, “A taxonomy of functional units for information use of scholarly journal articles,” in *Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem - Volume 47*, ser. ASIS&T ’10. Silver Springs, MD, USA: American Society for Information Science, 2010, pp. 31:1–31:10. [Online]. Available: http://portal.acm.org/citation.cfm?id=1920331.1920376

[12] Qiqqa, Academic and research PDF management. [Online]. Available: http://www.qiqqa.com/

[13] JournalFire, Online Journal Clubs and Discussion. [Online]. Available: http://journalfire.com/

[14] M. D. P. Leland, R. S. Fish, and R. E. Kraut, “Collaborative document production using quilt,” in *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, ser. CSCW ’88. New York, NY, USA: ACM, 1988, pp. 206–215. [Online]. Available: http://doi.acm.org.login.ezproxy.library.ualberta.ca/10.1145/62266.62282

[15] Crocodoc. [Online]. Available: http://crocodoc.com/

[16] ReviewBasics. [Online]. Available: http://www.reviewbasics.com/

[17] Academia.edu, Follow Research. [Online]. Available: http://academia.edu/

[18] LinkedIn, World’s Largest Professional Network. [Online]. Available: http://www.linkedin.com/

[19] F. Corubolo, P. Watry, and J. Harrison, “Location and format independent distributed annotations for collaborative research,” in *Research and Advanced Technology for Digital Libraries*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4675, pp. 495–498.

[20] CiteULike: Everyone’s Library. [Online]. Available: http://www.citeulike.org/

[21] Mendeley, Academic reference management software for researchers. [Online]. Available: http://www.mendeley.com/

[22] ResearchGATE. [Online]. Available: http://www.researchgate.net/

[23] Computing Reviews, the leading online review service for computing literature. [Online]. Available: http://www.reviews.com/

[24] V. Henning, J. J. Hoyt, and J. Reichelt, "Crowdsourcing real-time research trend data," in *Significance*, 2010.

[25] V. Henning and J. Reichelt, "Mendeley - a last.fm for research?" *2008 IEEE Fourth International Conference on eScience*, pp. 327–328, 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4736778

[26] H. Zaugg, R. E. West, I. Tateishi, and D. L. Randall, "Mendeley : Creating communities of scholarly inquiry through research collaboration," *TechTrends*, vol. 55, no. 1, 2011.

[27] T. A. Phelps and R. Wilensky, "Robust hyperlinks: Cheap, everywhere, now," in *Proceedings of Digital Documents and Electronic Publishing*, ser. Lecture Notes in Computer Science, 2000.

[28] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321 (Informational), Internet Engineering Task Force, Apr. 1992, updated by RFC 6151. [Online]. Available: http://www.ietf.org/rfc/rfc1321.txt

[29] T. A. Phelps and R. Wilensky, "Multivalent documents: Inducing structure and behaviors in online digital documents," in *Proceedings of the 29th Hawaii International Conference on System Sciences Volume 5: Digital Documents*. Washington, DC, USA: IEEE Computer Society, 1996, pp. 144–152. [Online]. Available: http://portal.acm.org/citation.cfm?id=795702.798640

[30] ——, "Multivalent annotations," in *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*. London, UK: Springer-Verlag, 1997, pp. 287–303. [Online]. Available: http://portal.acm.org/citation.cfm?id=646630.759387

[31] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine, 2000.

[32] R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Transactions on Internet Technology (TOIT)*, pp. 115–150, 2002.

[33] D. Crockford, "Introducing json," May 2009. [Online]. Available: http://json.org. Retrieved: July 3, 2009.

[34] D. Rathi and L. Given, "Research 2.0: A framework for qualitative and quantitative research in web 2.0 environments," in *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, jan. 2010, pp. 1–10.

[35] X. Y. Wang, D. G. Feng, X. J. Lai, and H. B. Yu, “Collisions for hash functions md4, md5, haval-128 and ripemd,” rump session of Crypto’04, E-print, 2004.

[36] National Institute of Standards and Technology (NIST), FIPS Publication 180-3: Secure Hash Standard, October 2008.

[37] T. V. Wal, “Folksonomy coinage and definition.” [Online]. Available: http://vanderwal.net/folksonomy.html, Published: February 2 2007, Retrieved: May 31 2011

[38] H. Halpin, V. Robu, and H. Shepherd, “The complex dynamics of collaborative tagging,” in *Proceedings of the 16th International Conference on the World Wide Web (WWW’07).* Banff, Canada: ACM Press, 2007, pp. 211–220.

[39] T. A. Phelps and R. Wilensky, “The multivalent browser: a platform for new ideas,” in *Proceedings of the 2001 ACM Symposium on Document engineering*, ser. DocEng ’01. New York, NY, USA: ACM, 2001, pp. 58–67. [Online]. Available: http://doi.acm.org/10.1145/502187.502197

[40] V. Ganev, Z. Guo, D. Serrano, B. Tansey, D. Barbosa, and E. Stroulia, “An environment for building, exploring and querying academic social networks,” in *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, ser. MEDES ’09. New York, NY, USA: ACM, 2009, pp. 42:282–42:289. [Online]. Available: http://doi.acm.org/10.1145/1643823.1643875

[41] QR Code official website. [Online]. Available: http://www.denso-wave.com/qrcode/index-e.html

[42] Fab4Browser Annotation Model (XML Schema). [Online]. Available: http://bodoni.lib.liv.ac.uk/AnnotationSchema/AnnotationSchema.xsd

Appendix A

API Documentation

In this appendix a brief documentation of basic CoRAL API methods is brought, as the current version supports. The following notes apply to description of all the methods in this appendix:

- The password fields should be sent encoded by SHA1 algorithm.
- The JSON arrays provided to explain the form of the request and response messages contain key and value pairs. The keys are constant values that are specified for every message. The values are arguments of the input or output message. In this document, the constants are shown by normal font and the variables are shown by italic font.
- The *base_address* is the address on which the API is deployed and made accessible.
- ‘Serialized annotation’ refers to the XML representation of the annotations according to Fab4Browser’s annotation schema [42]. This schema contains the publication ID and reviewer ID as annotation properties, so they are not present in the other fields of the corresponding JSON message. The content and attitude associated to the content annotations are optional properties embedded in the ‘xml_body’ attribute of the XML schema.

A.1 Reviewer Methods

This section documents API methods that directly update user information.

A.1.1 Authenticate and/or get user information

Functionality: Authenticates user with the given username and password. Returns user's information.

Input: a JSON array containing username and password.

Input JSON format:

{"jaxbBean":[{"key":"username","val":"*username*"},{"key":"pass","val":"*password*"}]}

Output: A JSON array of key-value pairs. One always present key is 'state' which shows the state of the operation. Values for state are: '0' if authentication is successful, '1' if password is wrong, '2' if username does not exist, and '3' for server fault. If the operation is successful (state = 0), then user information is also sent by these keys: 'name','email', and 'aff' for affiliation.

Output JSON format:

for a successful authentication:

{"jaxbBean":[{"key":"state","val":"0"},{"key":"name","val":"*name*"},{"key":"email","val":"*email*"}
,{"key":"aff","val":"*affiliation*"}]}

for a sample unsuccessful authentication (wrong password):

{"jaxbBean":[{"key":"state","val":"1"}]}

Address: *base_ address*/publishService/users/auth

A.1.2 Register a new user

Functionality: Registers a new user with given information.

Input: User's information: username, name, password, email, affiliation, description. All these keys are required. You may leave the values as empty strings for description and affiliation.

Input JSON format:

{"jaxbBean":[{"key":"username","val":"*username*"},{"key":"password"
,"val":"*password*"},{"key":"name","val":"*name*"},{"key":"email","val":"*email*"}
,{"key":"affiliation","val":"*affiliation*"},{"key":"description","val":"*description*"}]}

Output: An integer as a string: 0: operation was successful, 1: username already exists, 2: email already exists, and 3: server fault.

Address: *base_ address*/publishService/users/add

A.2 Add/Delete/Retrieve Annotations

The following methods are for adding and retrieving content annotations, tags, and ratings. A delete operation is also available for content annotations.

A.2.1 Post a new annotation

Functionality: posts a new content annotation, tag, or rating on a publication by a user to the API.

Input: the username and password of the user who is making the annotation. And, an XML representation of the annotation (serialized annotation).

Input JSON format:

{"jaxbBean":[{"key":"username","val":"*username*"},{"key":"password","val":"*password*"}
,{"key":"anno","val":"*serialized annotation*"}]}

Output: an integer value representing the output code. 0: ok, 1: user not authenticated, 2: server fault, and -1: problem with parsing the annotation.

Address: *base_ address*/publishService

A.2.2 Delete a comment

Functionality: deletes a comment given its ID. A comment can only be deleted by its author (authenticated by username), and only if there is no reply on that comment.

Input: the username and password of the person who is deleting the comment (it must be the author). And, the ID of the comment as it is given by the API when retrieving the comments. The comment ID must be an integer.

Input JSON format:

{"jaxbBean":[{"key":"username","val":"*username*"},{"key":"password","val":"*password*"}
,{"key":"annoId","val":"comment's ID"}]}

Output: an integer value representing the output code. 0: ok, 1: not authenticated, 2: exception, 3: cannot delete because it has a reply, 4: user is not the author (deletion not allowed).

Address: *base_ address*/publishService/del

A.2.3 Retrieve comments of a publication by URL

Functionality: given a URL, all annotations of that specific URL can be retrieved by this service.

Input: the URL. If the URL is local, it should start with: "username:file:/", and global URLs start with HTTP or FTP.

Output: a JSON array of serialized comments retrieved for the URL.

Output JSON format: as a special case, here only the keys are important. This is done in this way because we are using a single function to create and send all request and responses. Here, each response object only needs one key or value (not a key-value pair). The serialized comment goes into 'key' and the value is ignored. This could have been the reverse, giving key a name or a number showing the index, but the index does not matter and it is easier to retrieve if one only looks into keys.
{"jaxbBean":[{"key":"*serialized comment[1]*","val":null}

,...

,{"key":"*serialized comment[n]*","val":null}]}

Address: *base_ address*/searchService/

A.2.4 Retrieve comments of a publication by BibTex

Functionality: given a URL, all annotations of the publications whose bibliographic information match with the bibliographic information of the specified URL can be retrieved by this service.

Input and output: input and output are the same with the URL search (Section A.2.3).

Address: *base_ address*/searchService/sameBib

A.2.5 Retrieve comments of a publication by digest

Functionality: given a document textual or binary digest, all annotations of the matching publications can be retrieved by this service.

Input: the value of the digest and the type of digest (textual or binary) should be specified.

Input JSON format:

- for the binary document digest:

  {"jaxbBean":[{"key":"checksum","val":"*binary digest converted to string*"}

  ,{"key":"method","val":"MD5"},{"key":"encoding","val":"base64"}]}

- for the textual document digest:

  {"jaxbBean":[{"key":"checksum","val":"*textual digest converted to string*"}

  ,{"key":"method","val":"multivalent annotation"}

  ,{"key":"encoding","val":"base64-MD5-text"}]}

Output: similar to the output of the URL search (Section A.2.3

Address: *base_ address*/searchService/checksum

A.3 Adding a Publication

Functionality: This service should be called if a publication is accessed. In order to: first, add the publication and its information if it does not exist, second, to store users' reading information.

Input: JSON array with following key-value pairs:

- **Required:** username of the reader (username), URL of the publication (url) even if locally accessed.
- **Optional:** There are two ways to call this service by adding optional parameters, according to the available information. But, only one of them should be chosen for each service call:
  - If the bibtex is available: add full bibtex entry as the optional input.

  - If separated information about publication is available, not in the form of a bibtex entry: parameters are: title (required if this set of parameters are chosen), any of these: abstract, keywords, full authors information

Notes:

- If the URL is local, it should start with: "username:file:/"
- Keywords should be separated with ';' or ','
- 'full authors info' starts with the list of authors as a string like this: "first author, second author, etc author" or "first author, second author, and etc author". The list of authors may follow an ordered list of affiliations and/or emails.

Input JSON format:

- Input format without any information about the publication:

  {"jaxbBean":[{"key":"username","val":"*username*"},{"key":"url","val":"*URL*"}
  ,{"key":"bibtex","val":null}]}

- Input JSON format with bibtex available:

  {"jaxbBean":[{"key":"username","val":"*username*"},{"key":"url","val":"*URL*"}
  ,{"key":"bibtex","val":"*bibtex*"}]}

- Input JSON format with other forms of data available:

  {"jaxbBean":[{"key":"username","val":"*username*"},{"key":"url","val":"*URL*"}
  ,{"key":"title","val":"*title*"},{"key":"abstract","val":"*abstract*"}
  ,{"key":"keywords","val":"*keywords*"}
  ,{"key":"fullAuthorInfo","val":"*full Author Info*"}]}

Output: an integer as a string: the ID of the publication being added or updated
Address: *base_ address*/publishService/resources/add

A.4 Following

This section covers methods needed to add following relationships.

A.4.1 Request to follow someone's work

Functionality: User A requests to follow user B's work in CoRAL. System sends a notification email to user B and adds the following relation as a one-way relationship (A->B).

Input: User A's username, User B's username, User A's password

Input JSON format:

{"jaxbBean":[{"key":"requester","val":"*A's username*"},
{"key":"requested","val":"*B's username*"},{"key":"password","val":"*A's password*"}]}

Output: An integer (sent as string) showing the status of operation:

- HTML error codes if someone attempts the request with invalid arguments or extra arguments.
- Or, 1: ok, 2: if the requested person is not a valid user, 3: if user A fails in authentication, 4: email sending failed but relation is added, 5: server fault

Address: *base_ address*/publishService/friends/follow

A.4.2 List who I follow

Functionality: List who user A follows.

Input: User A's username, User A's password (password)

Input JSON format:

{"jaxbBean":[{"key":"username","val":"*A's username*"},{"key":"password","val":"*A's password*"}]}

Output: list of followees and their information in a JSON array.

Output JSON format:

{"User":[
{"name":"*name[1]*","username":"*username[1]*","affiliation":"*affiliation[1]*"
,"description":"*description[1]*", "email":"*email[1]*","id":"*ID[1]*"},
... (same object for other followees).... ,
{"name":"*name[n]*","username":"*username[n]*","affiliation":"*affiliation[n]*"
,"description":"*description[n]*", "email":"*email[n]*","id":"*ID[n]*"}
]}

Address: *base_ address*/publishService/friends/listFriends