

Face2face social bookmarking with recommendations: WebQuests in the classrooms

Raffaella Grieco Delfina Malandrino Giuseppina Palmieri Vittorio Scarano

ISISLab

Dipartimento di Informatica ed Applicazioni “R.M. Capocelli”

Università di Salerno, Italy

Email: {rafgri,delmal,palmieri,vitsca}@dia.unisa.it

Abstract—In this paper we present SYNCBOOK, a distributed system that offers the functionalities of a face-to-face cooperative bookmarking system and of a recommendation system. Our overall objective was to design and realize a practical tool that can be used in project-based learning, in the classroom, to participate in the WebQuests [12] or to cooperatively build an annotated *Webliography*, i.e., a set of URLs, organized and scaffolded with group annotations. Moreover, our system also offers advanced awareness tools as well as recommendations based on the items in the *Webliography* and their *Google-related* pages.

I. INTRODUCTION

A lively field of research on the World Wide Web has the goal of transforming the Web from an *invisible* place, i.e. where the navigation of others was unknown to a user, to a *social* place, by tracking users presences and behaviors, by providing a compact and augmented overview of the navigated hyperspace, by allowing guided tours, by also allowing adaptation, personalization and recommendation mechanisms.

The rationale is that we are used, in our daily activities, to be continuously affected by the ideas, the opinions and the behaviors of others. In particular, this is well-known among the pedagogists: *situated learning* [24] is defined as a function of the activity, context and culture and is strongly based on the social interactions [31] among learners, and among learners and the environment.

In general, (remote) collaboration on the Web is, nowadays, available in many shapes. The effect is that the, so called, *social navigation* on the Web improves the task of finding relevant information by making users aware of other users activities, who have similar tastes or interests. Moreover, social navigation uses information about other people’s navigation to give suggestions on what to do. The idea is that by inspecting users behaviors and interactions it is possible to outline common interests and suggest information related to them. Finally, interactions and communications among users are useful in aid learning and problem solving other than experience sharing.

Another shape of (remote) collaboration is the *social bookmarking* that is an increasingly popular way to save, locate, categorize and share interesting Web resources through lists of accessed URLs. By organizing subsets of information it is possible to reduce the cognitive overhead that affects users when overwhelmed by a large amount of interesting information.

Bookmark systems represent a well-known strategy for building these personal information spaces and for placing landmarks on them, by allowing a simple retrieving of items that have previously found and thought to be interesting.

By making them cooperatively accessible by any user, they allow users to take advantage of the expertise and the knowledge of other similar users and, in the meantime, discover new users with similar interests.

A. Our goals

Our research lies at the intersection of different research areas, and in particular, the areas of cooperative bookmark systems, synchronous and adaptive systems and aims to the educational setting (often called Computer Support for Collaborative Learning (CSCL)), where face2face (i.e. synchronous and co-located) computer-mediated interaction is still largely unexplored (to wit, a recent survey in [25] provides only few examples of systems that are specifically designed for face2face interaction). As a matter of fact, the research is mainly focussed in bridging space and time with collaborative environments.

On the other hand, our overall objective is to build a cooperative synchronous tool that allows and stimulates the face2face, classroom activities to participate in WebQuests or to build *Webliographies* i.e. a set of usefully grouped, annotated URLs that are the final result of a cooperative learning workgroup on a specific project.

Our result is a distributed application, SYNCBOOK, that allows cooperation among learners in a classroom while performing a Web search, guided by the facilitator and with appropriate awareness of the group’s behaviour, comments and annotations, as well as with the active support of the tool that adapts the HTML pages “on-the-fly” in such a way to emphasize possible interesting links. The idea is that our tool can foster cooperative learning environments that [21] challenge students to seek out new ideas, build content connections, work with others, and develop flexible communications through collaborative, generative, and interactive projects. SYNCBOOK is, to the best of our knowledge, the first tool that offers all these functionalities in an integrated, smooth and efficient way.

B. The pedagogical context

Collaboration and argumentation seem crucial in the learning process [3] since they force learners to express more formally their thoughts and listen and react to other persons' ideas. Supporting group awareness in synchronous systems is very difficult since all operation for discovering new users and their interests, and for distributing these information to all people participating into the group, must be implemented from scratch. Several recent examples of using face-to-face argumentation in the classroom by using ad-hoc tools are shown in [27], [30].

SYNCOBOOK has been designed by observing that technology is not currently able to support an important task of cooperative work in the classroom, namely, the research of relevant material (on the Web) about a particular topic. What is usually done is, often, only to sum up the individual navigations on subtopics, such as, e.g., part of the classroom is looking for material on the life of an artist while the other part is working on material on his/her work. In general, social bookmarking tools suffer from the lack of immediateness, i.e., no awareness of others' actions are available to each learner. Also, the navigation and the search for material is not guided by previous choices, i.e., accessing material that is "close enough" to URLs that have already been inserted into the common Webliography should be emphasized to each learner.

As an archetype way of using the Internet and the resources on the World Wide Web as learning tool, among the most popular ones, we can find WebQuest. A WebQuest is "an inquiry-oriented activity in which some or all of the information that learners interact with comes from resources on the Internet, optionally supplemented with videoconferencing", according to the definition provided by the originator of WebQuest, Bernie Dodge¹ [12], [13]. WebQuests are based on constructivist learning theory and have a considerable role in project-based learning and teaching in classrooms; there is a extensive repertoire of WebQuests, in very diverse contexts (e.g., [8]) and for any kind of audience (i.e. from K-12 to adults, to young people with disabilities [18]): its success is documented by the "more than a decade"-long widespread usage [23].

WebQuests should contain at least the following building blocks: (a) an introduction, to provide background information; (b) a task, that could be a problem to be solved or anything that involves the learners to analyze and transform the gathered information; (c) a process, that delineates the steps the learners should follow in accomplishing the task; (d) a set of evaluation criteria to evaluate the work accomplished by learners; (e) a conclusion, to inform the learners about their results and finally a teacher page, that could be useful for other teachers to implement WebQuests. An important consideration affects the cooperative goal of a WebQuest, in fact, learners are divided in groups and they must cooperate in order to achieve knowledge and contribute, by using only problem-

solving strategies and cooperative learning, to the knowledge of the whole classroom.

Organization of the paper: The paper is divided in 3 sections: the next one will introduce related work and background from the four areas our research is stemming from. In Section III, we present the system, by providing an overview of the distributed architecture of SYNCOBOOK, a description of the functionalities and some details on the implementation of the components; in Section IV we give an example tour of its characteristics. Finally, in Section V, we will present some final comments, and conclude the paper by illustrating further work that is undergoing in the project.

II. RELATED WORKS AND BACKGROUND

Our work lies at the intersection of four distinct research areas: (a) face to face, synchronous collaboration, (b) adaptive hypermedia [6] and recommendations systems, (c) social navigation and (d) social bookmarking. We think that to put in the right perspective the objectives of our work, a presentation of related works in these areas is helpful.

Face to face synchronous collaboration systems: Seeing users during their navigation can suggest information about the interest around a particular topic and the overall relevance of a document achieved by analyzing important information, such as who have accessed it, who have rated it and the number of access time to it.

Understanding other people's activities can provide a context for the overall group's activities while the individual contribution of each user could be relevant for the experience of the whole group. Unfortunately, the systems that envisage this scenario are tailored to business, decision-making processes, and are not specifically tailored for the educational setting (i.e. they can support argumentation and problem-solving, but not the collaborative information finding).

GroupSystems² is a software platform for business process management and collaboration. It supports ad hoc collaboration between leaders and participants. In particular, GroupSystems is a collection of tools that support group processes such as brainstorming, list building, information gathering, voting, organizing, prioritizing, and consensus building. In GroupSystems, works are grouped into folders containing an Agenda and activities for a meeting or project.

MeetingWorks³ is a software that can be used in a meeting room where each participant is connected on a Local Area Network. A server participant, the chauffeur, has the main goal to control the meeting. Each participant is assigned a unique ID. It can be used for face-to-face meetings and remote meetings (with participants that are not in the same meeting room). Each meeting is organized through an Agenda, a list of all the topics and tasks, called steps, that will take place during the meeting. Steps take place in a specified sequence and the results of one step can be used by subsequent steps to complete their tasks.

¹See the WebQuest web site at <http://webquest.sdsu.edu/> for further information and examples.

²<http://www.groupsystems.com/>

³<http://www.entsol.com/>

Among others systems, we can cite the Multi-user Awareness UI toolkit (MAUI) [15], a Java-based awareness system that provides components for collecting, distributing and visualizing group awareness information and for easily develop interfaces that support group awareness. The MAUI system is very useful in face to face situations, when it is important to observe the effects of other people's activities and their interactions with the objects available in the shared space. In particular, MAUI toolkit provides components for telepointers, participant lists and chat tools.

Adaptive hypertext and recommendation systems: Sharing of knowledge through recommendation, also know as collaborative filtering, represents the best way to distribute useful information. In particular, active filtering involves users to provide pointers to interesting documents or resources, while automatic recommendation involves the application of algorithms to inspect users' behaviors and interactions, and then, achieve useful information to suggest to other member of the group.

The SELECT project [2] consists of a proxy-based system whose main goal is to help users to easily and quickly find interesting information by reducing the information overload. SELECT exploits two filtering techniques: the first makes recommendations according to the users' decisions taken in the past, and the second takes advantage from other users' experiences by making recommendations according to their behaviors.

Fab [4] is an implementation of a hybrid content-based, collaborative Web pages recommendation system by inheriting the advantages of both approaches. Collaborative recommendation allows to compute the similarity of the users and not the similarity of the items.

SiteSeer [28] is a collaborative recommendation system, where user profiles are extracted by bookmark files, taking into account the content of the referenced documents and the structure of the bookmark file. Folders represent the user's categories of interest. The recommendation is realized by analyzing the bookmarks provided by the users and discovering matching among them, in terms of URLs contained therein.

Social navigation: In the context of educational collaborative environments we can cite different systems like CoFIND [14], Educo [20] and Knowledge Sea II [16].

CoFIND (Collaborative Filter In N Dimensions) is a Web-based collaborative learning environment that organizes the online resources through a voting mechanism. Learners can associate different types of qualities, such as *simple*, *funny*, *good for beginners*, with each resource to organize them in the system. As a result, the system is mainly user-centered (even the teacher can make interventions) since the database of URLs pointing to learning resources is created and maintained by the learners who use it. Qualities are used to give ratings to resources which provide potentially indications of the value of a resource. A larger number of quality labels imply more useful results for the learners that follow those links.

Educo is a collaborative learning environment that supports synchronous and asynchronous social navigation. The

synchronous social navigation is achieved through real-time discussion via chat rooms. The asynchronous social navigation is achieved through an annotation mechanism, where resources are rated according to their access (i.e., number of visits). The asynchronous feature is interesting for online settings in which users can access resources at a different times and from different locations. It is also possible to search for a specific document in the system and to be informed if some condition occur.

Knowledge Sea II has been developed to help students of a classroom to find useful information in online textbooks or tutorial pages. To this aim it provides traffic-based and annotation-based navigation support. The idea of the traffic-based social navigation support is to visualize the navigational history of each group of learners. Resources are organized in an information map, where each cell include links to similar pages. The background color of the cell represents the intensity of the group access to the documents in that cell. By following the evolution of the color of the cells, students can have an idea of the position and, therefore, the interest of others. The system also allows students to add annotations, augmented with visual cues. A small sticky note highlights the presence of annotations while a thermometer icon shows the *temperature* of the annotations (warmer is the temperature positive is the annotated page). Finally, student are able to perform keyboard-based search and get recommendations about useful links.

Social bookmarking: Recently there has been an explosion of the social bookmarking phenomenon as shown by the popularity of sites such as del.icio.us⁴, CiteULike⁵, Connotea⁶ and much more. These systems allow to save bookmarks on a remote database in order to make them accessible to everyone from any location in the world.

In particular they exhibit the following common functionalities: storing of bookmarks, through user-defined tags or keywords (to allow a quickly further access to them), sharing of these resources with other users, browsing by tag or category and, finally, discovering of new interesting links through recommendation mechanisms.

The *tagging* mechanism, widely used other contexts than bookmarking such as to classify photos in Flickr⁷ or to categorize ideas and projects as in 43things⁸, does not impose a rigid taxonomy, conversely it allows users to use any different term they prefer to classify documents.

Although most of the systems are able to promote collaboration (by allowing the creation of community of users around a topic) and ensure the awareness of other users' experiences, some drawbacks include the confusion produced by interfaces that are not user-friendly and the errors produced by the tagging mechanism (i.e. the definition of different tags for the same topic). Another consideration is that the choice of tag follows something that resemble the Zipf law curve [32],

⁴<http://del.icio.us/>

⁵<http://www.citeulike.org/>

⁶<http://www.connotea.org/>

⁷<http://www.flickr.com/>

⁸<http://www.43things.com>

well-known in Web-related traffic studies and measurements. Few tags could account for a high percentage of all the tags chosen, and a long tail of few other tags make the remaining low percentage, with most of them that occur just once or twice [5].

Tagging allows to add a specific keyword to URLs to bookmark, but very often the authors itself forget the reasons for the bookmarking of a specific resource. In this context Diigo [11] is able to offer the annotation functionality to the traditional social bookmarking mechanism. Users can highlight any text in a Web page and thus bookmark the page by saving the corresponding highlighted text. It is possible to easily extract and compile all highlights across multiple pages for a given subject, exchange viewpoints on any specific area of a Web page, useful for collaboration or debating an issue.

WebTagger [17] is a bookmark system that allows users to store, organize and evaluate URLs both personally and within a group. The system allows users to concentrate on a tagging of URLs with the most appropriate category rather than forcing users to select a folder in a strong hierarchy. Users can supply feedback, whereas they are later used to provide dynamically updated ranking of resources.

DAD [10] is a Distributed Adaptive Directory for cooperative bookmarks management. It provides a Peer to Peer environment that allows to share bookmarks with all the users connected to the system. DAD also offer an adaptive environment since suggestion or recommendation about the navigation are provided by taking into account the bookmark categories, the feedback provided by the users and the structure of the Web. Finally, adaptivity is achieved by employing an extended version of the Kleinberg [19] algorithm, used in systems like CLEVER [7].

The GiveALink [29] is a social bookmarking site where people can donate their bookmarks to the Web community. Bookmarks are analyzed to build a new Web mining technique and new techniques to search, recommend, personalize and visualize the Web. In particular, Web bookmarks, provided by users, represent the set of URLs that the system analyzes to find what Web pages are important to people and then to discover similarities among them.

The Dogear social bookmarking system [26] has been developed to investigate the usefulness of a social bookmarking application for a large enterprise. The Dogear system allows to save bookmarks by tagging and adding comments to them. Moreover, the Dogear system provides information about the bookmarks more recently added (title, description, creation date, author info) and a list of people that have bookmarked the same set of URLs.

III. SYNCBOOK

In this section we describe SYNCBOOK, its functionalities, its main components and their communication mechanisms.

A. Architecture overview and dataflow

The SYNCBOOK's framework consists of three main components, that is, the *Gateway-component*, the *Browser-component*, and the *Proxy-component* (see Fig. 1).

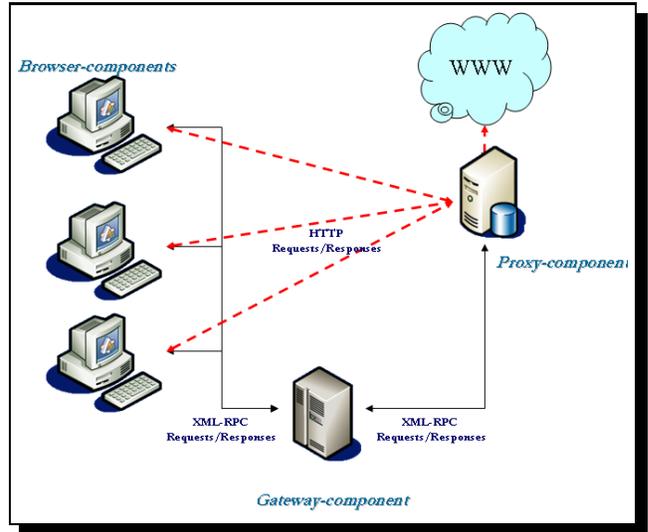


Fig. 1. SYNCBOOK's overview.

The *Browser-component* represents the user interface to the users and takes care of the visualization of data in the Explorer Bar within the main Explorer window. This component is implemented in C#, under the .NET framework and is shown as an additional bar into the Microsoft Internet Explorer main window.

The *Gateway-component* is placed midway between all the *Browser-components* and the *Proxy-component*. Its tasks are the coordination and synchronization of the clients activities and the contemporaneous communication with the *Proxy-component*.

The *Proxy-component* is an HTTP-intermediary application, developed on a programmable HTTP proxy, that is used (as HTTP proxy) by all the clients during their navigation. Its goal is to collect users' activities (sent back to the Gateway-component for broadcasting) and adapt the pages by showing relevant information embedded within the HTML page.

The communications among the components are all realized by encoding data in XML format and by using XML-RPC communication protocol.

B. SYNCBOOK's functionalities

SYNCBOOK's requirements are realized by offering the following functionalities to two different categories of users: the *facilitator* (teacher/instructor) that initiates the cooperation by loading (to all the users) the Webliography template and the *learners* that can add/annotate bookmarks but cannot perform the initiation of the Webliography.

First of all, SYNCBOOK offers a shared bookmarking annotation system, where bookmarks can be saved and shared within a community of learners. Then, it acts as a collaborative filtering system since it provides useful information and recommendations by taking into account the navigation and the bookmarking of users. Of course, as a cooperative system, it offers to each user the awareness of other learners's behaviours and navigations, which are important not only for the real-time

feedback to learners but also as a support to the facilitator that can direct, if necessary, learners' actions. The facilitator can also "guide" the navigation of the learner through a "follow me" mechanism that forces his/her navigation to follow facilitator's. Finally, reports of all the bookmarks, annotations and use is available to the facilitator, for documentation and/or further analysis.

We, now, describe in detail these functionalities.

1) *Social Bookmarking*: In SYNCBOOK, bookmarks can be saved, annotated with interesting observations and shared within the community. In particular, any learner can add a new bookmark in any folder of the provided Webliography. The initial Webliography, i.e. the empty hierarchical structure with folders, is loaded by the facilitator and, then, the *Gateway-component* takes care of distributing the copies to the learners and to keep it synchronized in real time, by broadcasting any modification to all the *Browser-components*. The capability of providing a "scaffolded" environment for the group activity is often underlined by practical experiences of WebQuests [23] and the facilitator can use the initial Webliography as a temporary structure designed to help learners in their tasks.

Any bookmark can be annotated with comments to it, the first one being the creator annotation, plus additional information (such as creation date/time and the author's name). All information are saved in a XML format and are available, also, in the final report.

2) *Awareness*: The higher the awareness of a shared workspace, the better the cooperation among users; this is even more crucial when face-to-face collaboration, within the classroom, is employed, as it is the case in our scenario, since a high-quality awareness enhances and stimulates verbal, informal communication among the learners in the classroom.

Any learner is aware of other learners' activities around the bookmarks already inserted, since, if a (single) learner is accessing a URL in the Webliography, an icon is added to the corresponding bookmark, and a different icon is added several learners are accessing it. The information is derived by the *Proxy-component* and is sent to the *Gateway-component* that broadcasts it to the clients. By hovering the mouse over the bookmark, it is possible to find the list of all the users that are accessing the URL.

3) *Recommendation*: The *Proxy-component* of the architecture is able to retrieve the current Webliography by the *Gateway-component* in such a way to provide active support by adding information to the anchors in the HTML page requested by a client. As a matter of fact, the support distinguishes among three type of anchors:

- *Level 0 Matching*: When a user is requesting an HTML page whose URL is saved in the Webliography, an alert window is displayed, containing information about the author of the bookmarking, the date/time of the creation and all the annotations.

- *Level 1 Matching*: The user is requesting an HTML page that contains a link to a bookmark of the Webliography. The link to the URL in the Webliography is highlighted with an icon added by the *Proxy-component*. By clicking on

the icon, it is possible to get information through a hidden layer window (implemented with HTML <DIV with some Javascript) about the author of the bookmarking, the creation date/time and comments, if present. In the same place, a link [Open Tree] allows to *highlight* the position of the bookmark within the treeview of the *Browser-component* by unfolding all the folders on the path from the Webliography root to the bookmark. This action, although locally shown to the user, is accomplished in a distributed way by querying the *Gateway-component* to send back an `expand` command to the user's *Browser-component*.

- *Level 2 Matching*: The user is navigating a Web page that contains a link whose destination is *related* to a bookmark previously saved in the Webliography. The notion of "relationship" has been based on the Google "related" feature, but only slight modifications are required to use other knowledge engines. As for Level 1 matching, an icon is added by the *Proxy-component* and it is possible to get additional information about the bookmark. The "highlight" mechanism is also provided here, unfolding all the folders from the Webliography root to the bookmark that is (Google-) related to the link in the requested HTML page.

4) *Follow me*: With SYNCBOOK, the facilitator can become a guide, by synchronously forcing all the clients to show the URL that is viewed by the facilitator. In this way, the teacher can intervene, whether appropriate, at any time, by showing to all the learners useful pages, or pre-assembled material for initiating or concluding the cooperative work about a topic. For examples, if an initial set of bookmarks are provided by the facilitator, they can be synchronously shown and orally commented and illustrated to the learners.

This is an important characteristics to be included: to wit, the research on WebQuest experiences [22] underlines that the most successful learning experiences were those where the facilitator was paying careful attention to the students activity (which is provided by the awareness functionalities) and is able to guide them through critical stages of their projects.

5) *Reporting*: Of course, the whole Webliography is stored centrally (on the *Gateway-component*) and is available to the facilitator as an HTML page (as well as the source XML code) to simplify storing and further analysis of the work.

Again, the opportunity to provide a detailed and easily built final document represents an important step for applications to project-based learning, such as WebQuest, where the end of the project is documented by the organized results of the learners but also by teacher material about the WebQuest.

C. The architecture

The three components of SYNCBOOK are leveraging on different programming languages and environments. The *Browser-component* is implemented in C# under the Microsoft .Net Framework in order to provide the interface to the system. The synchronous collaboration is performed by using the *Gateway-component* that is a stand-alone application written in Perl. The *Gateway-component* uses Frontier [1] for the XML-RPC communication and represents a centralized component

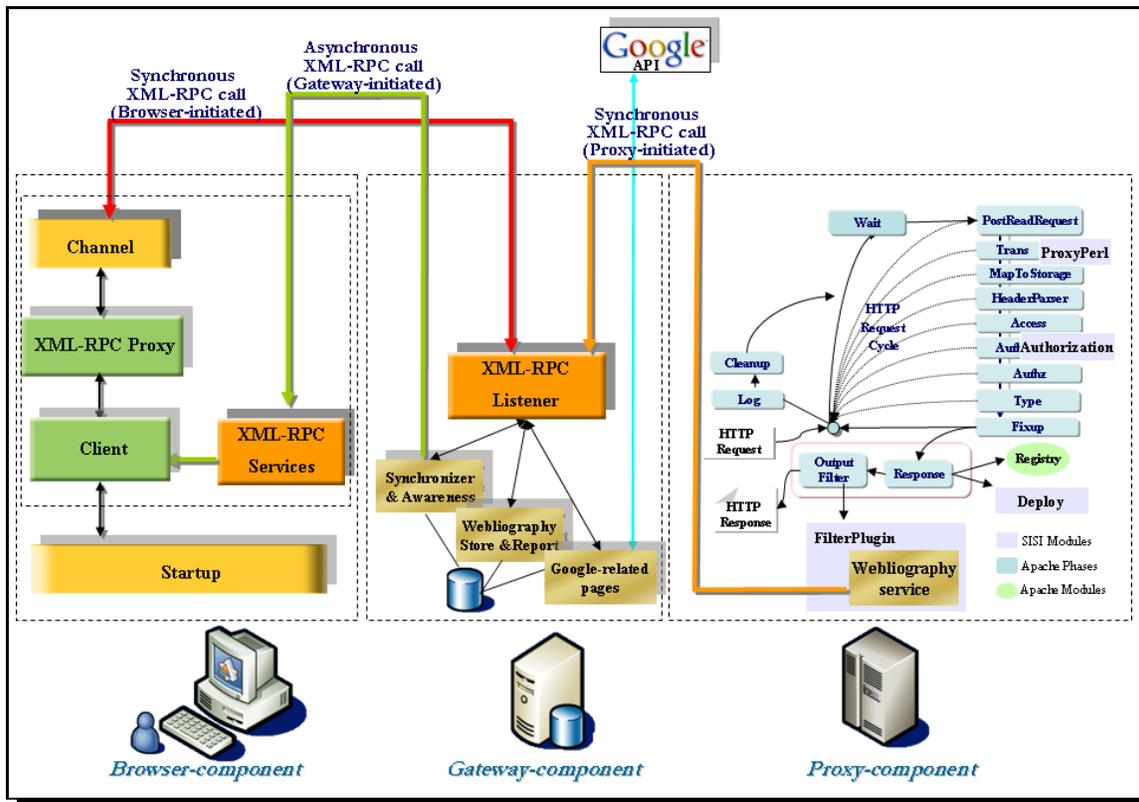


Fig. 2. SYNCBOOK's components, in detail. For more details on SISI architecture (depicted to the right) see [9]; in our architecture, the *Proxy-component* is realized by the *Webligraphy service* that is based on SISI and Apache architecture.

that maintains information about users logged into the system, saves bookmarks added to the Webligraphy, acts as intermediary between the browsers and the proxy components and provides mechanisms for collaboration. Finally, the Proxy-component is written in Perl but is executed on a programmable proxy (SISI [9]) that is built on top of Apache with `mod_perl`; it is able to provide back the awareness of users' actions and to adapt the HTML pages in order to provide recommendations.

Communication among these heterogeneous components occurs via the XML-RPC protocol. XML-RPC⁹ encodes the requests into XML and send them over a standard HTTP connection to a server or a listener component. The listener decodes the XML, executes the requested procedure, packages up the results in XML and returns back them toward the client. The client decodes the XML by converting them into its own standard language data types.

XML-RPC is particularly suitable for our application since it is a lightweight protocol that allows communications between components written in different programming language and operating system platforms. In particular, the choice of XML-RPC (with respect to SOAP and Web Services, for example) was made by considering the light workload imposed on the *Gateway-component*, both from the computational and from the deployment/management point of view (see our

comments in the Conclusions).

In our architecture, within SYNCBOOK, the XML-RPC protocol is used for the following communications (see Fig. 2): (a) remote synchronous invocations by C# *Browser-component* on the Perl *Gateway-component*; (b) remote asynchronous (i.e. no return value) invocations from the Perl *Gateway-component* and the C# *Browser-components*; (c) remote synchronous invocations from the *Proxy-component* toward the Perl *Gateway-component*.

1) SYNCBOOK Gateway-component: First of all, the *Gateway-component* deals with users login/logout, with the awareness of their presence (list) and provides also a simple chat. Then, it acts as broadcaster for the synchronization of the actions each user performs: it deals with bookmark insertion/deletion in the treeview that is used to show the Webligraphy to each user, annotations management and synchronized navigation ("Follow me"). It also provides the local storage for the Webligraphy as well as the final report building both in HTML (for printing, exporting and analysis) and in XML (to enable further sessions of SYNCBOOK on the same Webligraphy).

The *Gateway-component* is in charge of evaluating also the "related" pages (recommendations) that will be provided to the *Proxy-component* on demand by using Google APIs to obtain a list of related URLs. In fact, Google offers (as search engine but also through its APIs) several special query terms

⁹<http://www.xml-rpc.net>

that allow, for example, to retrieve what Google indicates as **Related Links** by adding the prefix “related:” to the query URL.

The *Gateway-component* retrieves the related Web pages for each URL that is inserted into the Weblibliography and stores the first ten results.

From the *Proxy-component*, this component also receives information on the URL requested by each user, in such a way to provide to *Browser-components* the awareness of the number (shown on the client-side with an icon) and the names (shown with a tooltip) of the users that are currently accessing a URL in the Weblibliography. Finally, by the proxy, the *Gateway-component* receives the instructions to highlight the position of a URL in the Weblibliography for a given user.

2) *The SYNCBOOK Browser-component*: To implement the *Browser-component* we use the XML-RPC.NET library, that is a .NET class library for implementing XML-RPC clients and servers in the Microsoft .Net environment.

This component is an Explorer Bar of Microsoft Internet Explorer, that is a child window within the main IE window, and can be placed horizontally or vertically and called from View/Explorer bars menu. Explorer Bars are a particular type of BandObjects¹⁰ that are Component Object Model (COM) objects, where the COM technology dictates which standards use to implement IE components. The main advantages by using COM objects are their dynamic linking, their self-versioning, and code-reusability.

The *Browser-component* is composed of two important modules, the first one (called “Startup” in Fig. 2) adds an Explorer Bar into the Internet Explorer browser, while the second one provides facilities for establishing a remote communication with the *Gateway-component* and can be subdivided into Client, XML-RPC Proxy, Channel and XML-RPC Services.

The main module is the Client, that manages the user’s interactions with the toolbar and, as a result, can invoke XML-RPC services on the *Gateway-component* (which is done by using the Channel and the XML-RPC Proxy modules). It also reacts to the asynchronous invocations by the gateway via its XML-RPC services modules, that acts as a simple XML-RPC server. This communication is used when the *Gateway-component* broadcasts other users’ actions.

Of course, it is necessary that each user configures his/her Internet Explorer browser to use as HTTP proxy the *Proxy-component* of our architecture.

3) *The SYNCBOOK Proxy-component*: SYNCBOOK has been developed on top of the SISI intermediary framework [9].

SISI is a flexible, dynamic and *programmable* intermediary infrastructure that enables universal access to the Web content. This framework has been designed with the goal of guaranteeing an efficient and scalable delivery of personalized services at intermediate edge servers on the WWW.

SISI programmability is a crucial characteristics since it allows an easy implementation and assembling of adaptation

¹⁰http://msdn.microsoft.com/library/default.asp?url=/workshop/browser/ext/overview/explorer_bar_style.asp

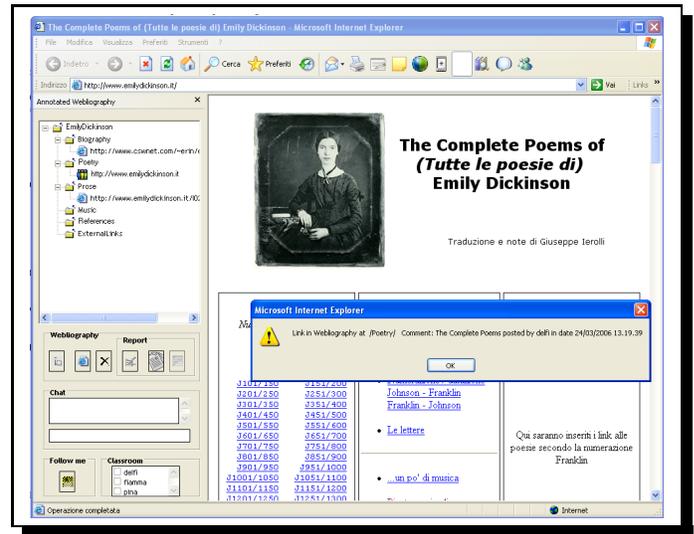


Fig. 3. An example of the alert window for Level 0 Matching.

services that enhance the quality of services perceived by users during their navigation. By simply exploiting the SISI programming model new services can be quickly designed and easily deployed.

The *SYNCBOOK Proxy-component* intercepts any HTTP request coming from the client and, according to user’s request, applies the corresponding actions. In particular, since its main goal is to provide the recommendations (calculated, for efficiency purposes, by the *Gateway-component* each time a URL is inserted in the Weblibliography), for any HTTP request, it adds into the HTML page information about Level 0, 1 and 2 Matchings.

In particular, for Level 0 Matching the proxy adds in the original Web page a JavaScript code to show “onload” (i.e. while loading the page) an alert window that warns the user the URL requested is currently in Weblibliography (see Fig. 3).

For Level 1 and 2 Matching functionalities, the proxy adds JavaScript code to show information about the bookmark in the Weblibliography that is at distance 1 and distance 2, respectively, such as author’s name, data/time of creation and the annotations made by the group.

The JavaScript functions added by the proxy in the Web pages are implemented as .js and .css files (loadable by the *Proxy-component*). The JavaScript code is added with the <DIV> </DIV> HTML tags, and contains invocation of the js functions with the correct parameters. The information is shown through an icon that makes (if clicked) the <DIV> block visible. Within the <DIV> block is also available the [Open Tree] link that allows the highlight of the URL in the Weblibliography, i.e. the Weblibliography is unfolded, in user’s Internet Explorer window, to show the path and position on the bookmark (colored in red) in the tree.

Finally, the *Proxy-component*, by intercepting the URL requested by each user, is in the position to be able to provide group awareness to the *Gateway-component* to be asynchronously broadcast to the clients about the users cur-

rently reading a page. Heuristics are used in order to obtain the information of the “main” URL requested by the user, (i.e. avoiding to consider embedded images, objects, CSS in the HTML page as explicit requests by the user) by using `Referrer: HTTP` header with timing information.

IV. AN EXAMPLE TOUR OF OUR SYSTEM

Here we show an example of usage of SYNCBOOK in the following scenario: the American Literature teacher is asking the learners to find materials on the Web about Emily Dickinson and her work, by preparing a Webliography that the teacher has already categorised in Biography, Poetry, Prose, etc.

At the beginning of the session the facilitator can load the Webliography structure to work as a scaffold for the successive learners interactions. The structure can be created by the facilitator, offline, as an XML file with any editor, and can be provided as a guideline to the learners. As a explicit choice, we decided to disallow learners to create categories in order to focus the collaboration on the content and the topics and not on meta-cognitive tasks. Of course, the teacher can choose to populate the structure with some URLs in order to stimulate coherent and guided search of the material.

The buttons below the treeview offer both the management to the facilitator and the interactions to the learners. The first one allows the facilitator to load a new Webliography (the empty XML structure we described before). This capability is, of course, disabled for learners. Then, there are two buttons: the first allows the learner to insert a new URL (if a category is selected in the treview above on the left) or to insert a comment (if a URL is selected from the treeview above on the left); the second one allows to delete a URL (the operation is allowed only to the author).

Then, there are the three buttons for the Reports: the first one is for showing all the list of comments for a selected URL in the Webliography, the second one for showing the complete list (in HTML) of URL and comments, and the third one allows to download (locally) the XML version of the whole Webliography, once the final report is created. The last two buttons are available only to the facilitator.

First of all, let us deal with the synchronous awareness of other users’ actions. It can be fruitfully used in the educational settings for comparing each others views, such as (for example) the categorization of links. Especially when the scaffolding is not trivial (as in more controversial subjects, like, e.g., a discussion on Genetically Modified Organisms) it is a crucial characteristics for the teacher. In Fig. 3, we show an example of the alert window that occurs for a Level 0 Matching: here the user (named *raffi*) accessed a URL that was already in the Webliography. Notice also that the icon close to the (only) link in the folder *EmilyDickinson/Poetry* shows that more than one user is accessing the link.

In Fig. 4, we show the example of the highlight action. Here the user is accessing a page that has a Level 1 matching, i.e., the page contains a link to a page that was found in the Webliography. By accessing the additional information

embedded by the *Proxy-component* in the HTML source, the learner can read the comment(s) added to the link. Moreover, it is possible to get information on the categorization provided by the author of the insertion by clicking on the [Open Tree] link which makes visible the category (folder) where the URL is located. The URL is revealed (if necessary) in the Webliography by unfolding all the folder from the URL to the root and coloring it in red/italics. We remind the reader that this is performed by intercepting the action by the *Proxy-component* that passes it to the *Gateway-component* that produces the effect on the specific (i.e. learner’s) *Browser-component*.

Finally, in Fig. 5, we show an example of all the functionalities at once: on the Webliography treeview, one can notice a URL that is accessed by many (i.e. more than one) users (i.e. <http://www.emilydickinson.it>) and a URL that is currently read by one user (named “*delfi*” since by moving the mouse, the name is shown on the tooltip. Two additional windows are shown (if user clicks on the images close to links at distance 1 or 2) that give comments, date/time of insertion, highlight, etc. The second one is a related link (by Google) and provides a recommendation to the learner.

V. CONCLUSIONS

SYNCBOOK’s objective is to allow synchronous, interactive and recommended collaboration over the construction of an artefact (be it a WebQuest or an annotated Webliography). In a sense, it should be seen as a tool for collaboratively explore the Web for resources, that offers smooth collaboration, guidance and persistence (the output can be reused for successive sessions in the classrooms).

The system, as described here, shares some interesting characteristics. Our architecture is *distributed* and *efficient*: the workload on each component (both computationally and as management) is moderate. The proxy can be shared by many different schools/teachers and can be placed even remotely by the classroom’s local network, where, on the other hand, the *Browser-* and *Gateway-component* should reside. Moreover, the components it is based on are efficient and stable (see, e.g., SISI for the proxy [9]).

Our architecture is also strongly *modular*: one can use only the collaboration mechanisms and the social bookmarking (but no recommendations) by installing only the first two levels of the architecture (i.e. *Browser-Component* and *Gateway-Component*). In fact, the communications between the proxy and the gateway are always initiated by the proxy side. This can be particularly useful if schools cannot manage to install and maintain an Apache Web Server (needed for the *Proxy-component*), while the other two components do not require much work for installation and management. In particular, care was taken to ensure that the *Gateway-component* can be deployed without particular computer science skills.

SYNCBOOK addresses important issues that are strongly perceived as critical by the educational community. In fact, the success of WebQuest as a methodology is witnessed by the interest it raises in all kind of schools and on heterogeneous

arguments. One of the many portals available (www.webquest.org) has a significant amount of material (tens of thousands of WebQuests have been created by teachers all over the world). The availability of our tool SYNCBOOK makes easy and smooth the implementation in class of such methodology.

About the next steps for the implementations, some efforts will concentrate on improving the synchronization of the “Follow-me” functionality, by synchronizing not only the URL navigation but also the scrollbar (up/down) and discussions of the whole final report by the whole class. Another important step will probably be enhancing SYNCBOOK with the opportunity of adding different roles as well as split the classroom in subgroups, whose findings are to be joined later. In general, functionalities that will make easier the class management will be considered, after a beta-testing in schools.

Finally, it will probably be included SYNCBOOK’s integration with a local Web server (e.g. the Apache Web Server that is underlying the SISI framework that supports the *Proxy-component*), in order to provide a centralized repository of Webliographies within a larger community (i.e. a school).

Of course, the porting of the *Browser-component* in a AJAX (Asynchronous Javascript and XML) environment would add complete portability to this component. Being the implementation as it is, in fact, SYNCBOOK is somewhat limited to work only on Internet Explorer browser, although it must be said that it is between the 50% to 80% of the browsers currently used, depending on the statistics.

Finally, our intention is also to explore the possibility of introducing SYNCBOOK within interactive argumentation-based learning scenarios, such as [3], [27], [30] by using it as a preliminary phase of material search, before the debate.

Acknowledgments: The authors gratefully acknowledge useful discussions with Antonio Iannaccone and Beatrice Ligorio on the goals of the systems and their educational purposes. Fruitful discussions with the partners of the EU-funded LEAD Project (VI framework) are also thankfully acknowledged. The authors also thank Gaetano De Girolamo for collaborating (at an early stage) at the implementation of a part of the cooperation components.

REFERENCES

- [1] Frontier-RPC-0.07b4. <http://search.cpan.org/~kmacleod/Frontier-RPC-0.07b4/>.
- [2] R. Alton-Scheidt, J. Ekhal, O. van Geloven, L. Kovacs, A. Micsik, C. Lueg, R. Messnarz, D. Nichols, J. Palme, T. Tholerus, D. Mason, R. Procter, E. Stupazzini, M. Vassali, and R. Wheeler. SELECT: Social and Collaborative Filtering of Web Documents and News. In *Proceedings of the 5th ERCIM Workshop on User Interfaces for All: User-Tailored Information Environments*, November 1999.
- [3] J. Andriessen, M. Baker, and D. Suthers. Argumentation, Computer Support, and the Educational Context of Confronting Cognitions. In *Arguing to Learn: Confronting Cognitions in Computer-Supported Collaborative Environments*, pages 1–25. Kluwer academic Publisher, 2003.
- [4] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, 1997.
- [5] M. Biddulph. Introducing delicious.us. <http://www.xml.com/pub/a/2004/11/10/delicious.html>.
- [6] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.
- [7] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource list compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*, 1998.
- [8] H. Chandler. Concept mapping & WebQuests in Social Studies. *Media & Methods*, 39(3), 2003.
- [9] M. Colajanni, R. Grieco, D. Malandrino, F. Mazzoni, and V. Scarano. A scalable framework for the support of advanced edge services. In *Proc. of HPCC 2005*, September 2005.
- [10] G. Cordasco, V. Scarano, and C. Vitolo. A P2P Distributed Adaptive Directory. In *Proc. of Adaptive Hypermedia 2004 (AH04)*, August 23–26, 2004.
- [11] Digest of Internet Information, Groups and Other stuff. <http://www.diigo.com/>.
- [12] B. Dodge. WebQuests: A Technique for Internet-Based Learning. *Distance Educator*, 1(2):10–13, 1995.
- [13] B. Dodge. FOCUS: Five Rules for Writing a Great WebQuest. *Learning & Leading with Technology*, 28(8):6–9, 2001.
- [14] J. Dron, C. Boyne, and R. Mitchell. Footpaths in the Stuff Swamp. In *WebNet 2001*, 2001.
- [15] J. Hill and C. Gutwin. The MAUI Toolkit: Groupware Widgets for Group Awareness. *Computer Supported Cooperative Work*, 13(5-6):539–571, 2004.
- [16] Jill Freyne and Rosta Farzan and Peter Brusilovsky and Barry Smyth and Maurice Coyle. Collecting Community Wisdom: Integrating Social Search & Social Navigation. In *Proceedings of International Conference on Intelligent User Interfaces (IUI07)*. Honolulu, Hawaii, USA., January, 2007.
- [17] R. M. Keller, S. R. Wolfe, J. R. Chen, J. L. Rabinowitz, and N. Mathe. A bookmarking service for organizing and sharing urls. *Comput. Netw. ISDN Syst.*, 29(8-13):1103–1114, 1997.
- [18] R. Kelly. Working with WebQuests: Making the Web Accessible to Students with Disabilities. *TEACHING Exceptional Children*, 32(6):4–13, 2000.
- [19] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.
- [20] J. Kurhila, M. Miettinen, P. Nokelainen, and H. Tirri. EDUCO – A collaborative learning environment based on social navigation, 2002.
- [21] A. Lamb. Extreme thinking: Transforming traditional student projects into effective learning environments. *Educational Technology*, 43(4):31–40, 2003.
- [22] A. Lamb, N. Smith, and L. Johnson. Wondering, wiggling, and weaving: A new model for project- and community-based learning on the web. *Learning and Leading with Technology*, 24(7):6–13, 1997.
- [23] A. Lamb and B. Tcelehaimanot. A Decade of WebQuests: A Retrospective. *Educational media and technology yearbook*, 30, 2005.
- [24] J. Lave and E. Wenger. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, 1990.
- [25] J. Lonchamp. Supporting synchronous collaborative learning: a generic, multidimensional model. *International Journal of CSCL*, 1(2), 2006.
- [26] D. Millen, J. Feinberg, and B. K. I. W. R. Center. Social Bookmarking in the Enterprise. *ACM Queue*, 3(9), November 2005.
- [27] M. Overdijk and W. Van Diggelen. An Orientation towards Social Interaction: Implications for Active Support. In *Conference in Artificial Intelligence in Education (AIED 2005) - Young Researcher Track Proceedings*, pages 131–136, 2005.
- [28] J. Rucker and M. J. Polanco. SiteSeer: personalized navigation for the web. *Commun. ACM*, 40(3):73–76, 1997.
- [29] L. Stoilova, T. Holloway, B. Markines, A. Maguitman, and F. Menczer. Givealink: Mining a semantic network of bookmarks for web search and recommendation. In *In Proc. of Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD-2005)*, August 2005.
- [30] M. Van Amelsvoort and L. Munneke. Computer Supported Collaborative Argumentation-based Learning: How students solve controversy and relate argumentative knowledge. In *Conference in Artificial Intelligence in Education (AIED 2005) - Young Researcher Track Proceedings*, pages 9–14, 2005.
- [31] L. Vygotsky. *Mind in Society*. Harvard University Press, 1978.
- [32] G. K. Zipf. *Human behaviour and the Principle of Least Effort*. Addison Wesley, 1999.

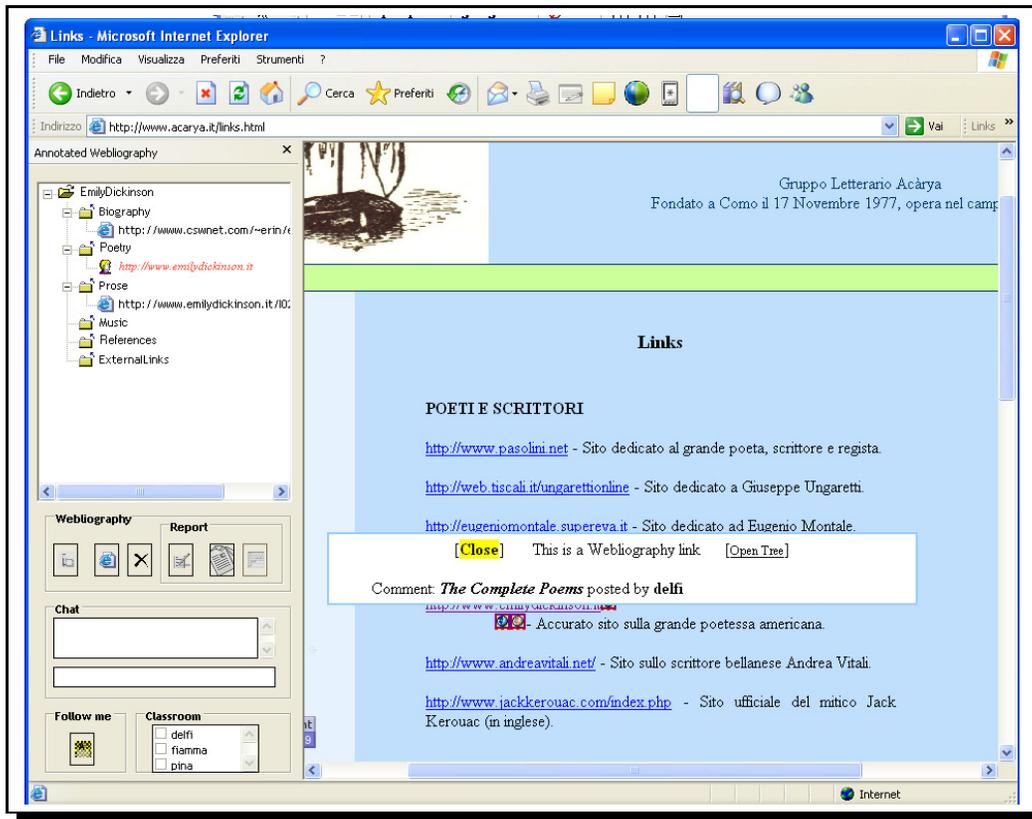


Fig. 4. An example of the highlight action: the link in red in the Webiography is easily identified as the Level 1 matching for the accessed URL.

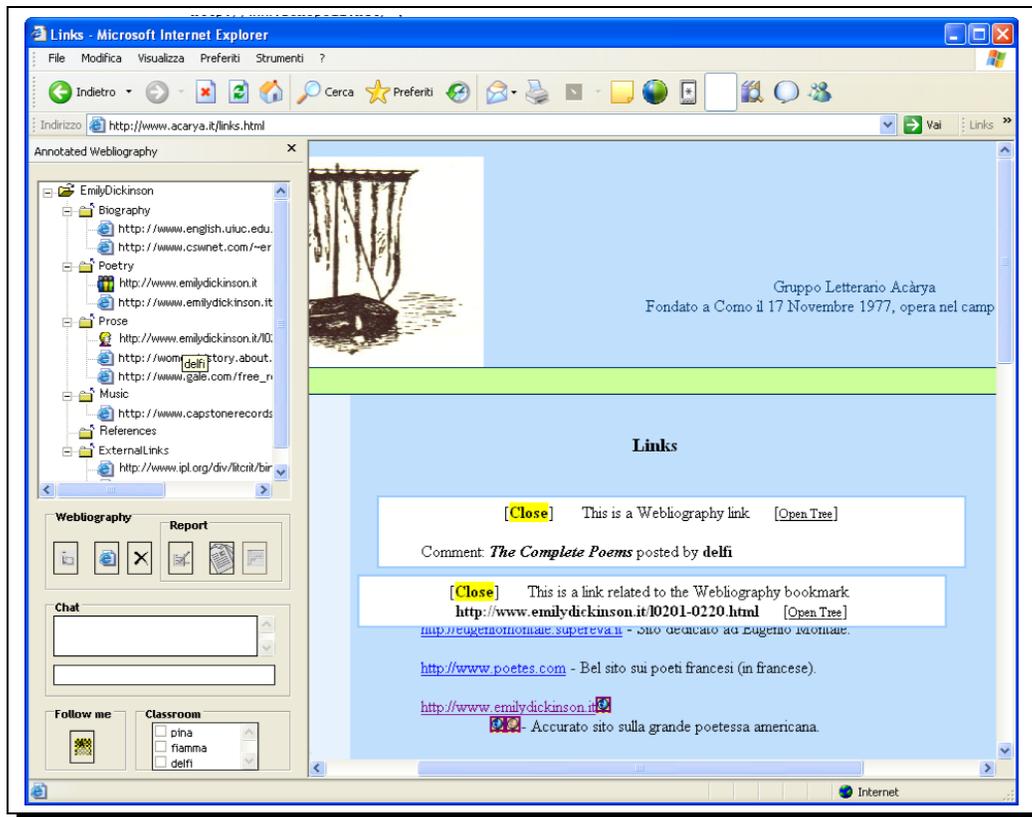


Fig. 5. An example of all the functionalities at once.