

Click, Click – and an Educational Presentation is Available on the Web

Christoph Trattner
Institute of Information Systems and Computer Media
Graz University of Technology, Austria
ctrattner@iicm.edu

Denis Helic
Institute of Knowledge Management
Graz University of Technology, Austria
dhelic@iicm.edu

Petra Korica-Pehserl
Institute of Information Systems and Computer Media
Graz University of Technology, Austria
pkorica@sbox.tugraz.at

Hermann Maurer
Institute of Information Systems and Computer Media
Graz University of Technology, Austria
hmaurer@iicm.edu

Abstract: We present a simple to handle tool that allows users to pick an arbitrary set of Web pages and put them together into a presentation, adding own remarks, changing the order at will, and allowing access to the presentation by the public or just a designated group of users. Although many tools are available for preparing presentations we believe that our tool is the first one that allows sharing between arbitrary users, allows to assign read-only or edit rights individually to users, does not require user-side software, is not violating any copyrights, allows to integrate information from non-public servers, and allows the reorganisation and integrating of own content easily. The tool has obtained the support of school authorities. It will be made available to the international educational community as free application service tool. It is felt that this tool will provide powerful support whenever students are supposed to prepare a presentation on some specific topic.

Introduction

Copy and paste leading to plagiarism is often seen as danger to school and academic work, as many have pointed out in the past [Maurer et al 2006]. Indeed a variety of techniques to combat plagiarism have been developed [Zaka et al 2007] with moderate success, and a strategy to prevent plagiarism but use copy and paste in a positive way has been discussed elsewhere [Kulathuramaiyer et al 2008]. Our new tool is also a step in this direction: rather than condemning copying its intention is to use it in a positive way, at the same time avoiding the violation of copyrights, allowing access to data that may not be available to all and assuring that the sources are always quoted properly.

The tool SIP (SImple Presentation) that we will describe is a server based application module that does not require the installation of any software on the user's PC or workstation. It is thus useable by anyone independent of hardware and operation system, as long as an Internet browser is available. SIP is installed on a publicly accessible server, in our case on www.austria-forum.org a platform that offers a variety of lexica and information items related to Austria. It can be easily installed on any JSP-Wiki Server [JSPWiki 2009]. Indeed it is our plan that after thorough testing, SIP will be made available for anyone interested. The only important assumption is that SIP is installed on a server (the "SIP server in what follows) where users obtain, by registration, the right to store their own information in a designated place (the "user's area" in what follows) and the right to define an arbitrary user group. Such a group could be the general public, a designated group of persons (like a class in a school, a group of friends working together, etc.) or a single individual.

It is our belief that pupils and students should be encouraged to use Web content for preparing materials for presentation or summaries in an open fashion, rather than using semi-legal and semi-moral copy and past techniques.

This is what SIP is for. We do hope that rather sooner or later we will be able to live the slogan: “Maybe you need a SIP!” ☺

SIP is different from PPT’s and similar presentation tools [Presentation SW Survey 2009] by allowing, just through browsing, incorporating arbitrary Web pages in a way that does not violate any copyright. It differs from recording software such as e.g. Camtasia [Camtasia 2009] by not requiring any client-side installation and that the recording is done selectively.

Approach - How SIP works

We distinguish three parts in this discussion:

- the preparation of a basic set of material, i.e. the collection phase
- the editing of what has been collected, and
- the perusal of the presentation sequence obtained.

The beauty of SIP is not just that data collection and editing is made easy, but that a presentation can be produced collaboratively, and modified at any stage, by the original producers or (if permission is granted) by others. Thus, a basic presentation may well lead to a “fan-out” of a number of presentations based on a common core.

Collection phase

Let us start with a simple minded explanation that will be refined afterwards. The user logs in on the SIP server, and navigates to the SIP application. Starting the collection phase application, the user defines a name for the presentation planned. SIP makes a button with two clickable entries “Record/ Stop” available. The user is now allowed to browse the SIP server and arbitrary other servers on the Web. When a Web page of interest is detected, the user clicks on the “Record” entry, thus adding the document shown to what has already been compiled. When the user hits the Stop entry a question pops up where to store the material collected, typically in an area “Presentations” that has been defined earlier as part of the user’s area.

After this, the user assigns rights to work with the collection just stored to some user group (typically, access rights to the general public or to a selected group of users, some even with edit privileges). It is now possible for the user and all others with editing privileges to add further material, to rearrange, annotate and edit the material in various ways, and to eventually convert it into a final form, ready for perusal. Before we go into more detail concerning editing and using the product we want to warn that the collection phase is much more tricky than it sounds on a first glance.

Note that to make the “Record/Stop” entry available throughout while browsing other Websites, the SIP application acts like a browser, i.e. any Web page accessed via a URL specified is not displayed by the standard browser, but via SIP. This allows SIP to store a list of URL’s, the complete content of what is available at that URL and a check-sum to later determine if the content has changed. SIP also produces for each HTML page a thumbnail and allows the user to give it a name to assist in a later editing process. Each Webpage recorded can be assigned the option “display as currently seen” or “display as currently seen followed by what is seen at this URL when the presentation is used”. If none of those two options is chosen then at presentation time the then visible contents of the URL is shown. Note that those options are particularly critical in connection with dynamically generated HTML pages, or frequently changing pages as would be the case e.g. on a news-server. Observe further that to always show pages recorded on the SIP server would infringe on copyright, and is harmful to the site the URL points to, if that site depends e.g. on advertisement revenue based on page requests. SIP therefore always tries to access URLs at presentation time, stored contents is only used as back-up. However, even if this happens SIP makes sure that a page request is sent to the URL at issue so that the number of page impressions is the same, whether information is fetched from the SIP archive or from the server on training the originally recorded information.

There is still one other crucial point to be observed: some information collected may come from servers where only registered users have access too, or where a fee has to be paid in some form when accessing such server. In this case, not only must the user be of course authorized to access the server, but at presentation time each user must have that privilege. Hence the necessary access control information also has to be stored by SIP for such pages. Although the SIP archive contains such protected pages it must not show them unless the user has the authority to access the server

where the pages come from. We will discuss this problem a bit more in detail in Section 2.3, i.e. when explaining the presentation phase.

Editing phase

After material has been collected in the collection phase, users privileged to do so can now edit the material in a variety of ways. The most basic options are to rearrange the order of slides (done by moving the thumbnails shown), by deleting slides, by adding comments to pages that are shown in the presentation phase, or by re-opening the collection phase to add further material, material that may also have been created on the SIP server for that very purpose.

The editing phase in the final version of SIP must also include further options. One is to show only parts of an HTML page: note that again the issue of page impressions at, or authorization for, the original site has to be handled as if the full page would be displayed. It is clearly also desirable to be able to add comments not just at the end of a page, but to overlay them on a page in a number of ways: like highlighting certain parts of a page, using arrows pointing at something important, in both cases allowing to show some written comments (or even allowing spoken comments). Clearly, the overlaying of information is more tricky than just adding something at the bottom of a page.

Future efforts also must be extended to handle non-HTML material. Suppose a Web site offers a couple of PDF files to be displayed. Comments or highlighting are not trivial tasks any more in such a setting. The situation becomes even more complicated when a video clip is shown and some written synchronized comments are desirable or even a synchronized audio-file is desirable, maybe even suppressing an original sound track!

Thus it should be clear that what may have sounded like an easy to implement tool is turning into quite an undertaking if all one wishes for is taken care of. At any rate, after all the editing is complete the presentation is ready to be used. The technicalities involved in this presentation phase are also more than trivial as will become clear in what follows.

Presentation phase

A user interested in showing a presentation compiled by SIP has to have all rights necessary: not just the access rights to the presentation on the SIP server, but also other rights, if the presentation includes material coming from servers that need special authorization, as is often the case with professional digital libraries or material used in school from school book publishers.

The “simple presentation mode” allows with each key-press or mouse-click to advance to the next page. Each page is associated with a URL. Standard procedure is to request the corresponding page. However, a number of special cases have to be considered.

- (i.) The server does not reply within a time out set by the user. In this case, the materials from the SIP archive are shown but with a warning of the type “The URL can currently not be accessed. Please make sure to check up on this after the presentation”. The reason for using the material from the archive is to make sure that a presentation that was tested some time before it is shown does not suddenly let down the presenter. Yet the message is necessary to show that corrective action is necessary, else a clear infringement on copyrights is happening.
- (ii.) The server does reply, but the check-sum in the SIP archive differs from the check-sum now calculated. This will happen quite regularly with dynamically generated HTML pages, or pages that by their very nature (e.g. news coverage) change frequently. Observe that already in the collection phase this has been possibly taken care of by automatically showing old and new page, or just the new page, or just the old page. The user of the presentation has the privilege to overrule the decision and choosing one of the options. The one that is critical is of course the option “show original page from archive”. In this case a warning similar to what was mentioned in (i) is issued, but also a request for the URL is sent without displaying what is obtained (!). We have resorted to this trick so that the owner of the URL gets a page impression (or some royalty if this is a license protected site) even if nothing from that site (except what has been archived) is shown. In this way we are trying to ensure that a presentation will not fail unexpectedly, yet we sites

used in a presentations get what they deserve (page impressions or royalties) even if their material cannot be used in an ongoing life presentation.

We have only discussed the “simple presentation mode”. It is also possible to improve navigation. One is by numbering the pages in a presentation sequence and allowing to choose an order different from the standard one see Fig. 1.

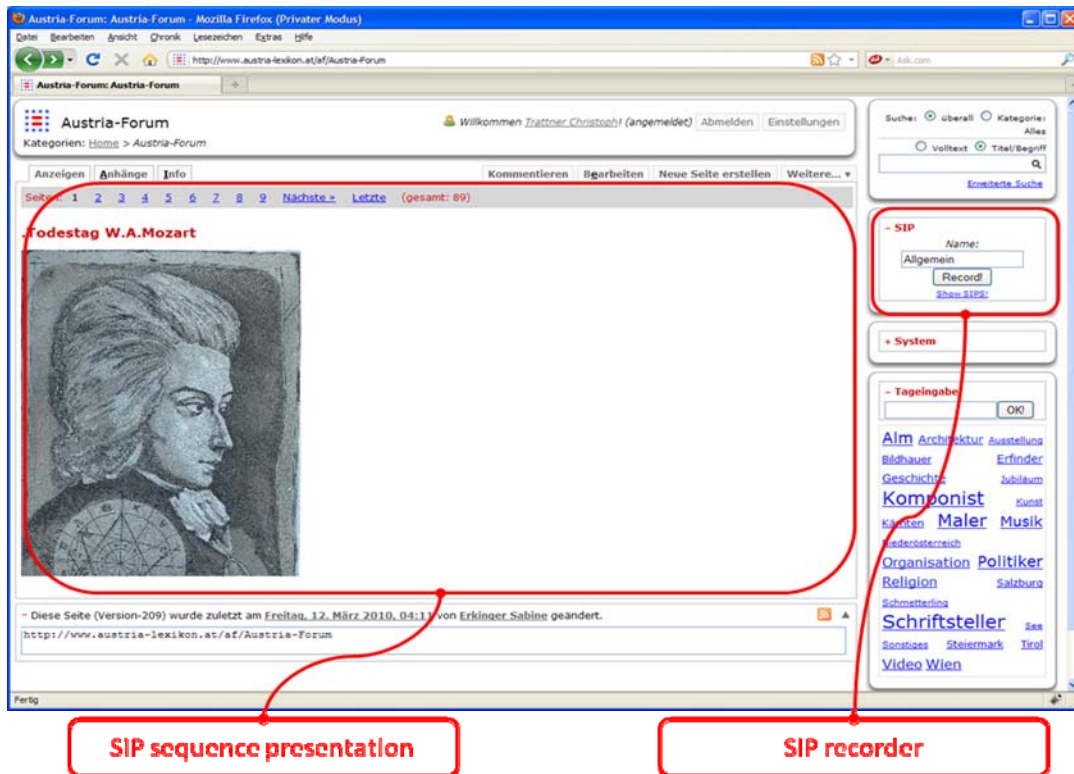


Fig. 1: Sequence based navigation during presentation phase

The second is to use the thumbnails mentioned for navigation. By clicking at one, the corresponding page is shown, a single click leads back to the thumbnails. This is shown in Fig. 2.

Implementation

As described in the introductory part of this paper, SIP is designed to run as a Web application with any Web browser available, i.e. independent from any operating system and applicable wherever an Internet connection is available. Since SIP is also meant to be a collaborative tool, we designed it to be installed as a Web service on JSPWiki based systems, as a first step, i.e. it is planned to make the system also available for other Wiki based systems, such as MediaWiki [MediaWiki 2010] or PHPWiki [PHPWiki 2010] in the future.

For a technical perspective, SIP can be divided into five core modules, which are the following:

- **Collection Module:** This module consists of two sub-modules: a client- and server sub-module. The client sub-module consists of a two (X)HTML elements: one (X)HTML form input field for annotating a SIP with a certain name and one (X)HTML input form button to start/stop a SIP recording. Furthermore the client part is implemented as a JSPWiki plug-in, to make integration at any point of the system easily. The server side sub-module is a Web service. It processes HTTP requests from the client part and stores them into a data storage module. Since the collection module can, simply speaking, be seen as collection module of a basic tagging system, where a bookmark B is stored as a 3-tuple in the form $B = (\text{Tag}, \text{URL}, \text{Time})$, we

have chosen an OpenSource and mySQL based tagging database called Scuttle [Scuttle 2010] for storing the information provided by the client side part of the collection module. Since the client module produces a new HTTP request whenever a page is viewed and the record button is active, and since the outer frame of the application should not be reloaded, communication to the server side part is done asynchronously via AJAX.

- **Caching Module:** In order to “persistently” store a bookmarked page within a SIP, i.e. in order to not break a SIP during presentation mode, the server side part of the collection module additionally provides a post-processing module which caches the whole content of a particular bookmarked Web site. The caching module is implemented as Web service that caches a content page only on demand. Thus, whenever a page is requested by the presentation module the caching module checks the checksum stored by requesting the stored URL for this site and storing it only in that case the content of the document has changed.

In order to store the whole Web content of a certain page the caching module implements two modules: a (X)HTMLParserModule and a FileProviderModule. The (X)HTMLParserModule parses the (X)HTML content of a page and extracts and downloads addition media-files such audio, video or image files. The FileProviderModule stores the content of a page on file system layer in a flat directory hierarchy, i.e. each page including all media files are stored into one separate directory (meta-data is stored into a XML based properties file in the same directory). This simple but effective storage strategy makes it possible to save even any kind of file format other than (X)HTML such as PDF for instance.

Since a thumbnail presentation of a particular SIP page is required during the editing phase, the caching module generates a thumbnail for any cached Web site as well. Since some pages may not contain an image file at all (an audio files for instance), particular replacement icons are used instead.

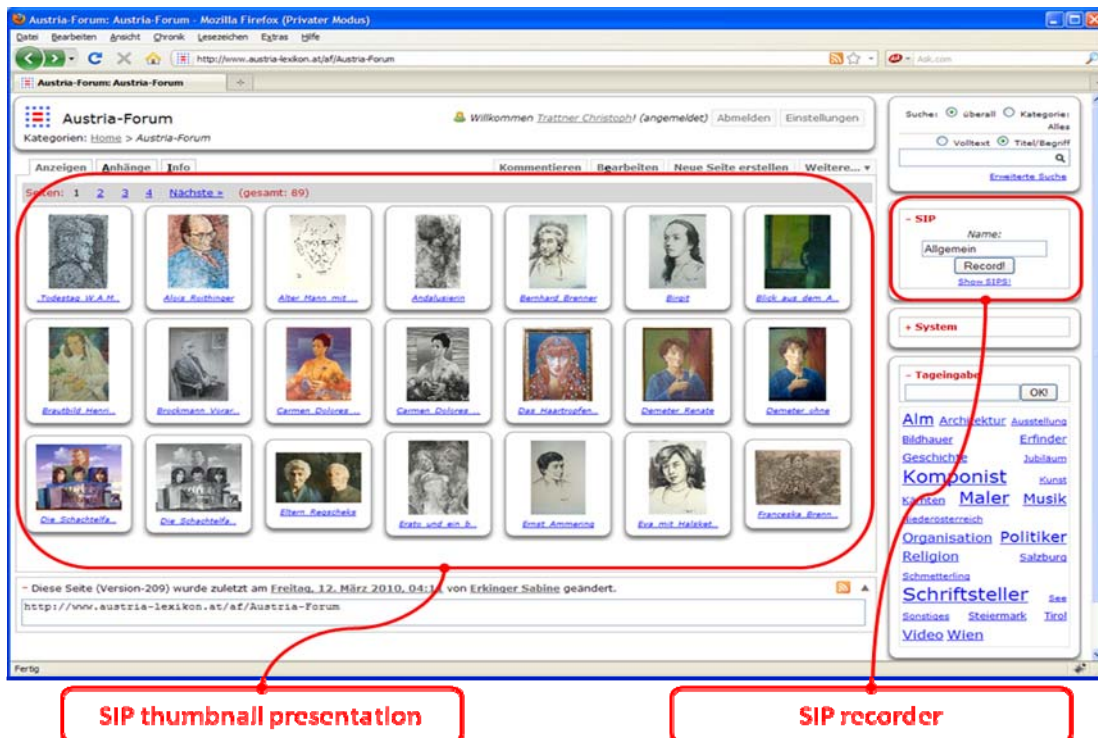


Fig. 2: Thumbnail based navigation during presentation phase.

- **Edit Module:** In order to modify a recorded SIP a special edit module is implemented. Basically this module can be seen as a simple extension of the (X)HTML editor called CKE [CKEditor 2010] editor

within JSPWiki system. The main difference compared to the basic version is that the editor offers a special “SIP-view”. This makes it possible to rearrange a particular page within a SIP via drag&drop. This feature is implemented via the Yahoo! JavaScript library [YUI 2010]. Of course one can delete or edit a single page as well. In order to edit a SIP one simply navigates to the SIPs category within the JSPWiki system and clicks on a SIP hyperlink to change into presentation mode.

- **Security Bridge Module:** Since SIP is intended to work also in an environment that is hidden from the general public, i.e. digital libraries for instance which require explicit authentication, SIP implements a special security bridge module. This module is intended to work as a security container for any site requiring explicit authentication. Generally this module is deactivated and provided as an addendum. We do not want to go into more details regarding this module in this paper.
- **Presentation Module:** The last, but maybe most important module within SIP is called the presentation module. As already described in the introductory part of this paper, this module is implemented to work as a browsing module during the collection phase and as browsing/presentation module during the actual presentation phase. Technically, this module consists of three basic elements. The first one is the URL form input field including a “go” button for requesting a certain page within SIP during the collection phase via JavaScript. The second element is the IFRAME for displaying the actual requested content of a page. The third component is the navigation bar implementing two JavaScript based back and forward buttons.

During the presentation mode, one can choose between a simple presentation mode (see Fig. 1) and a thumbnail presentation mode (see Fig. 2). Additionally the navigation bar disappears if loading a SIP.

Future Work

We want to conclude this paper by discussing some other important aspects which arise with a broader usage of SIP. Web 2.0 is all about different possibilities for collaboration and this should be considered when further developing the SIP tool. In the first section we have mentioned that in the collection phase users assign rights to material produced and rights to edit the collection of slides stored on the SIP server. Rights may be defined for some user group, e.g. for the general public, or for a selected group of users. If users decide to assign editing rights for a presentation one should also discuss the scope of rights assigned. Must users assign them for the whole presentation, or is it possible to give the rights to edit on a single slide basis, or even on an object basis – for example a part of the slide containing some special information? This would also mean that SIP needs a sophisticated and granular user and object rights model.

With collaboration the question of versioning arises immediately – should users check out the presentation when they are editing it? Furthermore, SIP could be extended to support the possibility of simultaneous editing inside the defined user’s group. Simultaneous editing is supported by state-of-the-art presentation software like Google Web Apps or Microsoft Office. Having this functionality would be very useful when working with colleagues on a shared task.

Often, users tend to take out slides out of their existing presentations when creating a new one. This could be solved by enabling each user to have a slide library consisting of only single slides. Such a slide library could of course be extended and SIP could enable users to incorporate or edit slides that people in the user’s group have provided. Here we could utilize some Web 2.0 functionality like linking to friends (or the user’s group) or simultaneous editing as mentioned above. A different approach to this would be that in the future users could upload single slides or a whole existing presentation (for example the PPT or PPTX file) that users have saved on their hard drives. The SIP could also be expanded to enable usage of other files users have on the hard drive and which are often used for presentations, like images, videos, documents (for example DOCX or PDF files). This would, in our opinion, add even more usability to SIP.

Another important aspect is the usage of existing presentations from the Web which are usually in PDF or PPT file format. We should discuss the best approach when dealing with this kind of presentations. Is it better to record them directly from the Web or should they first be downloaded to the user’s hard drive and then uploaded as discussed above? The idea of recording is particularly interesting in this context due to the fact that with the newest Microsoft Office version so called “Office Web Apps” have been introduced which offer functionality to view Office documents without client – just via browsers. Maybe SIP could use this functionality to display existing presentations on the Web. How to deal with copyright concerning this kind of slides will still have to be investigated.

A different important question for the future of SIP is whether SIP should be further developed so that users can make their own slides with their self-generated content - for example create new slides and content like charts. In this case SIP should offer some basis content generation functionality like fonts, charts, shapes etc.

Summary

We have briefly presented a new tool SIP that allows preparing sophisticated presentations based on material in the Web. The tool is an application service running on some server, i.e. is independent of the user platform (as long as the user has some internet browser available.) It handles copyright issues and access restrictions in a clean way and opens the road to usage for educational purposes in educational institutions and in companies.

References

[Camtasia 2009] http://de.wikipedia.org/wiki/Camtasia_Studio (Last visited December 6, 2009)

[CKEditor 2010] <http://ckeditor.com/> (Last visited March 22, 2010)

[JSPWiki 2009] JSPWiki - A feature-rich and extensible WikiWiki engine built around the standard J2EE components; <http://www.jspwiki.org> (Last visited: November 8, 2009).

[Kulathuramaiyer et al 2008] Learning Ecosystems dealing with the Copy- Paste syndrome; N. Kulathuramaiyer, H. Maurer, Journal for Research in Innovative Teaching, Vol.1., No.1 (2008), 1-25

[Maurer et al 2006] Plagiarisms-A Survey; H. Maurer, F. Kappe, B. Zaka, J.UCS vol 12, no. 8 (2006), 1050 – 1084

[MediaWiki 2010] <http://www.mediawiki.org/> (Last visited March 22, 2010)

[PHPWiki 2010] <http://www.phpwiki.org/> (Last visited March 22, 2010)

[Presentation SW Survey 2009] http://en.wikipedia.org/wiki/Presentation_program, (Last visited: December 6, 2009)

[Scuttle 2010] <http://www.scuttle.org/> (Last visited March 22, 2010)

[YUI 2010] <http://developer.yahoo.com/yui/> (Last visited March 22, 2010)

[Zaka et al 2007] Plagiarism – a problem and how to fight it; B. Zaka, H. Maurer, Proc. of ED-MEDIA 2007, ACCE, USA (2007), 4451-4458