Web Annotation System based on Web Services

Marco Fernandes[1], Miguel Alho[2], Joaquim A. Martins[3], Joaquim S. Pinto[4] and Pedro Almeida[5]

[1]Universidade de Aveiro, IEETA,
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
marcopsf@ieeta.pt

[2]Universidade de Aveiro, IEETA,
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

[3]Universidade de Aveiro, Electronics and Telecommunications Department,
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

[4]Universidade de Aveiro, Electronics and Telecommunications Department,
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

[5]Universidade de Aveiro, IEETA,
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

***Abstract*: Annotation tools allow users to complement existing documents with comments, suggestions or other information. This functionality is very useful in process management or workflow applications, where documents are changed frequently by multiple users. Currently, desktop annotation tools are much more sophisticated than those available for annotating web documents. In this paper we present EspiritUs, a Web Services based annotation system designed to allow users to create, share, and search annotations on the web. We also analyze in detail existing web annotation tools, their functionalities and their limitations. We present the adopted XML technologies based architecture for EspiritUs, we describe the system implementation and present two possible annotation scenarios. Finally, we present some conclusions and guidelines for future work for this project.**

***Keywords*:** Collaboration, Semantic Web, Web Annotation, Web Services**.**

I. Introduction

Annotations allow users to complement the information in a given document, make suggestions or remarks before delivering it to other users, or use them for many other purposes. Most desktop applications that somehow deal with document management have their own annotation system, either to help the person building the document or to share comments between different partners. Some text editing applications such as the Microsoft Office family [1] or Adobe products [2] have relatively sophisticated annotation tools already built-in. Such tools allow users to place free-text annotations on desired locations within the page.

Although this functionality is relatively well developed in desktop applications, it presents one major drawback – the annotations are usually not searchable, either within a document or across several files. Also, annotations are stored in the document itself, so sharing them with other users from an organization or institution requires documents to be sent back and forth, involving a huge overhead, just to share some comments on a given document.

Moving this annotation functionality to a web based system seems the logic way to go. Annotations may then benefit from the ubiquitous nature of the web and constant availability of its contents.

Documents stored and available on the Internet, called web documents from this point forward, may be retrieved using its location – the resource's URL (Uniform Resource Locator). URLs are strings that identify resources on the World Wide Web: web pages, images, services, mailboxes, and other resources [3].

However, if one wishes to annotate a web document, such as a HTML page or a multimedia object stored in a web server, the available web annotation tools are either very limited, do not present real benefits to users, or even have undesirable or unpredictable behaviors. When making annotations to web documents the annotation tools should uniquely associate them to the resources (using its URL, for example), make them available from anywhere with an internet connection, and restrain edit and removal operations to its authors and system administrators. Most tools available fulfill only some of these characteristics.

EspiritUs aims to fill these gaps in the web document annotation area, providing not only an annotation system accessible from the internet but also an application program interface (API) to enhance other systems with annotation capabilities.

Furthermore, each annotation may have two different scopes - public or private - thus being visible to all users or only to its author. Regardless of its scope, annotations may only be edited or removed by its authors.

As it will be described, the presented solution allows the integration and reusability of the system in different scenarios. Two implementation examples will be shown in more detail: a client-side annotation component, which connects to a public server and offers users the ability to view, create and edit annotations from anywhere; and the integration of an annotation component in a web application to allow registered users to annotate the site's contents.

Also, EspiritUs is a tool which helps promote the semantic web and the enhancement of digital libraries metadata. It can,

for instance, be used by librarians or other professional and credentialed users to enrich a digital library by extending basic documents descriptions with relevant and more specific information.

II. Related Work

The development of an annotation system may be divided in the resolution of two main problems: creating the system's core, which must store the annotations, manage the user privileges and provide search mechanisms; and designing and implementing an annotation interface. The following sections describe some of the most import work in both areas.

A. System Implementations

1) Annotea

The most relevant existing project on the web annotation area is Annotea [4], a W3C project that relies on RDF and HTTP to transport annotations. Each annotation is associated with the document URL and stored in a relational database. The project is built under the Semantic Web Advanced Development (SWAD).

Annotea's enhances collaboration not only via shared annotations but also bookmarks and their combinations. Annotations are categorized in comments, notes, explanations and others.

Annotea also uses XML Pointer Language (XPointer) [5] to locate annotations inside the annotated document. XPointer is built on top of the XML Path Language (XPath). The framework is intended to be used on resources with a text/xml or other xml based Internet media type, but can also be used on HTML.

One of Annotea's main disadvantages is the fact that no distinction is made between public and private annotations. Thus, the system cannot be privately used by an organization unless the network where the documents are to be annotated is an intranet. Moreover, it does not seem possible to search for annotations – by contents, date, author, etc. – in the entire repository, which usually is a desired characteristic on most systems.

Finally, tests show that no logical distinction is made between web pages with different querystrings. Consider, for instance, a news web site in which all news are displayed in the page shownews.aspx?news=[newsID] by simply changing the newsID value. By ignoring querystring changes, Annotea does not allow distinctively annotating news from this site, thus making all annotations to shownews.aspx visible regardless of its content.

2) Other systems

Although not as important, there exist some consistent alternatives to Annotea, which implement distinct technologies to accomplish similar results. An interesting one is Web Annotation [6] (WA), which is mainly aimed at the Moodle course management system.

WA uses the Range object [7] (fully available only in Mozilla browsers) instead of XPointer to determine the location of the selected content in a web document. Also, unlike Annotea, it uses Atom instead of RDF.

The biggest disadvantage is the lack of support for the system if not using Moodle. Another considerable drawback in this system is that there is no API or client-side software available.

B. Annotation Interfaces

There are essentially three main approaches when developing annotation interfaces for web documents: develop an application from scratch; integrate the annotation interface and the web documents in a web page; and integrate the annotation interface in the browser itself. In the following subsections we briefly describe some examples and discuss the advantages and disadvantages of each approach.

1) Creating an annotation application

In this solution, the developer creates the desired annotation features and, by creating the application from scratch, has complete control over the interface. Handschuh and Staab's tool [8] and Janno [9] are examples of such applications with annotation interfaces.

Despite the complete control over the interface gained by creating the application from scratch, this solution forces the user to change its web navigation habits and use the given tool instead of his favorite browser, or at least to do so whenever he needs to see or create annotations. Furthermore, users must install the application in all the desired internet access points.

2) Web page integration

A relatively simple solution consists in integrating the annotations and the annotation interface in the web pages. This can be accomplished basically in two ways: creating a web page formed by two or more frames - the visited web document is displayed on the main frame, while the annotation interface is placed on another frame; or merging the web resource and the web annotations in a single document.

The first approach, with a few changes, is adopted by Annotation Engine [10], which divides the web page into two frames. In the main frame the web resource is displayed while on a second frame is placed the annotation interface. When the resource is a web page, what the Engine displays is actually a modified version so that the viewed document includes a visual indicator where the annotation was made. This is accomplished by allowing users to define a unique string of text when creating annotations, and inserting the indicator in the HTML source file (next to that unique string) before showing it. Also, all links are rewritten so that navigation always occurs inside the frameset (from http://www.ua.pt to annotate.cgi?view=http://www.ua.pt, for instance).

Gibeo [11] follows the second approach. If a user wishes to annotate some page at http://site.com, he must type http://site.com.gibeo.net in the address bar. The site is then loaded within gibeo.net domain with all annotations indicated and available in the document. Like Annotation Engine, most links are rewritten and all annotations are public. Also, the tool restrains annotation to web pages, so annotation to content such as multimedia objects is not allowed.

E-marked [12] is a similar system with identical limitations but changes addresses to e-marked.com/nw/tf2.php?urnln=44, which offers non intuitive addresses, thus requiring users to navigate using e-marked web site.

One of the advantages of the web page integration approach is its simplicity (both in development and in usage). No installation is required and users may use their favorite browser. Despite that, the solution alters the browsing experience by changing documents and addresses from the browser location bar, which is a usability flaw. Moreover, the approach is highly fallible: all links are rewritten, but a document change caused by form submission or a flash link causes the process to fail.

3) Integration in the browser

An alternative to the previous approaches consists in integrating the annotation interface in an existing browser. On Internet Explorer (IE), for instance, such areas are called band objects and are integrated in the same manner as the Search and Favorites band.

From our point of view, this is the most transparent and user-friendly solution, since the browsing experience is not altered: the user may use a common browser and documents and addresses are not modified.

Obout Inc. created Memobook [13], a system to store links and annotations available in a web browser. Memobook uses the browsers search band to display the interface and stores the annotations in a central server, thus making them available from any internet access point. However, the annotations are generic and not associated with the viewed document, i.e. all annotations are always displayed, regardless of the current web resource.

Opera browser [14] includes, since its 7.1 version, a panel similar to IE's bands specifically designed to make annotations. However, annotations are not associated with web documents and are stored locally, so available only where the annotation took place.

Wikalong [15] is a browser extension for annotating with Firefox. It has an unintuitive interface and connects to a rather slow system which makes no distinction between public and private annotations and even allows editing on other users' annotations.

Annozilla [16] is an annotation interface to the Annotea server and has essentially the limitations of the underlying system.

One of the most popular web annotation browser's plug-in was the now extinct Third Voice [17]. Third Voice allowed users to select content from web pages and attach comments to it, which would then be visible to all other plug-in users. Third Voice quickly caused the ire of webmasters essentially for three reasons: the plug-in changed the visual aspect of sites, either by placing icons on the page or highlighting text; some users used the plug-in to place improper or offensive remarks; other companies used Third Voice to advertise their sites or products, sometimes comparing it to those on the commented site. Third Voice was eventually discontinued due to lack of financing.

4) Other solutions

Amaya [18] is a complete browsing and authoring environment from W3C. It already supports annotations by integrating Annotea.

World Forum [19] is an effort aimed at providing forums attached to any existing web site. Users add a set of JavaScript instructions to its favorites and run it on a web site whenever they to annotate it.

III. Objectives

As discussed above, EspiritUs' main objective is to create an annotation system which provides some important characteristics missing from the solutions available. The most significant are:

- Allow users to create and edit annotations to web resources from any place with an internet connection and make those annotations available regardless of where the annotations were made.
- Annotations must be unequivocally associated with each web resource. Furthermore, annotations must be searchable.
- EspiritUs' users should be able to create both public and private annotations. Public annotations must be visible and searchable to all users, while private ones only to its authors.
- Unlike Third Voice, Annotation Engine, Gibeo, and others, EspiritUs should not alter in any way the visual aspect of the visited sites.
- The developed system should be easily integrated in more complex systems and be interoperable and platform independent.

IV. Architecture

The EspiritUs' system architecture was designed as depicted on Fig. 1. As can be observed, communication with the annotation server's Web Service is accomplished by using the NoteAccess module, which provides the methods to view, create, edit and search annotations.

On the server side, the CNote module establishes all the required communication with the file system and the indexer. This is the core of the system. The authentication module is responsible for generating unique author identifiers, connecting to the User Information repository (either a collection of XML files or a relational database) and retrieving/updating the information.

CNote also uses URL Parser, which is responsible to parse web documents URLs and provides an URL to repository location mapping (determines where to store annotations and where to retrieve them from). It also helps restraining search operations to a single domain, when required.

Unlike Annotea, writing new annotation applications for EspiritUs is a rather trivial task. All developers must do is call the methods available in the NoteAccess module.

An integration diagram is depicted in Fig. 2. To enhance an application with annotation capabilities, the EspiritUs NoteAccess module should be used to communicate with the annotations Web Service. Whenever a new page is displayed on the presentation layer or an annotation is added or edited, an event should be triggered to request an updated list of existing annotations to NoteAccess.

The NoteAccess module requires only basic knowledge of the annotations structure from developers, since it parses annotation objects to and from XML and makes the communication with the Web Service transparent.

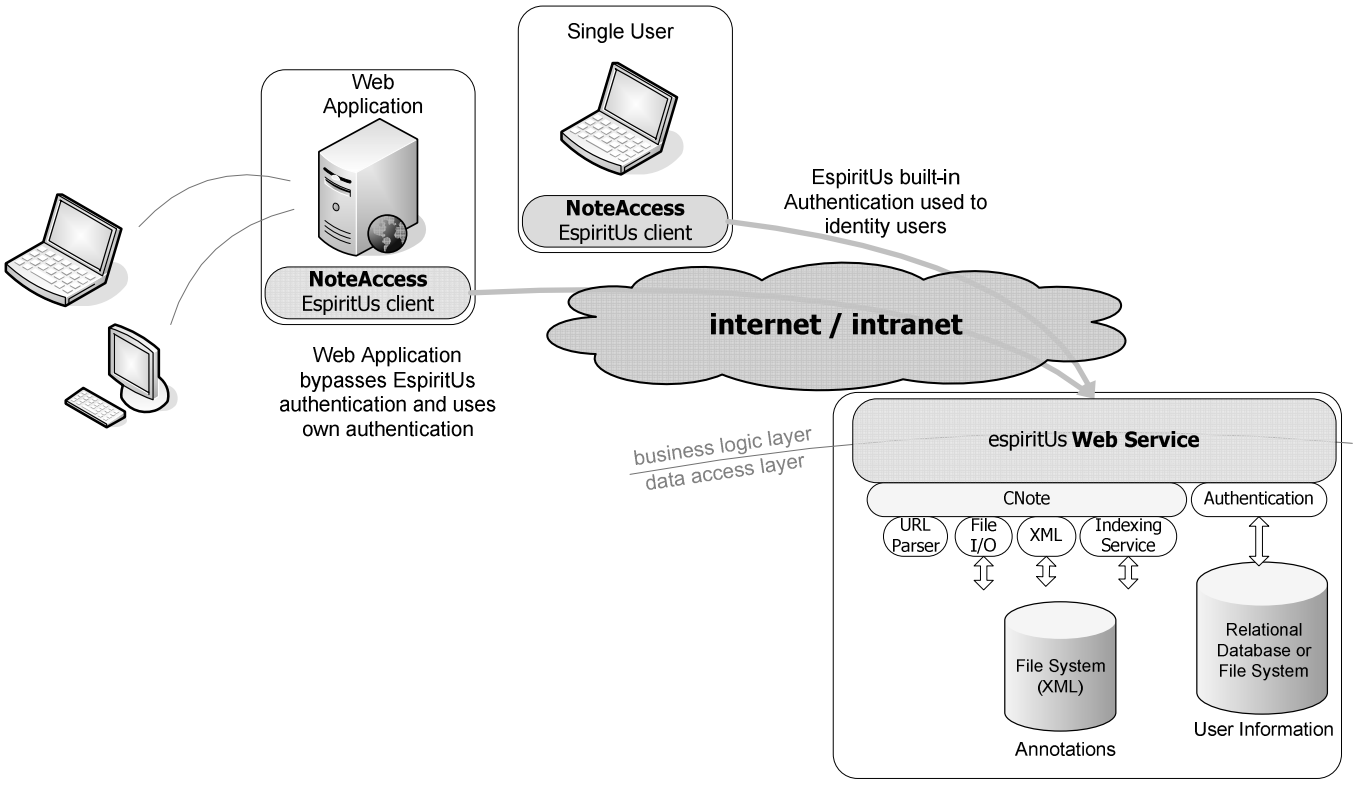

Figure 1. EspiritUs architecture

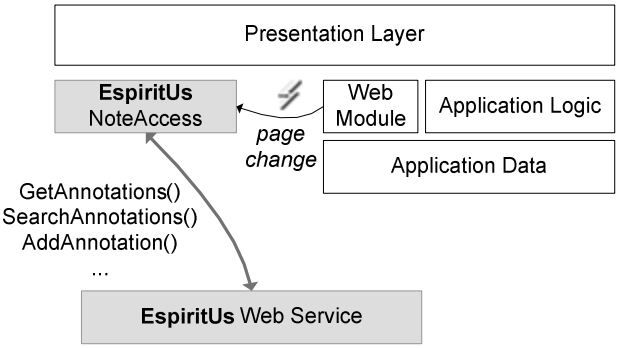

Figure 2. Integrating EspiritUs in an application

V. Annotation Repository

In respect to the data access layer, there were four main issues to address: location and organization of the stored annotations; implementation of search mechanisms, authentication, and availability of data to the upper layers in an interoperable way. The next four sections describe the methodology adopted.

A. Storage

To allow the access to annotations from anywhere, it is necessary to store them in a central server. There are essentially two ways to store them: in a relational database or in the file system.

Relational databases benefit from several years of technology development and consolidation, and have many advantaged over file system databases [20], such as improved security, data integrity, maintenance, backup, and data accessibility through built-in search mechanisms. However, they are more expensive and require more powerful hardware. Also, they aren't fit for storing highly heterogeneous and non-linear data, which is the case of annotations. On the other hand, if a future work requires changing the structure or fields of the annotations, the relational databases do not provide the required flexibility for a smooth transition.

Thus, the decision was made to store annotations in text files. To do so, the XML (Extensible Markup Language) [21] format presents itself as the perfect choice for a highly flexible and interoperable system. XML is the most used format in high interoperable systems to perform communication between different platforms and operating systems and is the becoming widely adopted.

1) XML Annotations

On a preliminary phase, a set of fields were chosen to represent the basic information needed for an annotation: web document's URL, annotation's ID, author, date, scope (public/private) and the annotation's text. The system can be easily extended or modified to support other metadata fields relevant to specific scenarios.

All annotations to each web resource (regardless of the querystring) are stored in one correspondent XML file. To correctly accommodate pages with distinct querystrings, the base URL (the complete address except the querystring) is stored in the outermost XML node and the annotations are stored inside each correspondent querystring node. Consider,

for example, an annotation file for the web site http://site.com:

```
<notes url="site.com/dir/page.asp">
 <query txt="id=31">
  <note id="155" author="Ted" date=".." private="true">
  Annotation test.</note>
 </query>
 <query txt="id=27">
  <note id="753" author="Bob" date=".." private="true">
     Another test made on a different page.</note>
 </query>
</notes>
```

By inspecting the XML file the system is able to determine that user "Ted" made annotation to the page http://site.com/page.asp?id=31, while "John" made and annotation to the page http://site.com/page.asp?id=27.

Each file is stored on the server's file system according to the web site's structure. For instance, annotations for the previous URL are stored in [Repository]/site.com/page.asp.xml, which clearly separates annotations from different pages and avoids having a large number of files in the same directory.

B. Indexing and Searching

Unlike relational databases, the file system does not have intrinsic search mechanisms. To make it possible to search annotations in an efficient manner, it is necessary to use an indexation application and create a searchable annotation index. At this point, EspiritUs uses Microsoft's Indexing Service [22], which by default filters only HTML, MIME messages, Office documents and simple text files. To enable the service to extract relevant and structured information from XML files, QLXFilter's [23] filter was installed and configured to parse the required XML attributes and values (such as URL and author).

An Indexing Service library was created to accept search requests and reply with the matching XML documents. EspiritUs uses the library when making free-text searches, date-related searches, and when retrieving all annotations from a single user.

To allow the system to be implemented on a non-Microsoft server, some alternatives to Indexing Service are being tested, such as Swish-e [24], an open-source tool with built-in XML index and search capabilities.

C. Authentication

Unlike annotations on desktop applications, web annotations must have a well identified author. This is a crucial aspect because each user must have access to his annotations and the access denied to others private annotations. The same behavior must occur when searching annotations.

An EspiritUs system may work with one of two possible configurations: using a built-in authentication module, with author identifiers generated and managed within the system; or bypassing the module and using an external authentication process. Each configuration applies to one of two main application scenarios discussed in section 6.

In the first case, each new user gets a unique identifier which, along with his personal information, is stored either in XML files or in a relational database, depending on how EspiritUs was deployed. This identifier is then used in the annotation XML's author tag.

On the second case, it is up to the external entity calling the EspiritUs Web Service to create unique identifiers for the authors. The system's authentication module is not used in this configuration.

D. Data Layer Interface

Once the organization model, the data structure, and the storage format were chosen, the interface between the data layer and the business logic layer was defined. To achieve a high interoperability and to allow access to the annotations by highly heterogeneous applications, a Web Service [25] interface was created at the Data Layer. Then, for each logical operation over the annotation system, a web method was created. The methods available are: Authenticate, GetNote, GetNotes, GetUserNotes, GetIfNotesExist, AddNote, RemoveNote, UpdateNote, and SearchNotes.

Web Services present some overhead due to serialization to and from the XML format. However, since XML is already the adopted format for the annotations, serialization time is most of the time negligible.

VI. Using the System

Due to its modular characteristics, the system may be reused in several distinct annotation applications. To do so, it is only required that the upper layers use the NoteAccess library. In the following sections, a brief description is made of two common usage scenarios.

A. Browser component

To allow any user to annotate web resources with EspiritUs, a component was developed to use in IE (5.0 or greater). After installing the plug-in, the browser is enhanced with the EspiritUs band (Fig. 3).

Once authenticated on the system, the user starts viewing annotations of the browsed web pages. The EspiritUs band displays only his annotations and all the public annotations from other users. The band also allows annotations to be added, edited or deleted.

Besides being able to view the annotations, the user may also do a word or phrase search in the entire repository by submitting a form. On each search result, a link is provided to browse to the web resource.

The EspiritUs administrator will periodically search for improper/offensive or advertising annotations. The annotations will be eliminated and their users eventually banned.

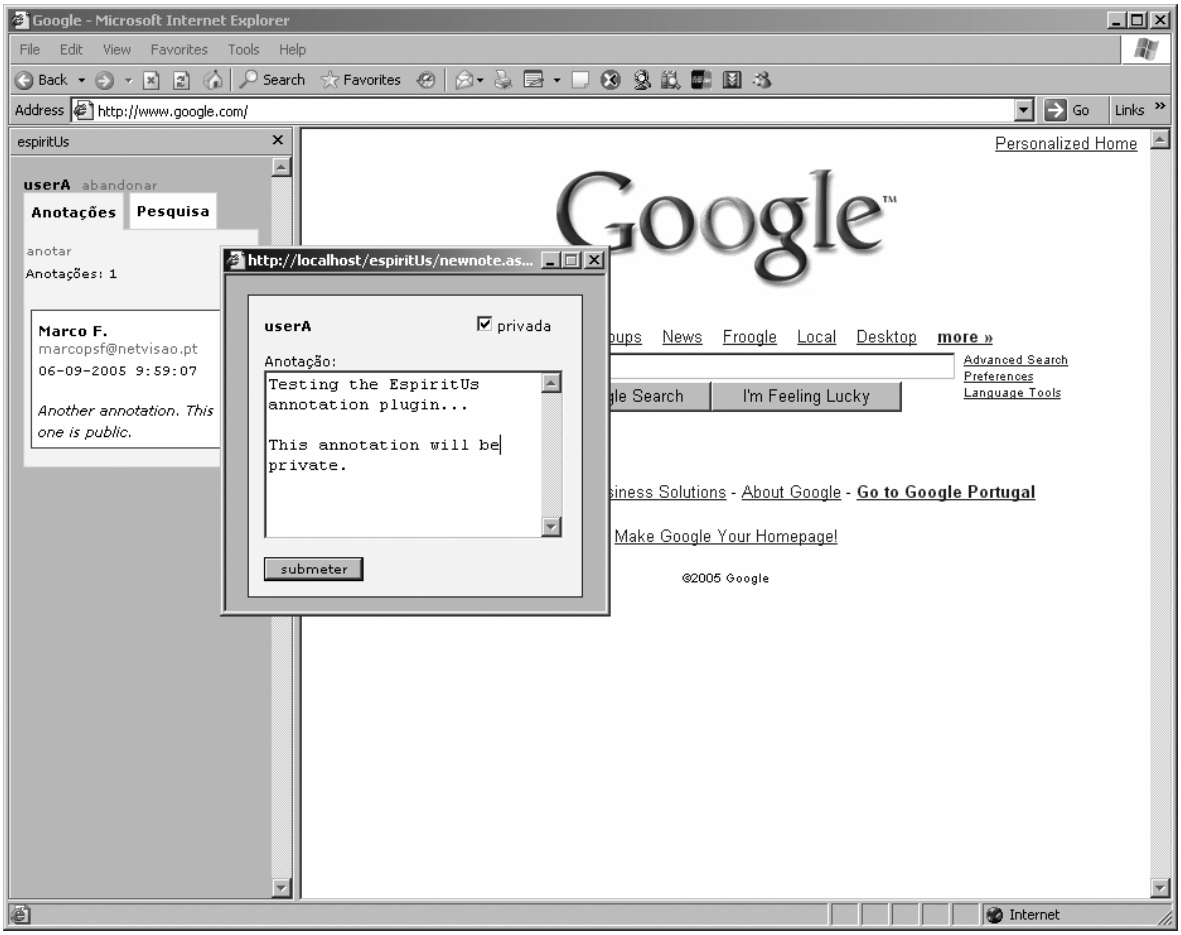

Figure 3. EspiritUs band on Internet Explorer

B. Web Application Scenario

Other possible usage scenario is integrating an annotation component on a web site. The administrator or webmaster may include this component only on the desired pages and allow its registered users to annotate those pages' content.

The main difference between this component and the client-side one resides on the search range. While on the client-side component the search is made on the entire public repository, with the web component the search is limited to annotations within the annotated private domain or site.

It is responsibility of the site's webmaster to remove improper annotations and eventually taking restricting measures over users.

VII. Usage Scenarios

There are several usage scenarios where EspiritUs can be used. Some of the most important are:

1. Development of web applications. By using the EspiritUs band object, developers and designers can easily communicate throughout the web site's construction. The band can also be used as a "to do" list.

2. Discussion forums. EspiritUs can be used in a manner similar to World Forum, using the band object, the component, or even both. Forums can be public or private.

3. Workflow and documental management. EspiritUs can also be used as a tool to aide the management of documents or even implement some sort of workflow process.

VIII. CASE STUDY: SVDP

The University of Aveiro developed the SVDP (Process Visualization and Digitalization System) web application for the Portuguese Justice Purveyor's Office.

SVDP is an application to allow searching, retrieving, presenting and annotating the digital processes. Its objective was to facilitate the consultation of processes without requiring going to the archives, building a digital archive. With SVDP, users can view the complete processes in digital format and create PDFs of them.

Once this phase was completed, the system was enhanced with the EspiritUs component on the digitalized processes pages (Fig. 4). Through EspiritUs, the Purveyor's Office's clerks can annotate processes and complement its limited information coming from the legacy database. EspiritUs can thus be used as personal notes (digital post-its) and to facilitate communication between employees working on a process. Besides that employees with the right privileges can use the system to enhance the digital processes with additional metadata, which will then be used by the index system. This way the overall digital archive will be incrementally enriched with metadata, as the users consult the processes and annotate them.

Furthermore, each site user may list his annotations and search for all annotations stored on the server.

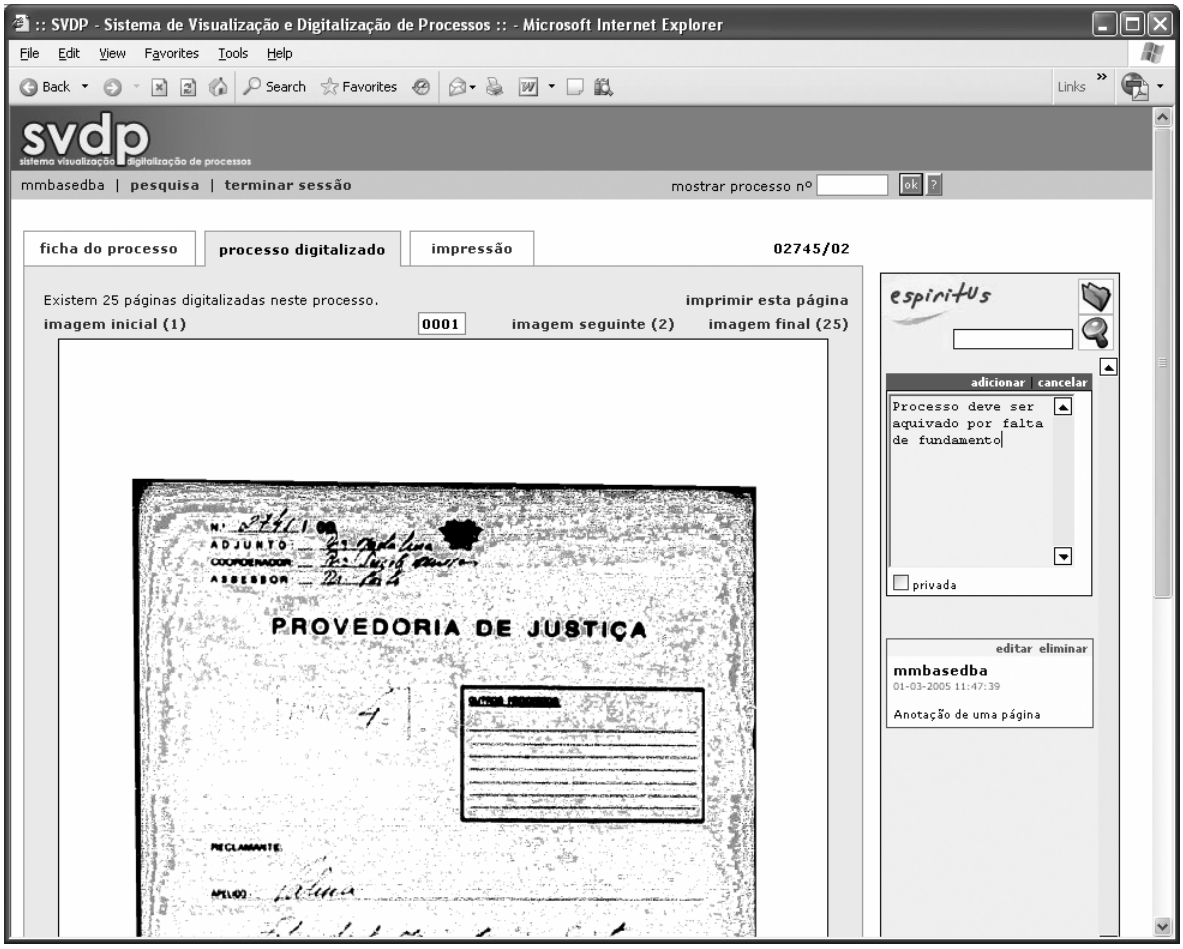

Figure 4. Case Study: EspiritUs on SVDP

Also, search capabilities on the web application were extended. On each page of a process, users can search for annotations made on that process. On the site's search page, users can search for annotations made on all existing processes. These are full-text searches.

IX. CONCLUSION

As desired, the annotation system allows unequivocally annotating a web document by associating the annotation to the resource's URL. Thanks to the Web Service interface on the data layer and the choice of the XML format to store and transport annotations, the system presents high interoperability and can be used by a vast number of applications from different platforms and operating systems.

On the other hand, due to the modularity of the system, the data layer may be reused by distinct applications. Practical cases of such implementations are the client-side component and the web application component already developed and in use.

One of the biggest advantages EspiritUs offers is the ability to search annotations in the entire system, those from a single domain, or simply those from a unique user. This is a feature missing from the related work analyzed.

X. Future Work

To allow for a larger flexibility using the EspiritUs system, future work involves dynamically inferring the XML annotation structure from a schema. This way, the annotation interfaces may also dynamically generate the forms and controls to view, create and edit annotations. For example, if the administrator adds a string element to the schema for a reply-to e-mail, the annotation interface could automatically generate a text box for filling that address and include it in the form.

A new scope is also planned to complement the existing public and private: group. Group annotations should be available only to a predefined group of users. This feature could be important in more complex scenarios with several permissions layers.

Considering the controversy generated by software such as Third Voice, tools will be developed to automatically remove spam and offensive annotations from the system. This is mainly applicable to the public server's system.

Other future works may involve creating new components for other browsers other than IE, such as Firefox, Netscape and Mozilla.

Acknowledgment

This work was funded in part by the Portuguese Justice Purveyor's Office (Provedoria de Justiça).

References

[1] Microsoft Corporation, "Microsoft Office Online Home Page", http://office.microsoft.com, accessed July 2005.

[2] Adobe Systems Incorporated, "Adobe Systems Incorporated", http://www.adobe.com, accessed July 2005.
[3] W3C, "Web Naming and Addressing Overview", http://www.w3.org/Addressing, accessed March 2005.
[4] W3C, "Annotea project", http://www.w3.org/2001/Annotea, accessed September 2005.
[5] W3C, "XPointer Framework", http://www.w3.org/TR/xptr-framework, accessed September 2005.
[6] Geof, "Web Annotation", http://www.geof.net/code/annotation, accessed September 2005.
[7] W3C, "Document Object Model (DOM) Level 2 Traversal and Range Specification", http://www.w3.org/TR/DOM-Level-2-Traversal-Range, accessed September 2005.
[8] Handschuh, S. e Staab, S., "Authoring and Annotation of Web Pages in CREAM", 11th World Wide Web Conference, Hawaii, USA, 2002.
[9] Ljunggren, C., "Janno", http://www.claus-ljunggren.dk/janno/web, accessed May 2004.
[10] Harvard Law School, "Annotation Engine - Berkman Center for Internet & Society", http://cyber.law.harvard.edu/projects/annotate.html, accessed March 2005.
[11] Gibeo LLC, "Gibeo Network – Community Web Annotation & Content Aware Tools", http://www.gibeo.net, accessed July 2005.
[12] Huyreka Solutions, "e-marked :: Home", http://www.e-marked.com, accessed September 2005.
[13] Obout Inc., "Memobook – The same at work and home (links/notes)", http://www.memobook.com, accessed July 2005
[14] Opera Software, "Opera Web Browser", http://www.opera.com, accessed July 2005.
[15] Wikalong, "Wikalong Firefox Extension", http://www.wikalong.org, accessed July 2005.
[16] Mozdev, "mozdev.org – annozilla: index", http://annozilla.mozdev.org, accessed July 2005.
[17] Lycos, Inc., "Wired News: Third Voice Trails Off", http://www.wired.com/news/business/0,1367,42803,00.html, accessed July 2005.
[18] W3C, "Amaya Overview", http://www.w3.org/Amaya, accessed July 2005.
[19] Kortermand, K., "Word Forum Home", http://www.worldforum.dk/en/home.php, accessed September 2005.
[20] Connoly, T., Database Systems – A Practical Approach to Design, Implementation, Management, Addison Wesley Logman Ltd, England, 1999.
[21] W3C, "Extensible Markup Language", http://www.w3.org/XML, accessed July 2005.
[22] Microsoft Corporation, "What is Indexing Service?", http://msdn.microsoft.com/library/default.asp?url=/library/en-us/indexsrv/html/ixintro_03l1.asp, accessed July 2005.
[23] QuiLogic/DV Inc., "XML Ifilter for indexing XML files", http://www.quilogic.cc/ifilter.htm, accessed July 2005.
[24] SourceForge, "Swish-e :: Home Page", http://swish-e.org, accessed September 2005.
[25] W3C, "Web Services Architecture", http://www.w3.org/TR/ws-arch, accessed July 2005.

Author Biographies

Marco Fernandes was born in Aveiro, Portugal, on May 1981. He received the Electronics and Telecommunications Engineer degree from the University of Aveiro, Portugal, in 2004. Since then he has been with the IEETA – Information Systems and Telematics Laboratory. He is a PhD student and his research focus primarily on Digital Libraries and Digital Archives and its implementation in distributed environments, web services and P2P. He is currently working in the project of the University's digital library/archive.

Miguel Alho was born in New Bedford, MA, USA, on July of 1981. He received his degree in Electronics and Telecommunications Engineering form the University of Aveiro in 2004. He is currently integrated in the University's digital library/archive development team at the information systems research laboratory at IEETA.

Joaquim A. Martins was born in Albergaria-a-Velha, Portugal, on November 21, 1956. He received the Electronic and Telecommunications Engineer degree from the University of Aveiro, Portugal, in 1979 and the PhD degree in the same area from the University of Aveiro, Portugal, in 1989. The subject of his thesis was a contribution to the study of protocol architectures for high speed local networks. Currently, his research is based on novel architectures of distributed information systems, multimedia digital libraries and archives, web services, semantic web, integration of IR technologies, integration of text and multimedia data, hypermedia and multimedia tools and systems and networks management resulting, up to now, in more than 100 papers. In 1998 he received the prize "Geriatria 'Dr. José Reis Jr.' do Biénio 1998-2000", from the Portuguese Society of Geriatrics and Gerontology. He began teaching in the Electronics and Telecommunications Department of the University of Aveiro in 1979, being Full Professor since 2004. From 1998-2001 he has been the Director of University of Aveiro Communications and Informatics Center. He is also senior researcher of the IEETA (Institute of Electronic Engineering and Telematics of Aveiro) Laboratory of Information Systems and Telematics. He participated in several National and European R&D projects in the aim of Telemedicine, Open and Distance Learning and Information Technology programs.

Joaquim S. Pinto is Assistant Professor in Electronics and Telecommunications Department at the University of Aveiro, Portugal. Mr. Pinto has a PhD since 1997 in Electronics and Telecommunications where he submitted a thesis titled "Architecture for a collaborative system based on hypermedia tools". He worked on several European projects on distance and collaborative learning: "Cooperative Learning" (Co-Learn), "Distance education and tutoring in heterogeneous telematics environments" (DEMOS) and "Distributed Virtual Laboratory" (DiViLab). He currently works on the development of Digital Libraries Tools and Applications and Web-based Information Systems. He has taught courses in programming languages, human-computer interaction and web-based information systems.

Pedro Almeida was born in Viseu, Portugal, on December 30, 1976. He received the Electronic and Telecommunications Engineer degree from the University of Aveiro, Portugal, in 2000 and the M.S. degree in the same area from the University of Aveiro, Portugal, in 2003. The subject of his dissertation was the development of High Granularity Video Servers, contributing his study to the development of some Audiovisual Archives in Portugal, namely the Parliamentary Audiovisual Archive and the Audiovisual Archive of the University of Aveiro. His research is based on the development of Digital Libraries, Archives and Museums, resulting up to now in more than 20 papers. In terms of work experience, he started in 2000 as a researcher in IEETA, being this his actual position.