Digitized ancient documents ... What's next?

Reim Doumat — Elöd Egyed-Zsigmond — Jean-Marie Pinon

Université de Lyon

LIRIS UMR 520

7, avenue Jean Capelle

F-69100 Villeurbanne

{reim.doumat, elod.egyed-zsigmond, jean-marie.pinson}@liris.cnrs.fr

ABSTRACT. Collections of cultural heritage documents are most of the time digitized images. These precious documents are now available on the internet and need manual annotations to make their content accessible and exploitable. Manual annotations are expensive and tedious; hence the use of a collaborative annotation space and the reuse of users' experiences, by tracing their actions -during the annotation process- in order to offer an assistance based on these traces is necessary. In this article we present our digital archive model and a prototype of the collaborative application to annotate online ancient manuscripts. The application offers an assistant for semi-automatic annotation, and a tracing system that saves traces of important actions in order to reuse them in a recommender system afterwards.

RÉSUMÉ. Les collections numériques de documents de l'héritage culturel sont la plupart du temps des images scannées. Actuellement ces documents sont disponibles sur internet, mais ont besoin d'annotations manuelles qui expliquent leur contenu afin de les rendre accessibles et exploitables. L'annotation manuelle est coûteuse et fastidieuse, mais le travail collaboratif peut l'accélérer. Ce travail collaboratif facilite la tâche d'annotation et mutualise l'expérience de chacun. Dans cet article, nous proposons un système qui trace les actions des utilisateurs pendant leur travail sur des documents et fournit une assistance basée sur ces traces. Nous présentons également un modèle d'archive numérique et un prototype pour l'annotation de documents numériques en ligne. Cette application permet aux utilisateurs de travailler en collaboration et de profiter d'un système d'assistance basé sur les traces d'utilisation.

KEYWORDS: cultural heritage documents, annotation, collaboration, traces, recommender system.

MOTS-CLÉS : documents d'héritage culturel, annotation, collaboration, traces, système de recommandation.

DOI:10.3166/DN.12.1.31-51 © 2009 Lavoisier, Paris

1. Introduction

Digitizing ancient texts, invaluable historical and cultural documents and artifacts (papyrus, stones and others) became widespread; it offers the possibility to all concerned institutions or persons to conserve old documents and to make them available to study and exploitation for a wide audience. Libraries that search to conserve and to facilitate the access of documents, historians who aim at studying old documents, monks who have precious ancient religious documents, museums that aim to exhibit their objects, and other organizations or people are interested in the digitization of ancient documents. Moreover, students or any attracted user can benefit from online available digital documents, since the original papers and artifacts are strictly conserved as well as they are in reach of only some authorized persons. The question is “*What after documents' digitization?*”. Users need to access the digitized documents easily to study their content, search them, and enrich them by annotations or classifications. For this reason, a management system is needed to store, visualize, organize, search and annotate these documents. That is why many projects and digital libraries have been developed for these purposes, and that is why we are interested in developing a digitized manuscript's management system.

Nearly all documents in digital libraries do not have *enough* annotations. Annotations are associated to documents either depending on metadata standards such as (DC) Dublin Core (Dublin Core Metadata Element Set, 2008), METS (METS, 2008), MARC21 (Marc Standards, 2008) and other standards of archive metadata such as ISAD/ISSAR¹, or depending on the application requirements with the aim of giving information about document content. In fact, we did not find a universal standard to be used in all libraries or museums. For example, annotations could be a type of local descriptors that mark some points in images (Valle *et al.*, 2008), and then these descriptors are used to identify, to index, to describe and to facilitate the retrieval of image documents. Other applications use textual annotations resulted by applying OCR (Optical Character Recognition) such as in the MEMORIAL project (Antonacopoulos and Karatzas, 2004), or by applying handwriting segmentation algorithms (Gatos *et al.*, 2007) to extract the content of scanned documents. Usually, in cultural heritage repositories, collections are sets of digitized images where the original papers are mostly handwritten documents. Document search and retrieval in these collections is annotation-based, it means that search tools use the descriptive metadata, transcriptions and other annotations to find the required document. Consequently, we perceive that their management system must allow adding annotations as metadata to facilitate the document retrieval. However, this work is laborious. Reading a manuscript page takes sometimes hours; thus publishing these documents online and offering annotation tools facilitates the annotation creation. The problem is that the created annotations are not validated, and users can do many mistakes (e.g. Haddad collection in Wellcome Library²). Therefore, allowing users to

1. www.ica.org/en/node/37326

2. <http://library.wellcome.ac.uk/haddad/419.asp>

correct or to comment on the annotations of other persons is useful. This is considered as a collaborative work to achieve good quality results in a difficult task. Furthermore, such a system must be intelligent enough to capture users' experience and to propose some assistance during the annotation process. In other words, observing users' actions can enhance simplicity and efficiency of application use, improve interaction quality, and save users time and effort.

In order to better understand users' needs, we contacted different organizations that seek to exploit and to study their rare digital collections like the Archdiocese of a catholic church in Aleppo, Syria or a library in Transylvania. All their documents are images of handwritten manuscripts and need manual annotations. Librarians aim to work in collaboration with researchers who are interested to study of these manuscripts.

In this paper we present an online digital archive application to manage and annotate ancient manuscripts manually and remotely in a collaborative and assisted environment. We offer a web application to exhibit the rare collections and to allow users to add their own annotations of different types, to share their work with others, and to benefit from an assistant for the annotation and the search. User interaction with the application is tracked to register the important actions and the consulted documents. After that, registered user actions are used by the assistant, to help and to recommend users according to what they are doing at that moment.

This article is organized as follows: Section 2 gives an overview about some annotation projects, collaborative and tracing systems. In Section 3, we present our assisted web archive design. Section 4 describes our prototype ARMARIUS, a web application to annotate scanned manuscripts online. We conclude by presenting our perspectives.

2. Background

Annotations and metadata are supplementary information added to an initial collection of data. While annotations are rather time-related "semantic" information, metadata are rather static information about a unit of a document³.

In our project, we are interested in three features:

1. Annotation and metadata structures to annotate different document categories (like image collections, images or image fragments), and making the used annotation types dynamically extendable.

2. The collaborative work between users in order to benefit from the experience of other persons.

3. [http://www.amiproject.org/pdf/L-Metadata and annotation standards.ppt/view](http://www.amiproject.org/pdf/L-Metadata%20and%20annotation%20standards.ppt/view)

3. Tracing users' actions and developing an assisting system based on the traces, to facilitate and accelerate users' work.

This section describes some annotation and user tracing projects. We show their work principles, and investigate their associated features and weaknesses.

2.1. *Annotation projects*

Since we are interested in manuscript image annotation through the web, we studied the projects about manuscript and image annotation as well as web page annotation. Numerous projects have developed diverse tools to annotate web pages, multimedia objects, or documents.

Several projects are directly interested in publishing annotated ancient manuscripts such as Scraps (documents from the World War I) (Scraps, 2008) and Terra de Santa Maria (Ribeiro *et al.*, 2007). The collections in these projects are previously transcribed and have metadata to reach them. Nevertheless, these projects do not enable users to annotate or to comment the exposed documents. The IPSA project (Agosti *et al.*, 2003) enables the annotation of herbal manuscripts. In this application user can add textual annotations remotely or add links between images. It also permits users to work in collaboration; however it does not offer the functionality to annotate image fragments.

The UVic IMT project (The UVic Image Markup Tools Project, 2008) offers image mark-up tools and permits to load and display a wide variety of different image formats. Uvic IMT enables the user to specify arbitrary rectangle shaped fragments on the image, to insert resizable and movable annotation areas on the image and to associate them with annotations. Uvic IMT marks the annotated parts in the image and stores the resulting data according to the TEI P5 XML (TEI: Text Encoding Initiative, 2008) syntax in files locally on the user's machine. The disadvantage of this application is that stored annotations cannot be seen by other internet users.

Another set of projects are focused on the collaborative annotation of multimedia web resources. During our study we distinguished projects that integrate plug-ins in web browsers to provide annotation tools such as Annotea (Kahan and Koivunen 2001); it permits to exchange web annotations and bookmarks between users in a collaborative way using RDF and semantic web compliant syntax. Nevertheless, the resources annotated in Annotea (web pages, images, texts, hyper links, etc.) are identified by standard URLs which means that image fragments are not linkable. For this reason we cannot use the Annotea plug-in to annotate our images. Another web browser extension project is MADCOW (Multimedia Annotation of Digital Content Over the Web) (Bottoni *et al.*, 2004); it offers the possibility to annotate many objects simultaneously and to use multimedia documents as annotations (audio registrations, figures...) in addition to textual ones. Yet again MADCOW annotates only XML documents and not images.

Although the semantic Web enables users to exchange their knowledge and to add meaningful information to documents, most existing resources are not designed to be “semantically” used online in a collaborative manner on multimedia documents. In Table 1 we summarize the interesting features of these projects.

Table 1. *A comparison between annotation projects*

		SCRAPS	Terra de Santa Maria	IPSA	Uvic IMT	Annotea	MADCOW
Media to annotate	Web pages					X	X
	images	X	X	X	X		
Specialized in ancient manuscripts		X	X	X			
Annotation type		DC	ISAD/ISSAR	Comments & linking annotations	TEI P5	RDF	Notes
Adding user annotations				X	X	X	X
Annotation stored	Locally				X		X
	server	X	X	X		X	X
Support collaboration				X		X	X

2.2. Collaborative annotations

A collaborative annotation environment is a platform with collaborative tools permitting to N participants to communicate, to annotate digital documents, to confront their annotations and opinions and to benefit from the experience of other users.

(Cabanac *et al.*, 2007) define an annotation as a *collective* object that can be used for *argumentative* purposes. The “collective” adjective refers to the fact that an annotation may be consulted by any user of the annotation system. The “argumentative” facet of an annotation is because when annotating a resource, the user may validate or refute its contents. Furthermore, annotating documents reflects the individual’s cognitive efforts (e.g. learning, correcting) while interacting with text documents (Cabanac *et al.*, 2008). The authors tackle the question about “*How to exploit collective annotations?*”. Users benefit from the group’s activities, and the group benefits from individual’s activities. The more users add annotations, the more search results are satisfying. Another example about distributing user knowledge is AnnForum (Chen and Persen 2008), an annotation tool that supports the reuse of collaborative knowledge and exploits the forum information as new learning resources.

We cited this example from the e-learning domain because of the likeness with our domain; we also need to distribute the information added by users into the archive. We perceive that collective annotations are recently privileged enabling users to exchange their knowledge; this is important in complex tasks such as

manuscript interpretation and annotation. Moreover, in a collaborative environment, tracing systems are used to know *who* did *what*.

One technique to support collaboration is to capitalize and reuse user experience. This is most commonly done by tracing their actions. This is why in the next section we talk about user traces.

2.3. User tracing projects

Tracing systems usually propose a technique based on a *task model*. A task model generally represents a schema for actions performed by the user. However, classical task models for user interfaces are not designed to represent web2.0 application use which is the case of our application. Task models such as CTT (Concur Task Tree) (CTT, 2008), Statechart diagrams (Statechart, 2008), GOMS Model (GOMS, 2008), HTA (Hierarchical Task Analysis) (Stanton, 2006) do not fit our application. We intend to register the user experience while using a web application where we need to model in a common structure complex and simple tasks in a multi hierarchical manner and state and window transitions as well.

Digital traces of information system use are registered elements of the interaction between the user and the system. Traces give information about the exploitation of the system (Laflaquière *et al.*, 2007). In this article, the authors define digital traces as automatic, usable, and temporal registrations issued from actions, where an action is a special type of activity or “intellectual work”. Trace components are observed elements that are related to the digital environment manipulation. A trace model contains the observed elements and the relations between them (temporal or other types); it is described by an ontology (RDF). The article, the authors propose the ATER system that collects, transforms, requests, visualizes and shares traces. The system assembles primary untreated traces, and then it registers them in a trace-base file in OWL/RDF. Users can navigate through traces and transform traces manually or through models. The disadvantage of the ATER system is that it has no defined trace model. Users and systems have to express their trace model and have to organize the storage, transforming and visualizing the trace. Trace transformation consists of selecting parts of traces to generate higher level traces relative to the user activity. Transformation types vary from a simple selection to developed filters.

Musette (Modelling USEs and tasks for tracing experience) (Champin *et al.*, 2003), is an approach to build traces conforming to a use model. Traces in Musette are composed by entities, events, and relations. Entities are objects being presented to the user in his interaction with the system, events are actions happening during the interaction, and relations are binary and can imply either entities or events. The use model describes what kind of entities, events and relations will actually be observable to produce a primitive trace. Trace components are not sufficient to build an observer, for this reason an observation model is needed to define the rules to produce traces. In Musette, the observation model is not specified.

The Trèfle model (Egyed-Zs, 2003) is quite close to the Musette model but represents users, actions and objects in the same graph. In addition, Trèfle introduces the notion of *potential graph* in order to filter the traces in comparable and reusable episodes. The basic assumption is that *users* manipulate *objects* through *actions* (procedures). For each application the manipulated objects modeled in a graph form the *use model*, the selected actions together with the users form another sub-graph called *observation model*. The objects are linked to the actions that manipulate them, each action having a set of objects as parameters. The use and observation models enable the tracing of the user-system interaction. The trace nodes are instances of users, objects and actions, they are attached with each other according to the relations defined in the observation and use model and they are also attached to the nodes they instantiate from these models. As the whole forms one graph, we can use graph theory based methods to filter the traces. A trace filter is expressed as a partially specified graph: the potential graph. Finding isomorphic sub-graphs in the traces gives episodes that can be compared and adapted to provide recommendations. This is done according to the case-based reasoning paradigm.

Traces are also frequently used in learning systems since they track users activities and help in improving the pedagogical scenarios. In (Settouti *et al.*, 2006) the authors talk about the necessity to help users based on traces by collecting, transforming, analyzing these later issued from observing human learning. Teachers define paths through documents that students have to follow depending on their learning objectives. Comparing teacher prescribed tasks with the user activities; traces participate to control the learning process. This is called TBS (Trace Based System) where traces are composed of observed elements associated to their trace model.

Traces in e-learning are used to study and to extract the user's behavior and his activity's objective. Every specific use of a resource may become a valuable source of knowledge about how that resource can be used. According to (Cram *et al.*, 2007), a trace is a computer object where data about the interaction between the user and the system are collected. The authors adopted the hypothesis that the visualization of actual user traces advances monitoring and analyzing processes during learning activities. They developed eMédiathèque, a collaborative platform with a tracing infrastructure to visualize actual user traces and to offer collaborative tracing tools. Visualizing a collective trace is in itself a collaborative tool since it permits to display the actions done by other users. The user in eMédiathèque can manipulate his trace to reveal the elements that he wants to be emphasized.

2.4. Conclusion

We found out from the state of the art that in spite of the large number of existing digital library systems and projects the majority of them do not offer annotation tools and do not contain collaborative work spaces. Even the personal data management is mostly reduced to a simple user identification to set document

access rights. No possibilities for example to keep bookmarks, subscribe to groups, precise documents or topics to follow. We think that it is important to provide such functionalities to increase the added value of online digital libraries. One of the purposes of these digital libraries is to provide a working space to geographically far located experts. With the arriving of Web 2.0 technologies that enable online access to functionally complex and graphically rich applications, it is necessary to provide theoretical digital library models and working prototypes that have these collaborative and personalized features. The interface should for example enable the selection of image fragments to annotate. It is also important to provide architectures that include image processing tools to assist and speed up the annotation process, although they can't make the document annotation fully automatic.

The progress in user modeling over recent years has shown that models learning from observing users' actions can enhance simplicity and efficiency, improve interaction quality and save users time and effort. However, relatively few applications use this technique hindering user assistance development. Users need an assistant that captures the experience from their use traces, for example to reduce the time and the effort of the search and the annotation process. Efficient trace exploitation is considerably better if the traces are conveniently modeled. It is possible to extract frequent items and correspondences using data mining methods, but if the tracing system has specific tools to cut the traces into comparable and reusable episodes, the capitalized experience will be retrieved faster and more precisely. Such tools are based on trace models that are often based on the tasks a user performs. The state of the art task models cover partially our needs because either they enable hierarchical task representation, like the Concur Task Tree model, or user interface related transitions like window transition diagrams. We try to model both in one model.

Another obstacle in intelligent user assistance is the bootstrapping problem. Very little about the user is known when the application is first installed on the user's system (Nguyen *et al.* 2007). To overcome this problem we search to assist the user during his work, depending both on his actions and on his profile.

In the next section we are explaining in details the design of our web archive to annotate manuscripts and to assist users during this process.

3. Assisting users in digitized manuscript annotation

3.1. Annotation

In this paper we consider annotations as information added by the users on document units. A document unit is an abstract document fragment that can represent a collection of images, a simple image or an image fragment. Figure 1 shows a screenshot of the annotation interface with annotated image fragments and annotation editing popup.

Figure 1. Annotation interface in ARMARIUS

Figure 2. System data model diagram

Figure 2 shows a simplified description of our data structure where we separated annotations, annotation types and values. The motivation is to have the flexibility to

add new types of annotations in the future (question, answer, comment, correction, title, author...) and to structure these annotations depending on one or several models. Annotation value is linked to the language in order to specify the language of each value. This technique enables the user to filter the results of his search to view the annotations in selected languages.

Only authenticated users can use the annotation tools, filter the visible annotations according to the groups of the author, type..., validate or correct them. Users do not have defined roles; they belong to groups. The groups have read and write access rights to collections. The system traces user actions in order to provide experience based assistance. It enables the exploitation of collective annotations, thus it distributes the experience and the knowledge of experts in a domain to other users. Since annotators are not all experts, there is a need to verify their production of annotations. In our application, users who have enough privileges can modify or delete other users' annotations. The objective is to make emerge correct and complete annotations about the document by using different user experiences.

3.2. Traces

Our system's main design objective is to capture user's activity, in order to build an assistant that is capable to reuse these activities in similar situations. The question is: how can the system potentially reuse information created by one user for another user? This is the role of the assistant.

Figure 3. An example of assisting annotations in Arabic manuscripts

Let us see an example about reusing a traced user action (experience) to assist another person (Figure 3). *UI* is annotating a fragment in an image; he defines the points that surround a fragment and begins the transcription by adding annotations to

the fragment. The system verifies if any of these words exist in the annotations, and then searches the fragments that contain this word. The assistant proposes the user to add the other words attached to these fragments. Furthermore, to benefit from other user traces, the assistant system searches in the traces to see if the value added by the user exists and if it has been modified to a new value (corrected). If this is the case: an earlier instance of the just entered keyword was corrected, the system proposes the new value (corrected one) held by the annotation. Figure 3, illustrates that (Fragment 2) was corrected by a user who knows the Arabic, thus when another user defines (Fragment3) at the last line, and writes a word corresponding to an old value of the (Fragment 2) before the correction, the system will notify the user that another fragment was corrected in a similar situation, and proposes him to change the annotation value. Alternatively, the user can refuse the assistant's proposition.

These few examples illustrate the different uses of trace based assistance. We consider that the user knowledge can be represented either by the annotations he adds or by the modifications that he makes. For instance, when a user adds transcriptions, this means that he knows the language of the manuscript, when he modifies the value of an annotation this indicates that he is probably correcting some mistakes. The proposed knowledge exchange system couples case-based reasoning with ontological structures (task model, annotation schemas, domain dependent taxonomies) to assist match-making between existing knowledge and knowledge demands. Figure 7 illustrates the assistance system which exploits the stored traces to provide step by step assistance to most current user actions. Traces are registrations of some users' activities while he is exploiting the system.

In order to introduce our trace based recommendation method, we define formally the traces. In our model a trace is a list of actions. Actions are chosen from an observation model (Figure 5) and have parameters. The observation model is taken from the Trèfle model (Egyed-Zs., 2003).

3.2.1. Definitions

– Definition 1: *Action* in our concept is an event made by the user that affects a document unit during a session. This action is registered with its parameters in order to be compared with other actions.

– Definition 2: *Action Parameter*, parameters are the objects affected by an action. Parameters have types and values. Ex: variables that hold the value of metadata (annotations) or the values of coordinate points that define a fragment in an image. In general any object that can be manipulated by the system can be action parameter.

– Definition 3: *Action type* is a pre-defined value the action has. In our application the traced action types are: *select*, *browse*, *add*, *create*, *delete*, *search* and *modify*.

– Definition 4: *Trace* in our notion is the record of user actions during the system exploitation.

– Definition 5: *Episode* in this document is a group of user actions concerning the work done on one document unit during a session. It is a sequence of actions extracted from a trace by an episode extraction function. We claim that episodes can represent other action series as well extracted on other criteria than the concerned document unit. These criteria can be: the manipulating user, a time interval, an action type, given annotation value...

An *episode extraction function* enables the extraction of action sequences from a trace. For the moment, in our system the episode extraction functions are based on document units. An episode can contain sub-episodes; in this case it is called *parent episode*.

3.2.2. Trace decomposition

The system traces the actions of the connected user; traces are stored in a relational database that keeps track of the actions together with the affected objects (collections, pages, metadata...). As shown in Figure 2, traces are composed of episodes ep ; each episode is composed of an ordered list of actions Ac_i :

$$ep = (Ac_i, \quad i = 1, n)$$

Actions have a type A_{Type} as shown in Figure 2 and a set of parameters P_i

$$Ac_i = (A_{Type}, P_i \ (i=1, m))$$

For example: an action with the type: *AddAnnotation* has the parameters: *AnnotationID* and *DocumentUnitID*.

A parameter P has a parameter type P_{Type} and parameter value P_{value}

$$P_i = (P_{Type}, P_{value})$$

An example about a user trace is shown in Figure 4 with the trace decomposition into episodes.

Figure 4. Episode composition

Figure 4 is an example of a trace; it is composed of an episode-parent *Episode1* about the collection “Arabic manuscripts”. This episode contains two sub episodes *Episode1.1* about image “img_01” and *Episode1.2* about the image “img_02”. Furthermore, *Episode1.2* has two sub-episodes: *Episode1.2.1* and *Episode1.2.2* concerning two fragments of the image.

In our application we consider that the actions (create, select, browse) are similar when we compare the traces. The actions (create, select, add, search) are used to cut the trace into episodes, these actions begin new episodes. Search begins a new episode and a sequence of search actions is grouped together in one episode. Usually the actions (delete and modify) follow the select action. Close action ends an episode and logout terminates a session trace.

The system captures a user experience by registering user's actions during his work, then cuts the actions sequence into episodes depending on the document unit id. Trace episodes are generated at the moment the user interacts with the system; they can have a hierarchical structure “levels” according to the document unit level. For example, first level concerns the episodes handling collections and also called collection level, second level “image level” concerns images and the third one “fragment level” concerns image fragments.

3.2.3. Trace comparison and similarity

The actions of the users are traced according to the task model in Figure 5. This model is inspired from the Concur Task Tree model with some modifications that include user interface based composition.

To assist a user during his session, we compare the last episode of his traces with the episodes of the already registered traces in the system.

Throughout this paper we use the word “*trace similarity*” as a measure of likeness between two users' traces. *Trace similarity* depends on several comparison factors; one of them is *Annotation similarity*, since users exploit mainly the system to annotate documents and to search information from annotations. Annotation similarity is the result of a semantic relation (like synonymy or hyponymy) or word similarity. It can be calculated by different algorithms such as Edit distance or the use of an ontology. In our application, given that annotations are associated to document units, the similarity between document units is based on their annotations similarity. Traces in our application are considered as data sequences cut into episodes depending on the concerned document unit level. A collection is at a higher level than a page which is above a fragment. Episodes are composed of actions related to one document unit. In what follows, we define a simple comparison method for episodes. We agree that more complex algorithms could be used, such as an adaptation of the BLAST (BLAST 2009) algorithm (Basic Local Alignment Search Tool). BLAST is used to rapidly compare a given gene sequence X to the sequences in a database to find all similar ones.

Figure 5. Task model part of the observation model

We define:

$|e_i|$ = episode i length, the number of the actions in the episode

$|a|$ = parameter number of the action a

The next formula gives the similarity Sim_{ep} between two episodes e_1 and e_2 :

$$Sim_{ep}(e_1, e_2) = \sum_{i=1}^{\min(|e_1|, |e_2|)} (Sim_{Action}(a1_i, a2_i))$$

where $a1_i \in e_1$ and $a2_i \in e_2$ with $i: 1.. \min(|e_1|, |e_2|)$

There are other similarity measures between two episodes, for example the edit distance but we chose this simple comparison for our first tests. To calculate the similarity Sim_{Action} between two actions $a1, a2$ we define:

$$Sim_{Action}(a1, a2) = Sim_{Type}(T_1, T_2) * \sum_{i=1}^{\min(|a_1|, |a_2|)} Sim_{par}(P1_i, P2_i)$$

Where: $i: 1.. \min(|a_1|, |a_2|)$

$$Sim_{Type}(T_1, T_2) = \begin{cases} 0 & \text{if } T_1 \neq T_2 \\ 1 & \text{if } T_1 = T_2 \end{cases}, \text{ and } Sim_{par}(P_1, P_2) = \begin{cases} 0 & \text{if } P_1 \neq P_2 \\ 1 & \text{if } P_1 = P_2 \end{cases}$$

The definitions above represent only a preliminary version of our episode comparison methods.

Comparing episodes enables finding similar situations to the current one. Once we identified the most similar situation we can calculate the recommendation to assist the user.

3.3. Assistance

The goal of the trace based recommendation is to provide help in situations that are difficult to formalize. The problems aimed are “what to do?” type problems rather than “how to do?” type problems. Transcribing a text fragment in an ancient manuscript needs interpretation, historical and linguistic knowledge and it is also possible that there can be several correct transcriptions. The assistance of tasks like the transcription is a typical situation where trace based assistance can work. It can be useful to see the fragments transcribed by the same words, but also to see the annotations added by users with a similar navigational path as ours.

We are developing an assistant system which recommends users to do an action or to modify some parameters. The recommendations depend on other users experiences. When comparing episodes, the system compares episodes that have the same level, and then the assistant can recommend user using episodes of the same or of other levels.

The assistant compares the higher level last unfinished episode in the current user trace with the episodes, of the same level, registered in the database to find the most similar episode to the unfinished one. It continues then with lower level unfinished episodes. Subsequently the assistant recommends either the actions that are in the similar episode or the actions that are in the next episode. A smarter assistant can study the context of the similar episode and suggest more intelligent recommendations. The goal of the task model and the episode comparison is to

guess the user's intention. A finished episode contains the final goal that is a higher level complex task, such as: annotate a document, find some fragments having just vague ideas of what it may contain. Besides the final goal, which is usually achieved by the last action, the episode contains also the preliminary actions leading to the realization of that final goal. An unfinished episode contains only preliminary actions. The objective of retrieving similar episodes to an unfinished one, based on the comparison of the existing preliminary actions, is to retrieve the final action of the capitalized similar episode. This action or its parameters, maybe adapted to the current context, will give the concrete recommendation.

Figure 6. Comparison between traces to make recommendations

Figure 6 shows an example of the assistant. The user “U3” is the current user. She annotates an image from the “Arabic manuscripts” collection. The system compares the traces of user “U3” with other users’ traces, registered in the database to find similar ones. The results of the comparison are the traces of “U1” and “U2” that concern the “Arabic manuscripts” collection, users have browsed it after searching the word “Arabic”. Episodes of images are similar in relation to “Img_01” image, hence the system continue to compare the fragment episodes. The system finds that “delete annotation” action is followed by “add annotation” action which means that a correction of the annotation “Food” was made in this fragment. The assistant will use this correction to propose to the user “U3” the correction “feed” when he enters the annotation “food”. The assistant supposes that the user made a mistake according to the trace of “U2”.

When a user marks some points on the image to create a fragment; the coordinates of these points are passed to the system to create the fragment

boundaries, then the user chooses an annotation type and begins entering his words. At this stage the system suggests the user some of existing keywords after calculating the similarity between his fragment and other stored fragments. The annotator can accept the suggestions or continue his work.

The system starts from the current user action to the proposition of the assistance (e.g. see similar document units, similar fragments, or annotation) depending on other comparable traces. Our objective is to assist the user in accordance with his actions.

We consider that it is important to imply the user in the tracing and recommendation process. The user has to be aware that his actions are traced, and how this tracing is done. We provide methods to show the trace in construction to the user, enabling him to stop the tracing. We also show for the recommendation the episode that gave them.

Of course, the tracing and the recommendation is to be shown in a non intrusive way on the user interface. The primary goal of the user is to carry out the annotation or the search. The currently created trace and the recommendation should appear somewhere at the edge of the main interface.

4. Our prototype (ARMARIUS)

We developed the application ARMARIUS to test the functionalities desired in treating scanned manuscripts. Our application contains three collections of ancient manuscripts, two of them are from Syria in Syriac and in Arabic, and the third collection is from Transylvania. Collections, metadata, annotations and user accounts and rights are stored on the server.

ARMARIUS prototype employs the pattern MVC2 (Model View Controller 2). We build it over three tier architecture: presentation layer, application layer and data or persistence layer. We used JSF for the presentation layer and Hibernate for the persistence layer. We choose Mysql for our database and Glassfish for the web server. Because of the separation between the three layers, the use of this architecture makes it easy to expose our application functions in web services (research, visualization, annotation, manual transcription...).

Research results are images of manuscript pages, thus users can directly inspect the images to see if they are of interest. Annotation is done via the web browser. Users annotate new image fragments by choosing the annotation tool then specifying the image fragment to be annotated, the system creates a closed shape based on the user's points then offers a dialog box allowing user to add keywords and/or transcriptions or other metadata as shown in Figure 1. Note that the current trace of the user is shown at the top of the interface and the recommendations are in a resizable and movable frame.

Moreover, users can add new annotations to already existing fragments or modify the annotations. The more annotations are added, the more it improves search results since the system returns the pages containing the searched keywords.

Figure 7. System architecture

Figure 7 gives a global view about our web application that consists of the following objects:

1. Collections of scanned manuscripts, metadata, and user annotations all are stored on a server that holds also the web application;
2. Metadata containing general information about the manuscripts stored also on the same server. In our application we chose Dublin Core to represent the metadata because of its simplicity and the ability to duplicate its elements;
3. Users can test the archive site and see a demo about its digital collections about the manuscripts;
4. Users who want to work on digital manuscripts need to contact the administrator in order to have a user profile;
5. Once a new user account is created, the user is added to a group or more of users;
6. User groups have different access rights to browse and to manipulate the stored data (images and annotations);

7. Users connect to the server through a web browser and interact with the application throughout web interfaces;

8. During a session, user produces different actions in browsing and annotating the images. These actions affect the document units (collections, image collections, or image fragments) and the annotations;

9. Chosen user actions about manipulating and browsing digital manuscripts, during a session, are registered in the database as traces;

10. The system uses the registered traces and the stored annotations as well as user group rights to recommend user with annotations or actions.

5. Conclusion and further work

In this article we presented a web application to expose and annotate the digitized images about old and handwritten manuscripts. The annotation process is done manually and remotely. We also explained that users annotate using different types of metadata (transcriptions, keywords, comments) with the ability to add more annotation types in accordance to each archive needs. Users can work in collaboration to realize a correct annotation; they can see the annotations added by other users and modify them under given access right conditions. Users can benefit from the experience of the others to annotate documents. The system builds an assistant based on the traces of expert users.

For the moment the system recommends only actions to carry out or parameters to use or modify. We are currently working on extending these recommendations to users to contact or groups to join.

We are aware that user access right control at the annotation level is an important issue also, this could for example enable users to define access rights to their own annotations (private or public), but it is not implemented yet. In our future work we intend to improve the tracing technique and the assistance system, to facilitate the annotation process for users who do not have enough experience in annotating manuscripts. We also intend to extend the task model to trace other actions.

6. References

- Agosti M., Benfante L., Orio N., "IPSA: A digital archive of herbals to support scientific research", *Digital Libraries: Technology and Management of Indigenous Knowledge for Global Access*, Springer Berlin/Heidelberg, 2003, p. 253-264.
- Antonacopoulos A., Karatzas D., "Document Image Analysis for World War II Personal Records", *First International Workshop on Document Image Analysis for Libraries DIAL'04*, USA, 2004, p. 336-341.
- BLAST, algorithm, <http://blastalgorithm.com>, 2009.

- Bottoni P., *et al.*, "MADCOW: a multimedia digital annotation system", *Proceedings of the working conference on Advanced visual interfaces*, 2004, p. 55-62.
- Cabanac G., Chevalier M., Chrisment C., Julien C., "Collective Annotation: Perspectives for Information Retrieval Improvement", *Proceedings of the 8th conference on Information Retrieval and its Applications RIAO'07*, May 2007, USA, CID.
- Cabanac G., Chevalier M., Chrisment C., Julien C., "Exploiting the Annotation Practice for Personal and Collective Information Management", *Proceedings of the International Workshop on Model Driven Information Systems Engineering MoDISE-EUS*, 2008, France, p. 67-78.
- Champin P.A., Prié Y., Mille A., "Musette : Modelling USEs and tasks for tracing experience", *Workshop 5 From Structured Cases to Unstructured Problem Solving Episodes For Experience-Based Assistance, ICCBR'03*, 2003, Trondheim, Norway, p. 279-286.
- Chen W., Persen R., "Reusing Collaborative Knowledge as Learning Objects- The Implementation and Evaluation of AnnForum", *Times of Convergence. Technologies Across Learning Contexts*, vol. 5192/2008, Springer Berlin/Heidelberg, p. 92-103.
- Cram D., Jouvin D., Mille A., « Visualisation interactive des traces et réflexivité : application à l'EIAH collaboratif synchrone eMédiathèque », *STICEF, Numéro spécial Analyse des traces d'interactions dans les EIAH*, vol. 14, 2007.
- CTT, <http://giove.isti.cnr.it/ctte.html>
- Digitized Arabic Manuscripts- American University of Beirut, <http://ddc.aub.edu.lb/projects/jafet/manuscripts>
- Dublin Core Metadata Element Set, <http://dublincore.org/documents/does>
- Egyed-Zsigmond E., Mille A., Prié Y., "Club (Trèfle): a use trace model", *Proceedings of International Conference on Case Based Reasoning ICCBR03*, Norway, 2003, Springer-Verlag, p. 146-160.
- Falk I., Représentation et stockage des données de la numérisation du dictionnaire Trévoux, 2006, Nancy, France, LORIA.
- Gatos B., Antonacopoulos A. and Stamatopoulos N., "Handwriting Segmentation Contest", *Proceedings of the Ninth international Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007, vol. 2, ICDAR. IEEE Computer Society, p. 1284-1288.
- GOMS Model, http://www.cc.gatech.edu/classes/cs6751_97_fall/projects/closet2000+/FinalEssays/goms.html
- Greenstone Digital Library software, <http://www.greenstone.org>
- Kahan J. and Koivunen M., "Annotea: an open RDF infrastructure for shared Web annotations", *Proceedings of the 10th international Conference on World Wide Web WWW '01*, 01-05 May, 2001, NY, ACM, p. 623-632.
- Laflaquière J., Settout L.S., Mille A., *Un environnement pour gérer des traces comme inscriptions de connaissances*, September, 2007, Lyon, France, LIRIS.
- Marc Standards, <http://www.loc.gov/marc/index.html>

- May M., George S., Prévôt P., "A closer look at tracking Human & Computer Interaction in Web-Based communications", *Proceedings of the 6th international Conference on Human Learning ICHSL'08*, 2008, 14-16 May, 2008, Toulouse, France.
- Metadata Encoding and Transmission Standard (METS), <http://www.loc.gov/standards/mets/>
- Nguyen A., Denos N., and Berrut C., "Improving new user recommendations with rule-based induction on cold user data", *Proceedings of the 2007 ACM Conference on Recommender Systems RecSys '07*, 2007, NY, ACM, p. 121-128.
- Riberio C., David G., Calistru C., "Multimedia in Cultural Heritage Collections: A Model and Application", *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, vol. 4822/2007, Springer Berlin/Heidelberg, 2007, p. 186-195.
- Scraps, http://lettuce.tapor.uvic.ca/~scraps/viewer/viewer.php?collection=ww_one & artefact=do
- Settouti L.S., Prié Y., Mille A., « Systèmes à base de trace pour l'apprentissage humain », *Colloque international Technologies de l'Information et de la Communication dans l'Enseignement Supérieur et l'Entreprise TICE'06*, Toulouse, 2006.
- Stanton, N.A., "Hierarchical task analysis: developments, applications and extensions", *Applied Ergonomics*, vol 37, 2006, p. 55-79.
- Statechart diagrams, http://en.wikipedia.org/wiki/State_diagram
- TEI: Text Encoding Initiative, <http://www.tei-c.org/index.xml>
- The UVic Image Markup Tools Project, http://www.tapor.uvic.ca/~mholmes/image_markup/
- Valle E., Cord M., Philipp-Foliguet S., "Fast identification of visual documents using local descriptors", *ACM Symposium on Document Engineering DocEng'08*, 2008, Sao Paulo, Brazil, ACM, p. 173-176.